

# EFFICIENT APPROXIMATE-ML DETECTION FOR MIMO SPATIAL MULTIPLEXING SYSTEMS BY USING A 1-D NEAREST NEIGHBOR SEARCH

Dominik Seethaler, Harold Artés, and Franz Hlawatsch

Institute of Communications and Radio-Frequency Engineering, Vienna University of Technology  
Gusshausstrasse 25/389, A-1040 Vienna, Austria (Europe)  
phone: +43 1 58801 38958, fax: +43 1 58801 38999, email: dominik.seethaler@tuwien.ac.at  
web: http://www.nt.tuwien.ac.at/dspgroup/time.html

## ABSTRACT

It is known that suboptimal (equalization-based and nulling-and-cancelling) detectors for MIMO spatial multiplexing systems cannot exploit all of the available diversity. Motivated by the insight that this behavior is mainly caused by poorly conditioned channel realizations, we propose the *line-search detector* (LSD) that is robust to poorly conditioned channels. The LSD uses a 1-D nearest neighbor search along the least significant singular vector of the channel matrix. It exhibits near-ML performance and has significantly lower complexity than the sphere-decoding algorithm for ML detection.

## 1. INTRODUCTION

It is well known that the diversity offered by multiple-input/multiple-output (MIMO) fading channels cannot be fully exploited by suboptimal (i.e., linear equalization and nulling-and-cancelling) detectors [1]. Maximum-likelihood (ML) detection exploits all of the available diversity but is computationally intensive. Thus, there is a strong demand for computationally efficient suboptimal detectors that can exploit a large part of the available diversity.

In this paper, we propose the novel *line-search detector* (LSD) for spatial multiplexing systems. The LSD minimizes the ML distance criterion over a reduced set of data vectors that is found through a 1-D nearest neighbor search along a straight line that is parallel to the least significant right singular vector of the channel matrix. This is motivated by the observation that the inferior average performance of standard suboptimal detectors is mainly caused by channel realizations with only one very small singular value.

Our paper is organized as follows. In the remainder of this section, we describe the system model and briefly review existing detection schemes. The 1-D nearest neighbor approach is discussed in Section 2, and the resulting LSD algorithm is described in Section 3. Finally, simulation results are presented in Section 4.

**SYSTEM MODEL.** We consider an  $(M_T, M_R)$  MIMO channel (i.e., there are  $M_T$  transmit antennas and  $M_R \geq M_T$  receive antennas), and a spatial multiplexing system such as V-BLAST [1]. At each time instant, the baseband system model is given by

$$\mathbf{r} = \mathbf{H}\mathbf{d} + \mathbf{w},$$

with the data vector  $\mathbf{d} \triangleq (d_1 \cdots d_{M_T})^T$ , the  $M_R \times M_T$  channel matrix  $\mathbf{H}$ , the received vector  $\mathbf{r} \triangleq (r_1 \cdots r_{M_R})^T$ , and the noise vector  $\mathbf{w} \triangleq (w_1 \cdots w_{M_R})^T$ . The data symbols  $d_m$  are assumed zero-mean and white with variance  $\sigma_d^2$ . The noise components  $w_i$  are assumed independent and circularly symmetric complex Gaussian with variance  $\sigma_w^2$ . The channel  $\mathbf{H}$  is considered constant over a block of  $N$  consecutive time instants and perfectly known at the receiver.

**REVIEW OF DETECTION SCHEMES.** Major suboptimal detection schemes for spatial multiplexing systems include linear equalization

followed by quantization and nulling-and-cancelling (or decision-feedback) detection [1]. In linear equalization based schemes, the detected data vector is  $\hat{\mathbf{d}} = Q\{\mathbf{y}\}$  with  $\mathbf{y} = \mathbf{G}\mathbf{r}$ , where  $Q\{\cdot\}$  denotes componentwise quantization according to the symbol alphabet. The zero-forcing (ZF) equalizer is given by  $\mathbf{G} = (\mathbf{H}^H\mathbf{H})^{-1}\mathbf{H}^H$  (assuming that  $\mathbf{H}$  has full rank), while the minimum mean-square error (MMSE) equalizer is given by  $\mathbf{G} = (\mathbf{H}^H\mathbf{H} + (\sigma_w^2/\sigma_d^2)\mathbf{I})^{-1}\mathbf{H}^H$  [2]. In contrast to linear equalization schemes, where all layers are detected jointly, nulling and cancelling (NC) uses a serial decision-feedback approach to detect each layer separately. NC is based on the ZF or MMSE approach [1]; the corresponding detectors will be referred to as NC-ZF and NC-MMSE, respectively.

Finally, ML detection [3, 4] yields minimum error probability for equally likely data vectors. The ML detector is given by

$$\hat{\mathbf{d}}_{\text{ML}} = \arg \min_{\mathbf{d} \in \mathcal{D}} \|\mathbf{r} - \mathbf{H}\mathbf{d}\|^2, \quad (1)$$

with  $\mathcal{D}$  the set of data vectors. In general, the computational complexity of ML detection grows exponentially with  $M_T$ . Using the Finke-Phost sphere-decoding algorithm [4], ML detection can be achieved at an *average* complexity of roughly  $\mathcal{O}(M_T^3)$  [5].

## 2. THE 1-D NEAREST NEIGHBOR APPROACH

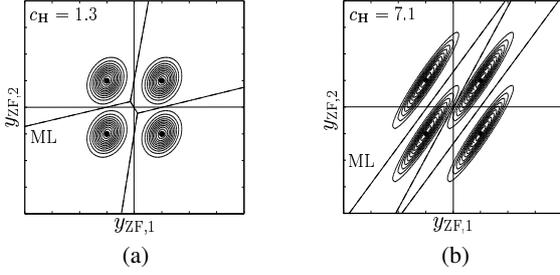
The LSD is motivated by the effect of “bad” (poorly conditioned) channel realizations on equalization-based detection. In what follows, we will use the singular value decomposition (SVD)  $\mathbf{H} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^H$ , where the diagonal matrix  $\mathbf{\Sigma}$  contains the singular values  $\sigma_m$  of  $\mathbf{H}$  (indexed in nonincreasing order) and the columns of  $\mathbf{U}$  and  $\mathbf{V}$  are the left and right singular vectors of  $\mathbf{H}$ , respectively [6]. The *condition number*  $c_{\mathbf{H}} = \sigma_1/\sigma_{M_T}$  is the ratio of the largest and smallest singular values; it is large for a bad channel.

### 2.1. Detector Performance for Bad Channels

As is well known, for  $c_{\mathbf{H}} = 1$  ZF and ML detection are equivalent but for growing  $c_{\mathbf{H}}$  suboptimal detectors perform much worse than ML detection (e.g., [7]). We will study this behavior in the “ZF-equalized domain” in which  $\mathbf{y}_{\text{ZF}} = \mathbf{d} + \tilde{\mathbf{w}}$  is the *undistorted* data  $\mathbf{d}$  corrupted by *correlated* noise  $\tilde{\mathbf{w}}$  with covariance matrix

$$\mathbf{R}_{\tilde{\mathbf{w}}} = \sigma_w^2 (\mathbf{H}^H\mathbf{H})^{-1} = \sigma_w^2 \mathbf{V}\mathbf{\Sigma}^{-2}\mathbf{V}^H.$$

The contour surfaces of the probability density function (pdf) of  $\tilde{\mathbf{w}}$  are hyperellipsoids whose  $m$ th principal axis is given by the  $m$ th eigenvector  $\mathbf{v}_m$  of  $\mathbf{R}_{\tilde{\mathbf{w}}}$ , with its length proportional to  $\sigma_w/\sigma_m$  [8]. Thus, ZF equalization results in a distortion of the noise pdf relative to the spherical pdf geometry of  $\mathbf{w}$ . For illustration, Fig. 1 shows the pdf of  $\mathbf{y}_{\text{ZF}}$  for a “good” and a “bad” realization of a real-valued (2, 2) channel and BPSK modulation. Also shown are the ZF decision regions (the four quadrants) and the ML decision regions (indicated by dash-dotted lines). For the good channel, the ZF and ML



**Figure 1:** pdf contour lines and ZF and ML decision regions in the ZF-equalized domain for a real (2,2) channel and BPSK modulation: (a) “Good” channel realization ( $c_{\mathbf{H}} = 1.3$ ), (b) “bad” channel realization ( $c_{\mathbf{H}} = 7.1$ ).

decision regions are similar. For the bad channel, however, they are very different. In particular, two of the boundary lines separating the ML decision regions have an offset perpendicular to the dominant principal noise axis  $\mathbf{v}_{M_T}$ ; such an offset cannot be obtained with any linear scheme. We note that also the decision regions of the MMSE and NC schemes are not completely adapted to the distorted pdf of  $\tilde{\mathbf{w}}$ , although they are better than the ZF decision regions.

## 2.2. ML Detection for an Idealized Bad Channel Model

Experiments indicate that for a bad channel, the largest ZF-domain noise component (whose variance is the largest eigenvalue of  $\mathbf{R}_{\tilde{\mathbf{w}}}$ ,  $\sigma_w^2/\sigma_{M_T}^2$ , and whose direction is given by  $\mathbf{v}_{M_T}$ ) tends to dominate all the other noise components. Hence, this dominant noise component causes the main part of the bad channel effects that plague linear and NC detection. Accordingly, we will approximate a bad channel  $\mathbf{H} = \mathbf{U}\Sigma\mathbf{V}^H$  by an *idealized bad channel model* (IBCM)  $\tilde{\mathbf{H}} \triangleq \mathbf{U}\tilde{\Sigma}\mathbf{V}^H$  for which the smallest singular value is zero and the remaining singular values equal the largest singular value of  $\mathbf{H}$ :  $\tilde{\sigma}_{M_T} = 0$  and  $\tilde{\sigma}_1 = \tilde{\sigma}_2 = \dots = \tilde{\sigma}_{M_T-1} = \sigma_1$ . Thus, the singular vectors  $\mathbf{v}_m$  of  $\mathbf{H}$  are maintained but the principal-axis noise variances are modified. By the IBCM approximation, the hyperellipsoids constituting the contour surfaces of the pdf of  $\tilde{\mathbf{w}}$  are approximated by hypercylinders of infinite length whose axis is the dominant noise axis  $\mathbf{v}_{M_T}$ . This is motivated by our desire to make the decision regions invariant to components in the direction of  $\mathbf{v}_{M_T}$ .

We now consider ML detection for the IBCM (to be termed *IML detection*). In the ZF domain, (1) can be equivalently written as

$$\hat{\mathbf{d}}_{\text{ML}} = \arg \min_{\mathbf{d} \in \mathcal{D}} \{(\mathbf{d} - \mathbf{y}_{\text{ZF}})^H \mathbf{H}^H \mathbf{H} (\mathbf{d} - \mathbf{y}_{\text{ZF}})\}. \quad (2)$$

Using the IBCM approximation, we have  $\mathbf{H}^H \mathbf{H} \rightarrow \tilde{\mathbf{H}}^H \tilde{\mathbf{H}} = \sigma_1^2 \mathbf{P}_{M_T}^\perp$ , with  $\mathbf{P}_{M_T}^\perp \triangleq \mathbf{I} - \mathbf{v}_{M_T} \mathbf{v}_{M_T}^H$  denoting the orthogonal projector on the space orthogonal to  $\mathbf{v}_{M_T}$ , and thus (2) becomes

$$\hat{\mathbf{d}}_{\text{IML}} = \arg \min_{\mathbf{d} \in \mathcal{D}} \|\mathbf{P}_{M_T}^\perp (\mathbf{d} - \mathbf{y}_{\text{ZF}})\|^2. \quad (3)$$

For a geometrical interpretation of this IML decision rule, let us define the *reference line*  $\mathcal{L}$  as the straight line that is parallel to the dominant noise axis  $\mathbf{v}_{M_T}$  and whose offset from the origin is  $\mathbf{y}_{\text{ZF}}$ :

$$\mathcal{L}: \mathbf{y}_{\text{ref}}(k) \triangleq k \mathbf{v}_{M_T} + \mathbf{y}_{\text{ZF}}, \quad k \in \mathbb{C}. \quad (4)$$

It can be shown [7] that the norm  $\|\mathbf{P}_{M_T}^\perp (\mathbf{d} - \mathbf{y}_{\text{ZF}})\|$  minimized in (3) is equal to the minimum distance of  $\mathbf{d}$  from  $\mathcal{L}$ :

$$\hat{\mathbf{d}}_{\text{IML}} = \arg \min_{\mathbf{d} \in \mathcal{D}} \|\mathbf{d} - \mathbf{y}_{\mathcal{L}}(\mathbf{d})\|^2, \quad (5)$$

where  $\mathbf{y}_{\mathcal{L}}(\mathbf{d})$  denotes the point of  $\mathcal{L}$  closest to  $\mathbf{d}$ . Thus, IML detection finds the  $\mathbf{d} \in \mathcal{D}$  closest to the reference line  $\mathcal{L}$ .

## 2.3. The Reduced Search Set $\tilde{\mathcal{D}}$

Let  $\mathcal{X}_i$  be the ZF decision region (in the ZF-equalized domain) corresponding to a data vector  $\mathbf{d}^{(i)} \in \mathcal{D}$ . That is, for any  $\mathbf{y} \in \mathcal{X}_i$  the ZF decision is  $\hat{\mathbf{d}}_{\text{ZF}} = \mathbf{d}^{(i)}$ . Because in the ZF-equalized domain the ZF decision is a simple componentwise quantization, every  $\mathbf{y} \in \mathcal{X}_i$  is closer to  $\mathbf{d}^{(i)}$  than to any other data vector  $\mathbf{d}^{(j)} \in \mathcal{D}$ . We shall also say that  $\mathbf{d}^{(i)}$  is “the data vector corresponding to  $\mathcal{X}_i$ .” The following theorem will serve as a basis for the LSD algorithm.

**THEOREM 1.** *The ML detector for the IBCM in (3) is equivalent to minimizing  $\|\mathbf{P}_{M_T}^\perp (\mathbf{d} - \mathbf{y}_{\text{ZF}})\|$  over  $\tilde{\mathcal{D}}$ , where the reduced search set  $\tilde{\mathcal{D}} \subset \mathcal{D}$  is given by all data vectors  $\mathbf{d} \in \mathcal{D}$  that correspond to ZF decision regions pierced by the reference line  $\mathcal{L}$  in (4).*

*Proof.* Assume that  $\hat{\mathbf{d}}_{\text{IML}}$  does not correspond to a ZF decision region pierced by  $\mathcal{L}$ , i.e., no point of  $\mathcal{L}$  lies in the ZF decision region of  $\hat{\mathbf{d}}_{\text{IML}}$ . Hence, in particular,  $\mathbf{y}_{\mathcal{L}}(\hat{\mathbf{d}}_{\text{IML}})$  does not lie in the ZF decision region of  $\hat{\mathbf{d}}_{\text{IML}}$ , and thus it must lie in the ZF decision region of some other data vector  $\mathbf{d}^{(j)} \in \mathcal{D}$ . That is,  $\mathbf{y}_{\mathcal{L}}(\hat{\mathbf{d}}_{\text{IML}})$  is closer to  $\mathbf{d}^{(j)}$  than to  $\hat{\mathbf{d}}_{\text{IML}}$ , i.e.,  $\|\mathbf{d}^{(j)} - \mathbf{y}_{\mathcal{L}}(\hat{\mathbf{d}}_{\text{IML}})\| < \|\hat{\mathbf{d}}_{\text{IML}} - \mathbf{y}_{\mathcal{L}}(\hat{\mathbf{d}}_{\text{IML}})\|$ . For this other data vector  $\mathbf{d}^{(j)}$ , we also have  $\|\mathbf{d}^{(j)} - \mathbf{y}_{\mathcal{L}}(\mathbf{d}^{(j)})\| \leq \|\mathbf{d}^{(j)} - \mathbf{y}_{\mathcal{L}}(\hat{\mathbf{d}}_{\text{IML}})\|$  because  $\mathbf{y}_{\mathcal{L}}(\mathbf{d}^{(j)})$  is the point of  $\mathcal{L}$  closest to  $\mathbf{d}^{(j)}$ , and thus  $\mathbf{y}_{\mathcal{L}}(\hat{\mathbf{d}}_{\text{IML}})$  cannot be closer to  $\mathbf{d}^{(j)}$ . Combining the above two inequalities yields  $\|\mathbf{d}^{(j)} - \mathbf{y}_{\mathcal{L}}(\mathbf{d}^{(j)})\| < \|\hat{\mathbf{d}}_{\text{IML}} - \mathbf{y}_{\mathcal{L}}(\hat{\mathbf{d}}_{\text{IML}})\|$ , which means that  $\mathbf{d}^{(j)}$  is closer to  $\mathcal{L}$  than  $\hat{\mathbf{d}}_{\text{IML}}$  is. But we know from (5) that  $\hat{\mathbf{d}}_{\text{IML}}$  is the data vector closest to  $\mathcal{L}$ . Thus, we have a contradiction, and the proof is complete.  $\square$

In the following, it will be advantageous to represent the complex reference line  $\mathcal{L}: \mathbf{y}_{\text{ref}}(k) = k \mathbf{v}_{M_T} + \mathbf{y}_{\text{ZF}}$ ,  $k \in \mathbb{C}$  as an equivalent real 2-D *reference plane*  $\mathcal{P}: \mathbf{y}_{\text{ref}}(k_R, k_I)$  with  $k_R \triangleq \text{Re}\{k\}$  and  $k_I \triangleq \text{Im}\{k\}$ . Replacing complex scalars by real 2-D vectors, the  $m$ th scalar component of the reference line equation can be written as

$$\mathbf{y}_{\text{ref},m}(\mathbf{k}) = \mathbf{V}_{M_T,m} \mathbf{k} + \mathbf{y}_{\text{ZF},m}, \quad m = 1, \dots, M_T, \quad (6)$$

with  $\mathbf{y}_{\text{ref},m} \triangleq (y_{\text{ref},m,R} \ y_{\text{ref},m,I})^T$ ,  $\mathbf{k} \triangleq (k_R \ k_I)^T$ ,  $\mathbf{y}_{\text{ZF},m} \triangleq (y_{\text{ZF},m,R} \ y_{\text{ZF},m,I})^T$ , and  $\mathbf{V}_{M_T,m} \triangleq \begin{pmatrix} v_{M_T,m,R} & -v_{M_T,m,I} \\ v_{M_T,m,I} & v_{M_T,m,R} \end{pmatrix}$ . To find the reduced

search set  $\tilde{\mathcal{D}}$ , it suffices to specify for each  $\mathbf{d} \in \tilde{\mathcal{D}}$  an arbitrary point  $\mathbf{y}$  of the decision region corresponding to  $\mathbf{d}$ , since  $\mathbf{d} = \mathbf{Q}\{\mathbf{y}\}$ . The LSD is based on the fact (cf. Theorem 1) that these points can be found in  $\mathcal{P}$ , and thus the search for  $\tilde{\mathcal{D}}$  can be restricted to  $\mathcal{P}$ .

## 2.4. The Partitioning of the Reference Plane

The intersection of the ZF decision regions with the reference plane  $\mathcal{P}$  induces a partitioning of  $\mathcal{P}$ . We will now show that for a “line-structured” symbol alphabet  $\mathcal{A}$ —which is such that the boundaries of the quantization regions in the 2-D real symbol domain are straight lines—this partitioning can easily be calculated. Examples of line-structured alphabets are ASK, QAM, and PSK but not, e.g., an hexagonal constellation.

Let  $P$  denote the number of boundary lines in the 2-D real symbol domain. The  $p$ th real boundary line can be written as

$$\mathbf{b}^{(p)}(k_b) = k_b \mathbf{u}^{(p)} + \mathbf{o}^{(p)}, \quad p = 1, \dots, P, \quad k_b \in \mathbb{R}. \quad (7)$$

For example, the 4-QAM alphabet  $\mathcal{A} = \{1 + j, -1 + j, -1 - j, 1 - j\}$  has  $P = 2$  orthogonal boundary lines defined by  $\mathbf{u}^{(1)} = (0 \ 1)^T$ ,  $\mathbf{u}^{(2)} = (1 \ 0)^T$ , and  $\mathbf{o}^{(1)} = \mathbf{o}^{(2)} = (0 \ 0)^T$ .

The ZF decision regions are bounded by  $M_T P$  hyperplanes. The  $(m, p)$ th hyperplane is obtained by setting the  $m$ th component of a vector  $\mathbf{y}$  equal to the  $p$ th boundary line (7). To calculate the partitioning of  $\mathcal{P}$ , we must thus calculate the intersection of  $\mathcal{P}$  with all boundary hyperplanes. The intersection of  $\mathcal{P}$  with the  $(m, p)$ th boundary hyperplane yields a straight boundary line  $\mathcal{B}^{(m,p)} \subset \mathcal{P}$  that can be calculated by equating (7) and (6), i.e.,  $\mathbf{y}_{\text{ref},m} = \mathbf{b}^{(p)}$ . This gives  $\mathbf{V}_{M_T,m} \mathbf{k} + \mathbf{y}_{\text{ZF},m} = k_b \mathbf{u}^{(p)} + \mathbf{o}^{(p)}$  or, equivalently,

$$\mathcal{B}^{(m,p)} : \mathbf{k} = \mathbf{k}^{(m,p)}(k_b) = k_b \boldsymbol{\alpha}^{(m,p)} + \boldsymbol{\beta}^{(m,p)}, \quad (8)$$

with  $\boldsymbol{\alpha}^{(m,p)} \triangleq \frac{1}{\rho_m} \mathbf{V}_{M_T,m}^T \mathbf{u}^{(p)}$  and  $\boldsymbol{\beta}^{(m,p)} \triangleq \frac{1}{\rho_m} \mathbf{V}_{M_T,m}^T (\mathbf{o}^{(p)} - \mathbf{y}_{\text{ZF},m})$ , where  $\rho_m \triangleq \sqrt{v_{M_T,m,R}^2 + v_{M_T,m,I}^2}$ . Because  $\mathbf{V}_{M_T,m}$  is an orthogonal matrix up to a factor, the angles between the  $P$  boundary lines corresponding to the  $m$ th component are equal to the angles between the boundary lines  $\mathbf{b}^{(p)}$  in the symbol alphabet domain.

The  $M_T P$  boundary lines  $\mathcal{B}^{(m,p)}$  partition the reference plane  $\mathcal{P}$  into elementary cells  $\mathcal{C} \subset \mathcal{P}$ . Each cell  $\mathcal{C}$  is the intersection of a ZF decision region with  $\mathcal{P}$ . Thus, any point  $\mathbf{y}$  of a given cell  $\mathcal{C}$  corresponds to a vector  $\mathbf{y}_{\text{ref}}(k)$  for which ZF decision (quantization) leads to the same  $\hat{\mathbf{d}}_{\text{ZF}}$ , i.e.,  $\hat{\mathbf{d}}_{\text{ZF}} = Q\{\mathbf{y}\}$  for all  $\mathbf{y} \in \mathcal{C}$ . This  $\hat{\mathbf{d}}_{\text{ZF}}$  belongs to the reduced search set  $\tilde{\mathcal{D}}$  because it corresponds to a ZF decision region sliced by  $\mathcal{P}$  (cf. Theorem 1). Therefore, any collection of arbitrary cell points—one point for each cell—defines the reduced search set  $\tilde{\mathcal{D}}$ . Note that also  $Q\{\mathbf{y}_{\text{ZF}}\}$  belongs to  $\tilde{\mathcal{D}}$ .

The size  $|\tilde{\mathcal{D}}|$  of the reduced search set equals the number of cells  $\mathcal{C}$ . If  $i-1$  boundary lines are given, it is easy to verify that each additional boundary line yields at most  $i$  new cells (it will yield  $i$  new cells if and only if it intersects at different points with all  $i-1$  previous lines). Thus, an upper bound on  $|\tilde{\mathcal{D}}|$  is obtained as

$$|\tilde{\mathcal{D}}| \leq |\tilde{\mathcal{D}}|_{\max} = 1 + \sum_{i=1}^{M_T P} i = \frac{(M_T P)^2}{2} + \frac{M_T P}{2} + 1.$$

For example,  $|\tilde{\mathcal{D}}| = |\tilde{\mathcal{D}}|_{\max}$  for 4-QAM since there are no parallel boundary lines and only two boundary lines intersect in an intersection point. Comparing  $|\tilde{\mathcal{D}}|_{\max}$  with  $|\mathcal{D}| = |\mathcal{A}|^{M_T}$ , we see that for reasonably high data rates  $|\tilde{\mathcal{D}}| \ll |\mathcal{D}|$  and thus we obtain a significant reduction of complexity. For example, for an (8,8) channel and 4-QAM modulation we obtain  $|\tilde{\mathcal{D}}| = |\tilde{\mathcal{D}}|_{\max} = 137$  pierced decision regions out of a total of  $|\mathcal{D}| = 4^8 = 65536$  decision regions.

### 3. THE LINE-SEARCH DETECTOR

The proposed LSD minimizes the ML distance  $\|\mathbf{r} - \mathbf{H}\mathbf{d}\|$  over the reduced search set  $\tilde{\mathcal{D}}$ . Note that we do not minimize  $\|\mathbf{P}_{M_T}^{\perp}(\mathbf{d} - \mathbf{y}_{\text{ZF}})\|$ , because actual channels will not conform to the IBCM. The IBCM is only used to motivate the definition of  $\tilde{\mathcal{D}}$ .

#### 3.1. Efficient Determination of $\tilde{\mathcal{D}}$

According to the previous section, the reduced search set  $\tilde{\mathcal{D}}$  can be determined by finding an arbitrary point  $\mathbf{k}$  of each cell  $\mathcal{C} \subset \mathcal{P}$ . This can be done efficiently by searching the reference plane along each boundary line. To search along the  $(m, p)$ th boundary line  $\mathcal{B}^{(m,p)}$ , we first calculate the intersection points  $\mathbf{k}_{\text{int}}^{(m,p,m',p')}$  of  $\mathcal{B}^{(m,p)}$  with all  $\mathcal{B}^{(m',p')}$ ,  $(m', p') \neq (m, p)$  that are not parallel to  $\mathcal{B}^{(m,p)}$  [7]. We then proceed as illustrated in Fig. 2. Based on the first intersection point  $\mathbf{k}_{\text{int}}^{(m,p;m_1,p_1)}$ , the first cell point  $\mathbf{k}_{\text{start}}$  and the associated first data vector  $\mathbf{d}^{(1)} = Q\{k_{\text{start}} \mathbf{v}_{M_T} + \mathbf{y}_{\text{ZF}}\} \in \tilde{\mathcal{D}}$  are calculated. All remaining data vectors  $\mathbf{d}^{(i)} \in \tilde{\mathcal{D}}$  associated with  $\mathcal{B}^{(m,p)}$  are uniquely

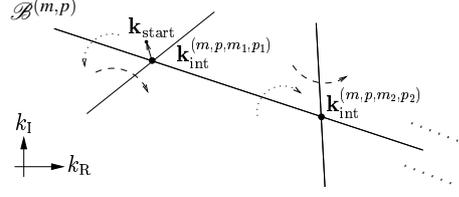


Figure 2: Determination of  $\tilde{\mathcal{D}}$ .

determined by the remaining intersection points.<sup>1</sup> To find them, we perform two hops at each intersection point. The first hop (dotted arrow in Fig. 2) is over the current search line  $\mathcal{B}^{(m,p)}$ ; this corresponds to an update of the  $m$ th data vector component across the  $p$ th boundary. The second hop (dashed arrow in Fig. 2) is over the intersecting boundary line  $\mathcal{B}^{(m',p')}$ ; this corresponds to an update of the  $m'$ th data vector component across the  $p'$ th boundary.

These data vector component updates can be performed without calculating new  $\mathbf{k}$  or  $\mathbf{y}$  vectors. Suppose we just obtained a specific  $\mathbf{d}^{(i)} \in \tilde{\mathcal{D}}$  and wish to determine the next  $\mathbf{d}^{(i+1)} \in \tilde{\mathcal{D}}$  corresponding to the cell we would reach by hopping over the intersecting boundary line  $\mathcal{B}^{(m',p')}$ . Now  $\mathcal{B}^{(m',p')}$  corresponds to the  $m'$ th component of  $\mathbf{d}^{(i)}$  and the  $p'$ th boundary line in the symbol alphabet domain. By hopping over  $\mathcal{B}^{(m',p')}$ , we thus move from  $\mathbf{d}^{(i)}$  to  $\mathbf{d}^{(i+1)} = \mathbf{d}^{(i)} + \Delta \mathbf{d}^{(i)}$ , where  $\Delta \mathbf{d}^{(i)}$  is zero except for the  $m'$ th component  $(\Delta \mathbf{d}^{(i)})_{m'}$  that depends on the  $m'$ th component of  $\mathbf{d}^{(i)}$  and on the  $p'$ th boundary. For example, for 4-QAM we have

$$(\Delta \mathbf{d}^{(i)})_{m'} = \begin{cases} -2j \operatorname{Im}\{(\mathbf{d}^{(i)})_{m'}\} & \text{if } p' = 1, \\ -2 \operatorname{Re}\{(\mathbf{d}^{(i)})_{m'}\} & \text{if } p' = 2, \end{cases}$$

and thus the update simply amounts to flipping the imaginary part ( $p' = 1$ ) or real part ( $p' = 2$ ) of the  $m'$ th data vector component.

#### 3.2. Efficient Determination of Distances

The squared ML distances  $\psi_i^2 \triangleq \|\mathbf{r} - \mathbf{H}\mathbf{d}^{(i)}\|^2$ ,  $\mathbf{d}^{(i)} \in \tilde{\mathcal{D}}$  can be calculated recursively. Again, we move along  $\mathcal{B}^{(m,p)}$ . Suppose that  $\psi_i^2$  has already been determined, and that the next data vector  $\mathbf{d}^{(i+1)}$  corresponds to a hop over the intersecting boundary line  $\mathcal{B}^{(m',p')}$ . As was shown in the previous subsection,  $\mathbf{d}^{(i+1)}$  is a neighbor of  $\mathbf{d}^{(i)}$  that differs from  $\mathbf{d}^{(i)}$  only in the  $m'$ th component, namely, by  $(\Delta \mathbf{d}^{(i)})_{m'}$ . Thus, the distances can be updated as  $\psi_{i+1}^2 = \psi_i^2 + \Delta_i$ , where it can be shown that

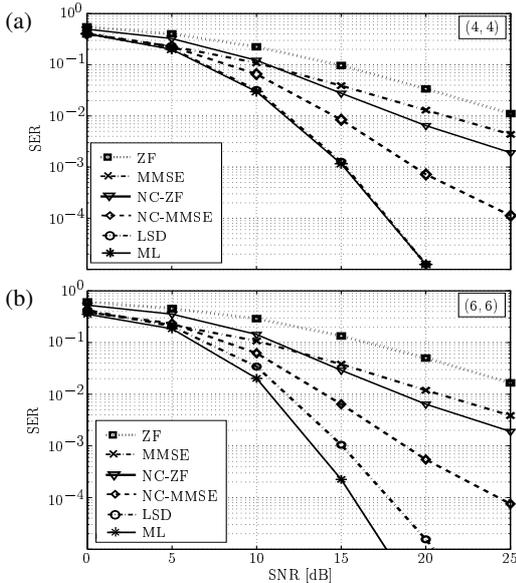
$$\Delta_i = |(\Delta \mathbf{d}^{(i)})_{m'}|^2 \|\mathbf{h}_m\|^2 - 2 \operatorname{Re}\{(\Delta \mathbf{d}^{(i)})_{m'} (\mathbf{r} - \mathbf{H}\mathbf{d}^{(i)})^H \mathbf{h}_m\},$$

with  $\mathbf{h}_m$  denoting the  $m$ th column of  $\mathbf{H}$ . Here, the vector  $\mathbf{r} - \mathbf{H}\mathbf{d}^{(i)}$  can be calculated recursively, too, and the  $\|\mathbf{h}_m\|^2$  can be precalculated. This recursion is initialized by calculating  $\psi_1^2 = \|\mathbf{r} - \mathbf{H}\mathbf{d}^{(1)}\|^2$  in a straightforward manner.

#### 3.3. Algorithm Summary and Computational Complexity

We will now summarize the LSD and discuss how the LSD's complexity depends on the parameters  $|\mathcal{A}|$  and  $M_T$  that determine the transmission rate. We assume  $M_T = M_R$  for simplicity.

<sup>1</sup>For simplicity, we assume that the intersection points of any two boundary lines are different. This holds for arbitrary QAM constellations; however, it does not hold for PSK constellations with  $|\mathcal{A}| > 4$  because here all the  $P$  boundary lines in the symbol plane intersect at the origin, and this geometry is maintained in  $\mathcal{P}$ . However, these points are known *a priori* and the following algorithm can easily be extended to this situation.



**Figure 3:** SER vs. SNR performance of the LSD and standard detectors using 4-QAM modulation: (a) (4,4) channel, (b) (6,6) channel.

The LSD comprises the following preparatory steps that are performed once for each data block over which the channel is assumed constant: (i) Calculation of  $(\mathbf{H}^H \mathbf{H})^{-1}$  and  $\mathbf{y}_{ZF}$  (complexity  $\mathcal{O}(M_T^3)$ ); (ii) calculation of  $\mathbf{v}_{M_T}$  (complexity  $\mathcal{O}(M_T^2)$  if the power method [6] is used); and (iii) calculation of  $\|\mathbf{h}_m\|^2$  for  $m = 1, \dots, M_T$  (complexity  $\mathcal{O}(M_T^2)$ ). Thus, the dominant complexity of the preparatory steps is  $\mathcal{O}(M_T^3)$  per data block.

Additionally, the following steps are performed for each transmitted data vector: (i) Determination of all  $M_T P$  boundary lines in  $\mathcal{P}$  and their intersection points (complexity  $\mathcal{O}(M_T^2 P^2)$ ); (ii) calculation of  $M_T P$  initial data vectors and their associated distances (complexity  $\mathcal{O}(M_T^3 P)$ ); and (iii) calculation of roughly  $2M_T P (M_T P - 1)$  distance updates (complexity  $\mathcal{O}(M_T^3 P^2)$ ). Hence, the overall complexity of the LSD is  $\mathcal{O}(M_T^3 P^2)$  per transmitted data vector. This is cubic in the number of transmit antennas  $M_T$  and, since usually  $P \propto |\mathcal{A}|$ , quadratic in the symbol alphabet size  $|\mathcal{A}|$ .

#### 4. SIMULATION RESULTS

In our simulations, we used 4-QAM modulation and MIMO channels with iid Gaussian entries of unit variance. The dominant eigenvector  $\mathbf{v}_{M_T}$  was computed by means of the power method [6] with 4 iterations. For NC detection we used optimal layer ordering according to [1, 9].

**SER PERFORMANCE.** Fig. 3 shows the symbol-error rate (SER) versus the SNR for the LSD and standard detectors<sup>2</sup>. We considered a (4,4) channel and a (6,6) channel. It is seen that the LSD performs substantially better than the standard suboptimal detectors and achieves higher diversity orders. Specifically, the LSD performs as well as ML detection for the (4,4) channel (the corresponding SER curves in Fig. 3(a) are nearly indistinguishable), and close to ML detection for the (6,6) channel. This shows that our strategy for constructing the reduced search set  $\tilde{\mathcal{S}}$  works very well.

**COMPUTATIONAL COMPLEXITY.** To convey a rough picture of the computational complexity of the LSD, Table 1 displays kflor

<sup>2</sup>The SNR is defined as  $E\{\|\mathbf{H}\mathbf{d}\|^2\}/E\{\|\mathbf{w}\|^2\} = M_T \sigma_d^2 / \sigma_w^2$ .

(a) kflors – block operations			(b) kflors – vector operations		
channel	SDML		SDML		LSD
	av.	LSD	av.	max.	
(4,4)	2.1	2.8	32	82	13
(6,6)	6.4	7.8	68	300	41
(8,8)	14.2	16.8	122	671	93

**Table 1:** Measured complexity in kflors for (a) operations performed once per data block and (b) operations performed once per data vector.

estimates (determined by using MATLAB V5.3) for the LSD and for the sphere-decoding algorithm for ML detection [4] (abbreviated SDML). We used three different channels ((4,4), (6,6), and (8,8)) and 4-QAM modulation. Table 1(a) shows the complexity of the operations performed once for an entire data block during which the channel is constant (e.g., calculation of  $(\mathbf{H}^H \mathbf{H})^{-1} \mathbf{H}^H$  and  $\mathbf{v}_{M_T}$ ). Table 1(b) shows the complexity of the operations performed once for each data vector (e.g., determination of  $\tilde{\mathcal{S}}$  and  $\psi_i^2$ ). The computational complexity of SDML strongly depends on the actual channel realization and the SNR. Therefore, in addition to the average complexity of SDML, Table 1(b) shows the maximum complexity obtained during 10000 simulation runs at an SNR of 10dB. (Note that the complexity of the LSD is independent of the channel realization and the SNR.) From Table 1, it can be seen that the overall complexity of the LSD is smaller than the average complexity of SDML and only a fraction of the maximum complexity of SDML.

#### 5. SUMMARY AND CONCLUSIONS

We have developed the *line-search detector* (LSD) for spatial multiplexing MIMO systems. The LSD minimizes the ML distance criterion over a reduced data vector set that is constructed by a 1-D nearest neighbor search along the least significant right singular vector of the channel matrix. This was motivated by an idealized model of “bad” channel realizations and by a desire to achieve robustness to bad channels. We also proposed efficient algorithms for calculating the reduced data vector set and the associated ML distances. Simulation results demonstrated that the LSD can yield near-ML performance and significantly increased diversity gains at a computational complexity that is much smaller than that of the sphere-decoding algorithm for ML detection.

#### REFERENCES

- [1] G. D. Golden, C. J. Foschini, R. A. Valenzuela, and P. W. Wolniansky, “Detection algorithm and initial laboratory results using V-BLAST space-time communications architecture,” *Elect. Lett.*, vol. 35, pp. 14–16, Jan. 1999.
- [2] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. Englewood Cliffs (NJ): Prentice Hall, 1993.
- [3] S. M. Kay, *Fundamentals of Statistical Signal Processing: Detection Theory*. Upper Saddle River (NJ): Prentice Hall, 1998.
- [4] U. Fincke and M. Phost, “Improved methods for calculating vectors of short length in a lattice, including a complexity analysis,” *Math. Comp.*, vol. 44, pp. 463–471, April 1985.
- [5] B. Hassibi and H. Vikalo, “On the expected complexity of sphere decoding,” in *Proc. 35th Asilomar Conf. Signals, Systems, Computers*, (Pacific Grove, CA), pp. 1051–1055, Nov. 2001.
- [6] G. H. Golub and C. F. Van Loan, *Matrix Computations*. Baltimore: Johns Hopkins University Press, 3rd ed., 1996.
- [7] H. Artés, D. Seethaler, and F. Hlawatsch, “Efficient detection algorithms for MIMO channels: A geometrical approach to approximate ML detection,” *IEEE Trans. Signal Processing, Special Issue on MIMO Communications Systems*, vol. 51, pp. 2808–2820, Nov. 2003.
- [8] C. W. Therrien, *Discrete Random Signals and Statistical Signal Processing*. Englewood Cliffs (NJ): Prentice Hall, 1992.
- [9] M. Rupp, M. Guillaud, and S. Das, “On MIMO decoding algorithms for UMTS,” in *Proc. 35th Asilomar Conf. Signals, Systems, Computers*, vol. 2, (Pacific Grove, CA), pp. 975–979, Nov. 2001.