

Efficient Detection Algorithms for MIMO Channels: A Geometrical Approach to Approximate ML Detection

Harold Artés, *Student Member, IEEE*, Dominik Seethaler, *Student Member, IEEE*, and Franz Hlawatsch, *Senior Member, IEEE*

Abstract—It is well known that suboptimal detection schemes for multiple-input multiple-output (MIMO) spatial multiplexing systems (equalization-based schemes as well as nulling-and-cancelling schemes) are unable to exploit all of the available diversity, and thus, their performance is inferior to ML detection. Motivated by experimental evidence that this inferior performance is primarily caused by the inability of suboptimal schemes to deal with “bad” (i.e., poorly conditioned) channel realizations, we study the decision regions of suboptimal schemes for bad channels. Based on a simplified model for bad channels, we then develop two computationally efficient detection algorithms that are robust to bad channels. In particular, the novel *sphere-projection algorithm* (SPA) is a simple add-on to standard suboptimal detectors that is able to achieve near-ML performance and significantly increased diversity gains. The SPA’s computational complexity is comparable with that of nulling-and-cancelling detectors and only a fraction of that of the Fincke–Phost sphere-decoding algorithm for ML detection.

Index Terms—Equalization, maximum likelihood detection, MIMO channels, nulling and cancelling, spatial multiplexing, sphere decoding, V-BLAST.

I. INTRODUCTION

MULTIPLE-INPUT multiple-output (MIMO) fading channels offer not only the potential of high data rates but also the promise of high reliability due to their inherently available diversity (e.g., [1]). It is well known that this available diversity cannot be fully exploited by suboptimal equalization-based detection schemes (e.g., [2]), including nulling-and-cancelling schemes [3]. Maximum-likelihood (ML) detection can exploit all of the available diversity but tends to be computationally intensive. This is also true for the Fincke–Phost sphere-decoding ML algorithm [4], as will be illustrated in Section VII. Thus, there is a strong demand for computationally efficient suboptimal detection algorithms that can exploit a large part of the available diversity.

In this paper, we develop such improved detection algorithms for spatial multiplexing systems such as V-BLAST [2]. Our starting point is to examine why the suboptimal detection

schemes fail to exploit all of the available diversity. One explanation is that suboptimal detection “uses up” the degrees of freedom that would otherwise offer diversity (e.g., [5]). Here, we explain the inferior performance of suboptimal detection schemes by comparing the decision regions of these schemes to the decision regions of the ML detector. The “improper” decision regions of suboptimal schemes are no problem for channel realizations with a condition number near to 1. (Note that for condition number 1, zero-forcing equalization followed by componentwise quantization is equivalent to ML detection.) However, for channel realizations with a large condition number, they cause a significant performance degradation. In fact, it turns out that these “bad”¹ channel realizations with large condition number are to a great extent responsible for the inferior average performance of suboptimal detection.

Motivated by this insight, we introduce an idealized model for bad channels that allows a substantially simplified implementation of ML detection. This efficient detection algorithm is then extended to nonidealized channels. A final modification and simplification yields the novel *sphere-projection algorithm* (SPA). The SPA is an efficient nonlinear add-on to standard suboptimal detection schemes that makes these schemes robust to bad channels. For spatial multiplexing systems of practical interest (e.g., six transmit antennas and six receive antennas), the detection schemes obtained by this approach are demonstrated to yield excellent performance at low computational complexity.

Our paper is organized as follows. In the remainder of this section, we describe the system model and give an overview of existing detection schemes for spatial multiplexing systems. In Section II, the effects of bad channels on equalization-based detection schemes are discussed. An idealized model for bad channels is introduced in Section III, and an efficient ML detection algorithm for this model is developed in Section IV. In Section V, this detection algorithm is extended to arbitrary MIMO channels (for which it will no longer be ML). In Section VI, a simplification of this latter detection algorithm yields the SPA. Finally, simulation results presented in Section VII show that the proposed algorithms can yield near-ML performance at just a fraction of the computational cost of the Fincke–Phost sphere-decoding algorithm.

Manuscript received December 20 2002; revised April 22, 2003. This work was supported by the FWF under Grant P15156. The associate editor coordinating the review of this paper and approving it for publication was Dr. Rick S. Blum.

The authors are with the Institute of Communications and Radio-Frequency Engineering, Vienna University of Technology, A-1040 Wien, Austria (e-mail: hartes@aurora.nt.tuwien.ac.at).

Digital Object Identifier 10.1109/TSP.2003.818210

¹The term “bad” refers to the poor performance of suboptimal detection schemes for spatial multiplexing systems. Note, however, that these channels are not necessarily “bad” in the sense of, e.g., low channel capacity.

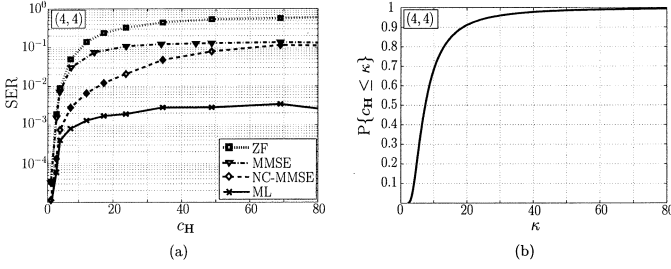


Fig. 1. Detector performance and channel condition number for a (4,4), iid Gaussian channel. (a) SER performance of various detection schemes versus condition number of the respective channel realization for QPSK modulation and an SNR of 15 dB. (ZF is ZF equalization based detection, MMSE is MMSE equalization based detection, NC-MMSE is nulling and cancelling using MMSE equalization, and ML is ML detection.) (b) Cumulative distribution function of the condition number.

A. System Model

We consider a MIMO channel with M_T transmit antennas and $M_R \geq M_T$ receive antennas [this will be briefly termed an (M_T, M_R) channel]. This channel is part of a spatial multiplexing system such as V-BLAST [2], where the m th data stream (or *layer*) $d_m[n]$ is directly transmitted on the m th transmit antenna. This leads to the well-known baseband model

$$\mathbf{r}[n] = \mathbf{H}\mathbf{d}[n] + \mathbf{w}[n] \quad (1)$$

with the $M_T \times 1$ transmit vector $\mathbf{d}[n] \triangleq (d_1[n] \ d_2[n] \ \cdots \ d_{M_T}[n])^T$, the $M_R \times M_T$ channel matrix \mathbf{H} , the $M_R \times 1$ received vector $\mathbf{r}[n] \triangleq (r_1[n] \ r_2[n] \ \cdots \ r_{M_R}[n])^T$, and the $M_R \times 1$ noise vector $\mathbf{w}[n] \triangleq (w_1[n] \ w_2[n] \ \cdots \ w_{M_R}[n])^T$. The data streams $d_m[n]$ are assumed zero-mean, uncorrelated, stationary, white random processes with variance σ_d^2 . The channel \mathbf{H} is considered constant over a block of consecutive time instants and perfectly known at the receiver. The noise processes $w_m[n]$ are statistically independent, zero-mean, circularly symmetric complex-valued, Gaussian, stationary, white random processes with variance σ_w^2 . For simplicity, we will hereafter omit the time index n and, thus, write (1) as $\mathbf{r} = \mathbf{H}\mathbf{d} + \mathbf{w}$.

B. Review of Detection Schemes for Spatial Multiplexing Systems

The major detection approaches for spatial multiplexing systems are linear equalization followed by quantization (e.g., [2]), nulling and cancelling (or decision-feedback) [2], and ML detection [4], [6]. The first two approaches are suboptimal but have significantly lower computational complexity than ML detection. We will briefly review these three detection schemes.

Linear Equalization Based Detection: In linear equalization-based detection, an estimate of the transmitted data vector \mathbf{d} is calculated as $\mathbf{y} = \mathbf{G}\mathbf{r}$, where \mathbf{G} is an equalization matrix. The detected data vector is then obtained as $\hat{\mathbf{d}} = Q\{\mathbf{y}\}$, where $Q\{\cdot\}$ denotes componentwise quantization according to the symbol alphabet used. (Componentwise quantization is suboptimal in general because the multiplication by \mathbf{G} introduces correlations of the noise components.) The *zero-forcing* (ZF) equalizer is given by the pseudo-inverse [7] of \mathbf{H} , i.e., $\mathbf{G} = \mathbf{H}^\# = (\mathbf{H}^H\mathbf{H})^{-1}\mathbf{H}^H$. (For the last expression, we

assumed that $M_R \geq M_T$ and that \mathbf{H} has full rank.) Thus, the result of ZF equalization (before quantization) is

$$\mathbf{y}_{\text{ZF}} = \mathbf{H}^\# \mathbf{r} = (\mathbf{H}^H\mathbf{H})^{-1}\mathbf{H}^H \mathbf{r} = \mathbf{d} + \tilde{\mathbf{w}}$$

which is the transmitted data vector \mathbf{d} corrupted by the transformed noise $\tilde{\mathbf{w}} = \mathbf{H}^\# \mathbf{w}$. The *minimum mean-square error* (MMSE) equalizer is given by [8] $\mathbf{G} = (\mathbf{H}^H\mathbf{H} + (\sigma_w^2/\sigma_d^2)\mathbf{I})^{-1}\mathbf{H}^H$, which minimizes the mean-square error $E\{\|\mathbf{y} - \mathbf{d}\|^2\}$. Thus, the result of MMSE equalization is

$$\mathbf{y}_{\text{MMSE}} = \left(\mathbf{H}^H\mathbf{H} + \frac{\sigma_w^2}{\sigma_d^2} \mathbf{I} \right)^{-1} \mathbf{H}^H \mathbf{r}.$$

Nulling and Cancelling: In contrast to linear data detection, where all layers are detected jointly, nulling and cancelling (NC) uses a serial decision-feedback approach to detect each layer separately (e.g., [2]). When a layer has been detected, an estimate of the corresponding contribution to the received vector \mathbf{r} is subtracted from \mathbf{r} ; the result is then used to detect the next layer, etc. In the absence of detection errors, NC progressively cleans \mathbf{r} from the interference corresponding to the layers already detected. To detect a specific layer, the layers that have not been detected yet are “nulled out” (equalized) according to the ZF or MMSE approach described above. Error propagation can be a problem because incorrect data decisions actually increase the interference when detecting subsequent layers. Thus, the order in which the layers are detected strongly influences the performance of NC.

ML Detection: ML detection is optimal in the sense of minimum error probability when all data vectors are equally likely. For our system model (1) and our assumptions, the ML detector is given by

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

Here, \mathcal{D} denotes the set of all possible transmitted data vectors $\mathbf{d}^{(i)}$, $i = 1, \dots, |\mathcal{D}|$. We note that $\mathcal{D} = \mathcal{A}^{M_T}$, where \mathcal{A} is the elementary symbol alphabet, and thus, $|\mathcal{D}| = |\mathcal{A}|^{M_T}$. In general, the computational complexity of ML detection grows exponentially with M_T . Using the Finkelstein–Phost sphere-decoding algorithm [4], ML detection can be achieved at an *average* complexity that increases with M_T roughly as $\mathcal{O}(M_T^3)$, as was shown in [9]. However, for bad channel realizations, the complexity of sphere-decoding can be very high (see Section VII-C).

It can be shown [4] that the ML decision rule (2) can be rewritten as

$$\hat{\mathbf{d}}_{\text{ML}} = \arg \min_{\mathbf{d} \in \mathcal{D}} \{(\mathbf{d} - \mathbf{y}_{\text{ZF}})^H \mathbf{H}^H \mathbf{H} (\mathbf{d} - \mathbf{y}_{\text{ZF}})\}. \quad (3)$$

This can be interpreted as the ML detector for an *identity* channel corrupted by zero-mean Gaussian noise $\tilde{\mathbf{w}} = \mathbf{H}^\# \mathbf{w}$ (the noise contained in $\mathbf{y}_{\text{ZF}} = \mathbf{d} + \tilde{\mathbf{w}}$) with covariance matrix

$$\mathbf{R}_{\tilde{\mathbf{w}}} = \sigma_w^2 (\mathbf{H}^H \mathbf{H})^{-1}. \quad (4)$$

That is, after ZF equalization the noise is generally *correlated*. We may view (3) as “ML detection after ZF equalization” or

“ML detection in the ZF-equalized domain,” as opposed to “direct ML detection” according to (2). Note, however, that (2) and (3) are strictly equivalent.

II. “BAD CHANNEL” EFFECT

In this section, we will demonstrate that the inferior performance of suboptimal detection compared with ML detection is mainly caused by the occurrence of bad channel realizations.

A. Detector Performance and Bad Channels

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 σ_m of \mathbf{H} and the matrices \mathbf{U} and \mathbf{V} contain, respectively, the left and right singular vectors of \mathbf{H} as columns [7]. We assume that the σ_m are indexed in nonincreasing order. The *condition number* $c_H = \sigma_1/\sigma_{M_T} \geq 1$ is the ratio of the largest to smallest singular value. For a bad (poorly conditioned) channel, c_H is large.

Experiments suggest that the performance of suboptimal detection schemes strongly depends on the channel’s condition number. In Fig. 1(a), we show the symbol error rate (SER) of various detection schemes versus the condition number of the channel realization. In this simulation, we used a (4,4) channel with independent and identically distributed (iid) Gaussian channel matrix entries, QPSK modulation, and an SNR of 15 dB. It can be seen that there is a significant performance gap between linear (i.e., ZF or MMSE) detection and ML detection for c_H about 4 or larger and between NC and ML detection for c_H about 20 or larger.

The impact of this behavior on the *average* SER performance of suboptimal detection of course depends on the probability with which bad channels occur. In Fig. 1(b), we show the cumulative distribution function (cdf) of c_H estimated in the course of the simulation described above. It is seen that the probability that c_H exceeds a value of 10, 15, and 20 is 32%, 15%, and 9%, respectively. This suggests that bad channels occur frequently enough to cause a significant degradation of the average performance of suboptimal detection schemes. Theoretical investigations show that for an increasing number of transmit and receive antennas, the cdf is essentially scaled (expanded) in the c_H direction [10]. We can thus expect the bad channel effect to become even worse for an increasing number of antennas.

While the performance of ML detection is fairly robust to bad channel realizations, it is noteworthy that the computational complexity of the Finke–Phost sphere-decoding algorithm for ML detection significantly increases for bad channels [4], [11]. Thus, there is a strong demand for computationally efficient suboptimal detectors that are able to achieve near-ML performance. The detection methods presented in this paper are designed to satisfy this demand.

B. Geometrical Analysis

The starting point for developing these improved detection methods is a geometrical analysis of the decision regions of suboptimal detection methods in the case of bad channels.

ZF Detection: We first consider linear detection based on ZF equalization. For a perfectly conditioned channel, i.e., $c_H = 1$,

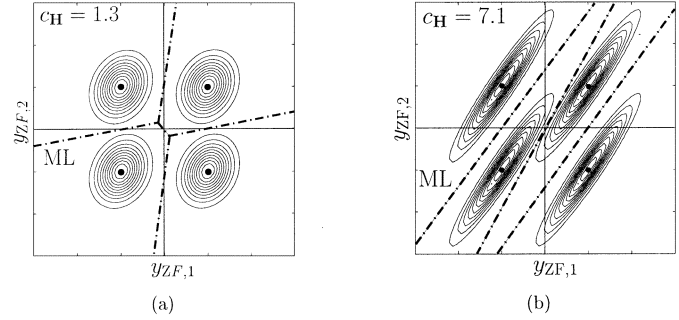


Fig. 2. Probability density function of \mathbf{y}_{ZF} and ZF and ML decision regions in the ZF-equalized domain for a real-valued (2,2) channel and BPSK modulation. (a) “Good” channel realization with condition number 1.3. (b) “Bad” channel realization with condition number 7.1. The ZF decision regions are the four quadrants; the ML decision regions are indicated by dash-dotted lines.

we have $\mathbf{H}^H\mathbf{H} \propto \mathbf{I}$. Here, (4) shows that the components of the noise vector $\tilde{\mathbf{w}} = \mathbf{H}^\# \mathbf{w}$ are uncorrelated; furthermore, the ML detector in (3) simplifies to $\hat{\mathbf{d}}_{ML} = \arg \min_{\mathbf{d} \in \mathcal{D}} \{\|\mathbf{d} - \mathbf{y}_{ZF}\|^2\}$ and thus becomes equal to the ZF detector solution $\hat{\mathbf{d}}_{ZF} = Q\{\mathbf{y}_{ZF}\}$. On the other hand, for a poorly conditioned channel, $\mathbf{H}^H\mathbf{H}$ is quite different from being proportional to \mathbf{I} . Thus, the components of $\tilde{\mathbf{w}}$ are generally correlated, and the ZF solution $\hat{\mathbf{d}}_{ZF} = Q\{\mathbf{y}_{ZF}\}$ must be expected to be far away from the optimal ML solution² $\hat{\mathbf{d}}_{ML}$.

For a geometrical analysis, we consider the probability density function (pdf) of the ZF-filtered Gaussian noise vector $\tilde{\mathbf{w}}$. The contour surfaces of this pdf are hyperellipsoids [12]. Using $\mathbf{H} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^H$, the covariance matrix $\mathbf{R}_{\tilde{\mathbf{w}}} = \sigma_w^2(\mathbf{H}^H\mathbf{H})^{-1}$ can be written as

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

The m th principal axis of the hyperellipsoids is such that its direction is given by the m th eigenvector \mathbf{v}_m of $\mathbf{R}_{\tilde{\mathbf{w}}}$, which is equal to the m th column of \mathbf{V} , and its length is proportional to the square root of the m th eigenvalue of $\mathbf{R}_{\tilde{\mathbf{w}}}$ [12], which is equal to

$$\sigma_{\tilde{w},m} = \frac{\sigma_w}{\sigma_m}. \quad (5)$$

Thus, ZF equalization results in a *distortion* of the noise pdf with respect to the spherical geometry of the pdf of the original noise vector \mathbf{w} .

For illustration, Fig. 2 shows the pdf of the received vector after ZF equalization \mathbf{y}_{ZF} for two different realizations of a real-valued (2,2) channel with condition numbers 1.3 and 7.1. The modulation format is BPSK. This figure also shows the ZF decision regions (the four quadrants) and the ML decision regions (indicated by dash-dotted lines). Whereas the ZF and ML decision regions are similar for the “good” channel with condition number 1.3, they are dramatically different for the “bad” channel with condition number 7.1. Indeed, in the latter case,

²Note that even when $c_H > 1$, it is possible that \mathbf{H} has orthogonal columns but with different norms. In this case, $\mathbf{H}^H\mathbf{H}$ is diagonal, which means that the components of $\tilde{\mathbf{w}}$ are still uncorrelated, and ZF detection will still be optimal. However, simulations show that for poorly conditioned channel realizations, a situation close to this case is very unlikely.

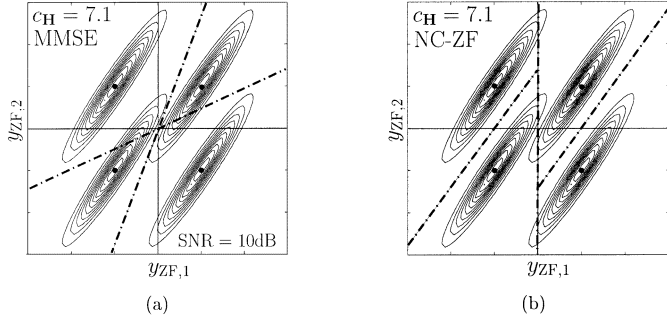


Fig. 3. Same as Fig. 2(b) but with (a) MMSE decision regions and (b) ZF-based NC decision regions instead of ML decision regions. The boundaries of the MMSE and NC decision regions are indicated by dash-dotted lines.

the ML decision regions are nicely matched to the distorted noise pdf, but the ZF decision regions are not because they correspond to simple componentwise quantization. In particular, the boundary lines of the ML decision regions differ mainly by offsets that are orthogonal to the dominant principal axis \mathbf{v}_{M_T} (corresponding to the dominant eigenvalue $\sigma_{\tilde{\mathbf{w}}, M_T}^2$ of $\mathbf{R}_{\tilde{\mathbf{w}}}$ and, thus, to the dominant noise component). This is intuitive, since any shift in the received vector in the direction of the dominant noise component is very likely caused by noise. For bad channels, it is thus desirable that the decision regions be approximately invariant to shifts in the direction of the dominant principal axis \mathbf{v}_{M_T} . In general, the decision regions of linear detection schemes cannot have this property because their boundary lines always go through the origin.

MMSE Detection: Fig. 3(a) shows the decision regions of MMSE equalization-based detection in the ZF equalized domain³ for the bad channel realization (condition number 7.1). The angles of the boundary lines of the MMSE decision regions are better matched to the distorted noise pdf than those of the ZF decision regions; however, the boundary lines still go through the origin and, thus, cannot implement the offsets that would allow them to become similar to the ML decision regions.

NC Detection: Finally, Fig. 3(b) shows the decision regions (again represented in the ZF equalized domain) for ZF-based NC detection. Because the first symbol is simply ZF detected, one boundary of the corresponding decision region is fixed to the ordinate, which is again quite different from the ML decision regions. For successive symbols, offsets can be realized only to a certain extent. As a consequence, ZF-based NC detection performs better than ZF or MMSE detection but is still significantly poorer than ML detection. For MMSE-based NC detection, a similar argument applies. Fig. 1(a) shows that for growing c_H , the performance gap between MMSE-based NC detection and ML detection increases more slowly than that of ZF or MMSE detection but eventually becomes just as wide as that of MMSE detection.

Furthermore, it can be shown that the diversity gain achieved by NC (both ZF- and MMSE-based) is given by $M_R - M_T + 1$ [3]. Thus, for $M_T \approx M_R$, the diversity gain is small.

³We represent and compare the decision regions of all detectors in a common domain, namely, the domain obtained after ZF equalization. Note that ZF equalization by itself does not imply any loss of optimality, and it has the advantage that the symbols are at the correct positions.

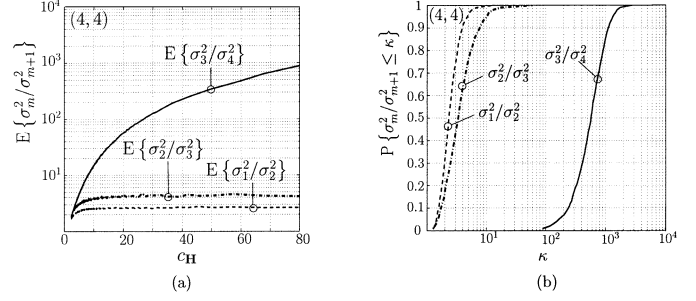


Fig. 4. Ratio of noise variances $\sigma_m^2/\sigma_{m+1}^2$ in the direction of consecutive principal axes in the ZF-equalized domain and channel condition number c_H . (a) Average ratios $E\{\sigma_m^2/\sigma_{m+1}^2\}$ versus the condition number c_H of the respective channel realization for a (4,4) iid Gaussian channel. (b) cdf of the ratios $\sigma_m^2/\sigma_{m+1}^2$ for $c_H = 70$.

III. IDEALIZED BAD CHANNEL MODEL

The previous results suggest that the average performance of suboptimal detection schemes can be improved by making these schemes robust to bad channels. Specifically, the decision regions should be made approximately invariant to a shift in the direction of the dominant noise axis. As a basis for such a modification, we will first formulate an idealized model for bad channels. In Section IV, this will allow us to derive an efficient near-ML detection algorithm for bad channels.

A. Formulation of the Idealized Bad Channel Model

Our idealized bad channel model can be motivated as follows. According to (5), i.e., $\sigma_{\tilde{\mathbf{w}}, m}^2 = \sigma_w^2/\sigma_m^2$, the ratios of the noise variances in the directions of consecutive principal axes are given by $\sigma_{\tilde{\mathbf{w}}, m+1}^2/\sigma_{\tilde{\mathbf{w}}, m}^2 = \sigma_m^2/\sigma_{m+1}^2$. Fig. 4(a) shows estimates of the average ratios $E\{\sigma_m^2/\sigma_{m+1}^2\}$ for a (4,4) channel with iid, circularly symmetric complex-valued, Gaussian entries versus $c_H = \sigma_1^2/\sigma_4^2$ (i.e., the $E\{\sigma_m^2/\sigma_{m+1}^2\}$ were estimated by averaging $\sigma_m^2/\sigma_{m+1}^2$ over an ensemble of channels with a given c_H). It can be seen that for a bad channel (i.e., large c_H), on average, σ_3^2/σ_4^2 is much larger than the other ratios σ_2^2/σ_3^2 and σ_1^2/σ_2^2 . Thus, on average, σ_4^2 is *much smaller* than the other singular values σ_1^2 , σ_2^2 , σ_3^2 , or equivalently, the largest principal-axis noise variance $\sigma_{\tilde{\mathbf{w}}, 4}^2 = \sigma_w^2/\sigma_4^2$ is *much larger* than the other principal-axis noise variances $\sigma_{\tilde{\mathbf{w}}, 1}^2$, $\sigma_{\tilde{\mathbf{w}}, 2}^2$, and $\sigma_{\tilde{\mathbf{w}}, 3}^2$. Simulations show that a similar behavior is also exhibited by higher dimensional channels. To demonstrate that the ratios $\sigma_m^2/\sigma_{m+1}^2$ are well concentrated about their mean $E\{\sigma_m^2/\sigma_{m+1}^2\}$, Fig. 4(b) shows the cdf of $\sigma_m^2/\sigma_{m+1}^2$ for $c_H = 70$.

This suggests that for a bad channel, the largest noise variance (corresponding to the smallest singular value σ_{M_T} of \mathbf{H} and the associated principal axis \mathbf{v}_{M_T}) dominates all the other noise variances and, hence, causes the main part of the bad channel effects that plague suboptimal detection. Therefore, we approximate a bad channel \mathbf{H} by an *idealized bad channel model* (IBCM) $\tilde{\mathbf{H}}$ that is constructed by setting the smallest singular value equal to zero and the remaining singular values equal to the largest singular value:

$$\tilde{\mathbf{H}} \triangleq \mathbf{U} \tilde{\Sigma} \mathbf{V}^H, \quad \text{with} \quad \tilde{\sigma}_{M_T} = 0 \\ \text{and} \quad \tilde{\sigma}_1 = \tilde{\sigma}_2 = \dots = \tilde{\sigma}_{M_T-1} = \sigma_1. \quad (6)$$

Note that whereas the singular vectors \mathbf{v}_m of the respective channel realization \mathbf{H} —and thus the principal axis directions of the ZF-domain noise $\tilde{\mathbf{w}}$ as well—are maintained, the principal-axis noise variances are modified because we use an *infinite* dominant noise variance $\tilde{\sigma}_{\tilde{\mathbf{w}}, M_T}^2 = \sigma_w^2 / \tilde{\sigma}_{M_T}^2 = \infty$ and *equal* remaining noise variances $\tilde{\sigma}_{\tilde{\mathbf{w}}, m}^2 = \sigma_w^2 / \tilde{\sigma}_m^2 = \sigma_w^2 / \sigma_1^2 = \sigma_{w,1}^2$ for $m = 1, \dots, M_T - 1$. (Actually, we will see presently that the values of these remaining noise variances do not matter as long as they are finite and equal.) Indeed, approximating a channel \mathbf{H} with the IBCM $\tilde{\mathbf{H}}$ is equivalent to approximating the hyperellipsoids constituting the contour surfaces of the pdf of $\tilde{\mathbf{w}}$ by hypercylinders of infinite length whose axis is the dominant noise axis \mathbf{v}_{M_T} . This approximation is motivated by our desire to make the decision regions invariant to a shift in the direction of \mathbf{v}_{M_T} , as described in Section II-B.

B. ML Detection for the Idealized Bad Channel Model

Although the IBCM $\tilde{\mathbf{H}}$ is only a crude approximation to a bad channel \mathbf{H} , it does capture an essential part of the “bad channel” effects that plague suboptimal detection. We now consider ML detection for the IBCM (termed *IML detection*); this will later serve as a basis for developing an efficient near-ML detector for bad channels. According to (3), the IML decision rule is

$$\hat{\mathbf{d}}_{\text{IML}} = \arg \min_{\mathbf{d} \in \mathcal{D}} \left\{ (\mathbf{d} - \mathbf{y}_{\text{ZF}})^H \tilde{\mathbf{H}}^H \tilde{\mathbf{H}} (\mathbf{d} - \mathbf{y}_{\text{ZF}}) \right\}. \quad (7)$$

Using (6), we have $\tilde{\mathbf{H}}^H \tilde{\mathbf{H}} = \mathbf{V} \tilde{\Sigma}^2 \mathbf{V}^H = \sigma_1^2 (\mathbf{I} - \mathbf{v}_{M_T} \mathbf{v}_{M_T}^H) = \sigma_1^2 \mathbf{P}_{\mathbf{v}_{M_T}}^\perp$, with $\mathbf{P}_{\mathbf{v}_{M_T}}^\perp \triangleq \mathbf{I} - \mathbf{v}_{M_T} \mathbf{v}_{M_T}^H$ denoting the orthogonal projector onto the space orthogonal to \mathbf{v}_{M_T} . Thus, (7) becomes

$$\begin{aligned} \hat{\mathbf{d}}_{\text{IML}} &= \arg \min_{\mathbf{d} \in \mathcal{D}} \left\{ (\mathbf{d} - \mathbf{y}_{\text{ZF}})^H \mathbf{P}_{\mathbf{v}_{M_T}}^\perp (\mathbf{d} - \mathbf{y}_{\text{ZF}}) \right\} \\ &= \arg \min_{\mathbf{d} \in \mathcal{D}} \left\{ \left\| \mathbf{P}_{\mathbf{v}_{M_T}}^\perp (\mathbf{d} - \mathbf{y}_{\text{ZF}}) \right\|^2 \right\} \end{aligned} \quad (8)$$

where we have used $\mathbf{P}_{\mathbf{v}_{M_T}}^\perp = \mathbf{P}_{\mathbf{v}_{M_T}}^H \mathbf{P}_{\mathbf{v}_{M_T}}^\perp$. Because of the projector $\mathbf{P}_{\mathbf{v}_{M_T}}^\perp$ occurring in (8), the norm of the component of $\mathbf{d} - \mathbf{y}_{\text{ZF}}$ perpendicular to \mathbf{v}_{M_T} is minimized, whereas the component of $\mathbf{d} - \mathbf{y}_{\text{ZF}}$ in the direction of \mathbf{v}_{M_T} (the dominant noise direction) is suppressed in this minimization. Thus, as desired, the IML decision regions are invariant to this latter component.

For an alternative formulation and geometrical interpretation of the IML decision rule (8), 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}_{ZF} :

$$\mathcal{L} : \mathbf{y}_{\text{ref}}(k) \triangleq k \mathbf{v}_{M_T} + \mathbf{y}_{\text{ZF}}, \quad k \in \mathbb{C}. \quad (9)$$

We maintain that the norm $\left\| \mathbf{P}_{\mathbf{v}_{M_T}}^\perp (\mathbf{d} - \mathbf{y}_{\text{ZF}}) \right\|$ minimized in (8) is equal to the distance of \mathbf{d} from \mathcal{L} . Indeed, let $\mathbf{y}_{\mathcal{L}}(\mathbf{d})$ denote the point of \mathcal{L} closest to a given data vector \mathbf{d} , i.e., $\mathbf{y}_{\mathcal{L}}(\mathbf{d}) \triangleq \mathbf{y}_{\text{ref}}(k_0)$ for $k_0 = \arg \min_{k \in \mathbb{C}} \{ \left\| \mathbf{d} - \mathbf{y}_{\text{ref}}(k) \right\|^2 \} = \arg \min_{k \in \mathbb{C}} \{ \left\| \mathbf{d} - (k \mathbf{v}_{M_T} + \mathbf{y}_{\text{ZF}}) \right\|^2 \} = \mathbf{v}_{M_T}^H (\mathbf{d} - \mathbf{y}_{\text{ZF}})$. Then,

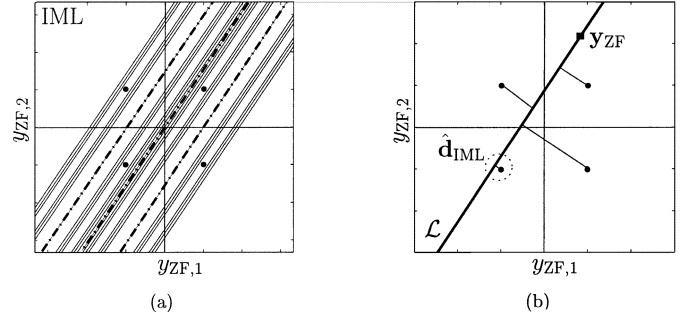


Fig. 5. Illustration of IML detection for a real-valued (2,2) channel and BPSK modulation. (a) pdf of \mathbf{y}_{ZF} for the IBCM corresponding to the bad channel of Figs. 2 and 3 and IML decision regions indicated by dash-dotted lines. (b) Reference-line geometry of IML detection according to (10).

the distance of \mathbf{d} from \mathcal{L} is given by $\left\| \mathbf{d} - \mathbf{y}_{\mathcal{L}}(\mathbf{d}) \right\|$, and this distance can easily be shown to be equal to $\left\| \mathbf{P}_{\mathbf{v}_{M_T}}^\perp (\mathbf{d} - \mathbf{y}_{\text{ZF}}) \right\|$:

$$\begin{aligned} \left\| \mathbf{d} - \mathbf{y}_{\mathcal{L}}(\mathbf{d}) \right\| &= \left\| \mathbf{d} - \mathbf{y}_{\text{ref}}(\mathbf{v}_{M_T}^H (\mathbf{d} - \mathbf{y}_{\text{ZF}})) \right\| \\ &= \left\| \mathbf{d} - \mathbf{v}_{M_T}^H (\mathbf{d} - \mathbf{y}_{\text{ZF}}) \mathbf{v}_{M_T} - \mathbf{y}_{\text{ZF}} \right\| \\ &= \left\| (\mathbf{I} - \mathbf{v}_{M_T} \mathbf{v}_{M_T}^H) (\mathbf{d} - \mathbf{y}_{\text{ZF}}) \right\| \\ &= \left\| \mathbf{P}_{\mathbf{v}_{M_T}}^\perp (\mathbf{d} - \mathbf{y}_{\text{ZF}}) \right\|. \end{aligned}$$

Thus, as claimed, *ML detection for the IBCM is equivalent to finding the data vector $\mathbf{d} \in \mathcal{D}$ that minimizes the distance $\left\| \mathbf{d} - \mathbf{y}_{\mathcal{L}}(\mathbf{d}) \right\|$ from the reference line \mathcal{L} :*

$$\hat{\mathbf{d}}_{\text{IML}} = \arg \min_{\mathbf{d} \in \mathcal{D}} \left\{ \left\| \mathbf{d} - \mathbf{y}_{\mathcal{L}}(\mathbf{d}) \right\|^2 \right\}. \quad (10)$$

This formulation will be essential for developing an efficient ML detection algorithm in Section IV.

To continue the example of Figs. 2 and 3, Fig. 5(a) depicts the pdf of \mathbf{y}_{ZF} for the IBCM associated with the bad channel realization with $c_{\text{H}} = 7.1$. The IML decision regions are also shown. The geometry of the reference-line formulation of IML detection (10) is illustrated in Fig. 5(b).

IV. EFFICIENT ML DETECTOR FOR THE IDEALIZED BAD CHANNEL MODEL

In this section, we develop an efficient ML detection algorithm with complexity $\mathcal{O}(M_T^3)$ for the IBCM. We first show that the reference-line formulation of the IML detector derived in the previous subsection allows a significant reduction of the search set \mathcal{D} .

A. Reduced Search Set $\tilde{\mathcal{D}}$

Let \mathcal{Z}_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 $\mathbf{y} \in \mathcal{Z}_i$, the ZF decision is $\hat{\mathbf{d}}_{\text{ZF}} = \mathbf{d}^{(i)}$. Because, in the ZF-equalized domain, the ZF decision is a simple component-wise quantization, every $\mathbf{y} \in \mathcal{Z}_i$ is closer to $\mathbf{d}^{(i)}$ than to any other data vector $\mathbf{d}^{(j)} \in \mathcal{D}$. We will also say that $\mathbf{d}^{(i)}$ is “the data vector corresponding to \mathcal{Z}_i .”

Theorem 1: The ML detector for the IBCM in (8) is equivalent to

$$\hat{\mathbf{d}}_{\text{IML}} = \arg \min_{\mathbf{d} \in \tilde{\mathcal{D}}} \left\{ \left\| \mathbf{P}_{\mathbf{v}_{M_T}}^\perp (\mathbf{d} - \mathbf{y}_{\text{ZF}}) \right\|^2 \right\} \quad (11)$$

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 (9).

Proof: As before, let $\mathbf{y}_{\mathcal{L}}(\mathbf{d})$ denote the point of \mathcal{L} closest to \mathbf{d} so that $\|\mathbf{d} - \mathbf{y}_{\mathcal{L}}(\mathbf{d})\|$ expresses the distance of \mathbf{d} from \mathcal{L} . Assume (proof by contradiction) 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 $\hat{\mathbf{d}}_{\text{IML}}$, but we know from (10) 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

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 reference plane \mathcal{P} —the 2-D plane $\mathbf{y}_{\text{ref}}(k_R, k_I)$ with $k_R \triangleq \text{Re}\{k\}$ and $k_I \triangleq \text{Im}\{k\}$. Replacing complex scalars by 2-D vectors composed of the real and imaginary parts, the m th scalar component of the reference line equation, given by $y_{\text{ref},m} = kv_{M_T,m} + y_{\text{ZF},m}$, can be written as

$$\mathbf{y}_{\text{ref},m}(\mathbf{k}) = \mathbf{V}_{M_T,m}\mathbf{k} + \mathbf{y}_{\text{ZF},m}, \quad m = 1, \dots, M_T \quad (12)$$

with

$$\mathbf{y}_{\text{ref},m} \triangleq \begin{pmatrix} y_{\text{ref},m,R} \\ y_{\text{ref},m,I} \end{pmatrix}, \quad \mathbf{V}_{M_T,m} \triangleq \begin{pmatrix} v_{M_T,m,R} & -v_{M_T,m,I} \\ v_{M_T,m,I} & v_{M_T,m,R} \end{pmatrix} \\ \mathbf{k} \triangleq \begin{pmatrix} k_R \\ k_I \end{pmatrix}, \quad \mathbf{y}_{\text{ZF},m} \triangleq \begin{pmatrix} y_{\text{ZF},m,R} \\ y_{\text{ZF},m,I} \end{pmatrix}.$$

The reduced search set $\tilde{\mathcal{D}}$ then consists of all data vectors $\mathbf{d} \in \mathcal{D}$ whose ZF decision regions are sliced by the real reference plane \mathcal{P} (since slicing by \mathcal{P} is equivalent to piercing by \mathcal{L}). To characterize $\tilde{\mathcal{D}}$, it is sufficient 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} can easily be recovered from \mathbf{y} as $\mathbf{d} = Q\{\mathbf{y}\}$. The idea behind the algorithm proposed next is that these points can be found in the reference plane \mathcal{P} , and thus, the search for $\tilde{\mathcal{D}}$ can be restricted to \mathcal{P} .

B. 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 wide class of symbol alphabets, this partitioning can be calculated very easily. Subsequently, this will serve as a basis for an efficient determination of the reduced search set $\tilde{\mathcal{D}}$.

We assume a “line-structured” symbol alphabet \mathcal{A} , which, by definition, is such that the boundaries of the quantization regions in the 2-D real symbol domain (i.e., the ML decision regions for transmission of a *single* symbol over an additive white Gaussian noise SISO channel) are given by straight lines. Examples of line-structured alphabets are ASK, QAM, and PSK but not, e.g., a hexagonal constellation. Let P denote the number of

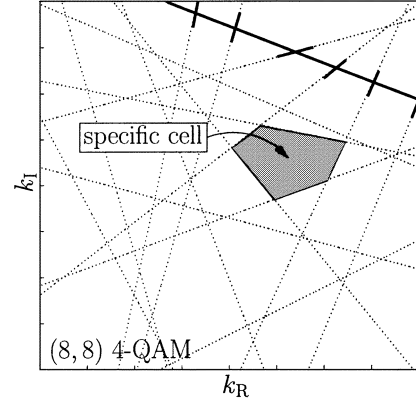


Fig. 6. Boundary lines $\mathcal{B}^{(m,p)}$ in the real reference plane \mathcal{P} for an (8,8) channel and 4-QAM modulation. (The boldface line segment will be addressed at a later point.)

boundary lines in the 2-D real symbol domain, and let the p th real boundary line be given by

$$\mathbf{b}^{(p)}(k_b) = k_b \mathbf{u}^{(p)} + \mathbf{o}^{(p)}, \quad p = 1, \dots, P \quad (13)$$

with $k_b \in \mathbb{R}$, where the 2×1 vectors $\mathbf{u}^{(p)}$ and $\mathbf{o}^{(p)}$ define slope and offset, respectively. 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 $\mathbf{o}^{(1)} = \mathbf{o}^{(2)} = (0 \ 0)^T$.

We now consider the partitioning of the reference plane \mathcal{P} induced by the ZF decision regions. The ZF decision regions are bounded by $M_T P$ th hyperplanes. The (m,p) hyperplane is obtained by setting the m th component of a vector \mathbf{y} equal to the p th boundary line (13). 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 (13) and (12), 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

$$\mathcal{B}^{(m,p)} : \mathbf{k} = \mathbf{k}^{(m,p)}(k_b) = k_b \boldsymbol{\alpha}^{(m,p)} + \boldsymbol{\beta}^{(m,p)} \quad (14)$$

with

$$\boldsymbol{\alpha}^{(m,p)} \triangleq \frac{1}{\rho_m} \mathbf{V}_{M_T,m}^T \mathbf{u}^{(p)} \\ \boldsymbol{\beta}^{(m,p)} \triangleq \frac{1}{\rho_m} \mathbf{V}_{M_T,m}^T (\mathbf{o}^{(p)} - \mathbf{y}_{\text{ZF},m})$$

where $\rho_m \triangleq v_{M_T,m,R}^2 + v_{M_T,m,I}^2$ (note that $\mathbf{V}_{M_T,m}^T \mathbf{V}_{M_T,m} = \rho_m \mathbf{I}$). 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 $\mathcal{B}^{(m,p)}$, $m = 1, \dots, M_T$, $p = 1, \dots, P$ partition the reference plane \mathcal{P} into elementary cells $\mathcal{C} \subset \mathcal{P}$. In Fig. 6, this cell partitioning of \mathcal{P} is illustrated for an (8,8) channel and 4-QAM modulation; here, $M_T = 8$ and $P = 2$ so that we obtain $M_T P = 16$ boundary lines. Each cell \mathcal{C} is the intersection of a ZF decision region with \mathcal{P} . Thus, all points of a given \mathcal{C} will lead to the same quantized data vector. More specifically, recall that each point of \mathcal{P} is parameterized by $\mathbf{k} = (k_R \ k_I)^T$ or, equivalently, by $k = k_R + jk_I$, and

the corresponding vector in \mathbb{C}^{M_T} is $\mathbf{y}_{\text{ref}}(k) = k\mathbf{v}_{M_T} + \mathbf{y}_{\text{ZF}}$ in (9). 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} . Therefore, *any collection of arbitrary cell points—one point for each cell—defines the reduced search set $\tilde{\mathcal{D}}$* . Note that the ZF estimate $\mathbf{y}_{\text{ZF}} = \mathbf{H}^\# \mathbf{r}$ is obtained for $k_R = k_I = 0$; see (9). Thus, $\hat{\mathbf{d}}_{\text{ZF}} = Q\{\mathbf{y}_{\text{ZF}}\}$ belongs to the reduced search set $\tilde{\mathcal{D}}$.

From the above, it follows that the size $|\tilde{\mathcal{D}}|$ of the reduced search set equals the number of cells \mathcal{C} . To find an upper bound on $|\tilde{\mathcal{D}}|$, recall that the cell partitioning is defined by $M_T P$ boundary lines. Assuming that $i - 1$ 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

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

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 a given intersection point. If there are parallel boundary lines (e.g., 16-QAM) and/or if more than two boundary lines intersect in the same point (e.g., 8-PSK), then $|\tilde{\mathcal{D}}| < |\tilde{\mathcal{D}}|_{\max}$.

Comparing $|\tilde{\mathcal{D}}|_{\max}$ with $|\mathcal{D}| = |\mathcal{A}|^{M_T}$, we see that for reasonably high data rates, $|\tilde{\mathcal{D}}| \ll |\mathcal{D}|$. 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 $|\mathcal{D}| = 4^8 = 65\,536$ decision regions. This illustrates the significant reduction of complexity achieved by the reduced search set $\tilde{\mathcal{D}}$.

We will next present an efficient algorithm for ML detection that is based on the reduced search set $\tilde{\mathcal{D}}$ and the insights gained so far. This algorithm consists of two parts: i) efficient determination of $\tilde{\mathcal{D}}$ and ii) efficient calculation of the distance of every $\mathbf{d} \in \tilde{\mathcal{D}}$ to the reference plane \mathcal{P} .

C. Efficient Determination of $\tilde{\mathcal{D}}$

According to the previous subsection, the reduced search set $\tilde{\mathcal{D}}$ can be determined by finding an arbitrary point \mathbf{k} of each cell $\mathcal{C} \subset \mathcal{P}$ since the corresponding $\mathbf{y}_{\text{ref}}(k)$ defines the data vector $\mathbf{d} \in \tilde{\mathcal{D}}$ associated with \mathcal{C} by $\mathbf{d} = Q\{\mathbf{y}_{\text{ref}}(k)\}$. To find a point \mathbf{k} of each cell \mathcal{C} in an efficient manner, we suggest a systematic search of the reference plane along each boundary line.

To search along the (m, p) th boundary line $\mathcal{B}^{(m, p)}$, we 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 $\mathcal{B}^{(m, p)}$. Equating $\mathbf{k}^{(m, p)}(k_b) = k_b \boldsymbol{\alpha}^{(m, p)} + \boldsymbol{\beta}^{(m, p)}$ with $\mathbf{k}^{(m', p')}(k'_b) = k'_b \boldsymbol{\alpha}^{(m', p')} + \boldsymbol{\beta}^{(m', p')}$ and solving for k_b yields

$$\mathbf{k}_{\text{int}}^{(m, p; m', p')} = k_b^{(m, p; m', p')} \boldsymbol{\alpha}^{(m, p)} + \boldsymbol{\beta}^{(m, p)} \quad (15)$$

with

$$k_b^{(m, p; m', p')} = \mathbf{z}^T \left[\frac{1}{\rho_{m'}} \mathbf{V}_{M_T, m'}^T (\mathbf{o}^{(p')} - \mathbf{y}_{\text{ZF}, m'}) - \frac{1}{\rho_m} \mathbf{V}_{M_T, m}^T (\mathbf{o}^{(p)} - \mathbf{y}_{\text{ZF}, m}) \right]$$

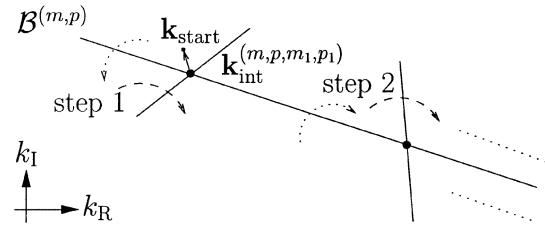


Fig. 7. Algorithm for determining the data vectors $\mathbf{d}^{(i)} \in \tilde{\mathcal{D}}$ corresponding to all cells bounded by the (m, p) th boundary line $\mathcal{B}^{(m, p)}$ (cf. the boldface line in Fig. 6). Based on the first intersection point $\mathbf{k}_{\text{int}}^{(m, p; m_1, p_1)}$, the first cell point $\mathbf{k}_{\text{start}}$ and the associated first data vector $\mathbf{d}^{(1)}$ are calculated. All remaining data vectors $\mathbf{d}^{(i)} \in \tilde{\mathcal{D}}$ associated with $\mathcal{B}^{(m, p)}$ are uniquely determined by the remaining intersection points, using the hops indicated by the dotted and dashed arrows.

where \mathbf{z}^T represents the first row of the inverse of the 2×2 matrix $((1/\rho_m) \mathbf{V}_{M_T, m}^T \mathbf{u}^{(p)}, -(1/\rho_{m'}) \mathbf{V}_{M_T, m'}^T \mathbf{u}^{(p')})$.

As we move along $\mathcal{B}^{(m, p)}$, the intersection points with the other boundary lines (calculated previously) tell us when we cross the border from one cell to the next.⁴ We can sort these intersection points according to, e.g., monotonically increasing real part coordinates. Let $\mathbf{k}_{\text{int}}^{(m, p; m_1, p_1)}$ be the intersection point with the smallest real part coordinate. Fig. 7 shows $\mathcal{B}^{(m, p)}$ (this could be the boldface line depicted in Fig. 6) and the intersection point $\mathbf{k}_{\text{int}}^{(m, p; m_1, p_1)}$. To move from $\mathbf{k}_{\text{int}}^{(m, p; m_1, p_1)}$ into the first cell, we add a small offset $\mathbf{k}_{\text{offset}}$ and obtain the new point $\mathbf{k}_{\text{start}} = \mathbf{k}_{\text{int}}^{(m, p; m_1, p_1)} + \mathbf{k}_{\text{offset}}$, as illustrated in Fig. 7. The vector in \mathbb{C}^{M_T} corresponding to $\mathbf{k}_{\text{start}}$ is given by $\mathbf{y}^{(1)} = k_{\text{start}} \mathbf{v}_{M_T} + \mathbf{y}_{\text{ZF}}$. We then obtain our first data vector $\mathbf{d}^{(1)} \in \tilde{\mathcal{D}}$ associated with $\mathcal{B}^{(m, p)}$ as $\mathbf{d}^{(1)} = Q\{\mathbf{y}^{(1)}\}$.

The remaining data vectors $\mathbf{d}^{(i)} \in \tilde{\mathcal{D}}$ along $\mathcal{B}^{(m, p)}$ are now determined by alternately “hopping” over $\mathcal{B}^{(m, p)}$ and an intersecting boundary line, as illustrated in Fig. 7. At each intersection, we perform one *step* consisting of two *hops*. The first hop (indicated by the dotted arrow) is over $\mathcal{B}^{(m, p)}$, i.e., the current search line; this corresponds to an update of the m th data vector component across the p th boundary. The second hop (indicated by the dashed arrow) is over $\mathcal{B}^{(m_1, p_1)}$, i.e., our first intersecting boundary line; this corresponds to an update of the m_1 th data vector component across the p_1 th boundary.

These data vector component updates can be performed without calculating new \mathbf{k} or \mathbf{y} vectors (i.e., no hops are actually implemented). 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, by definition, $\mathcal{B}^{(m', p')}$ 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

$$\mathbf{d}^{(i+1)} = \mathbf{d}^{(i)} + \Delta \mathbf{d}_{m', p'}^{(i)}$$

⁴For simplicity of exposition, 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 there, all the P boundary lines in the symbol plane intersect at the origin, and this geometry is maintained in \mathcal{P} . For example, for 8-PSK, we obtain M_T intersection points, each of which is the intersection of $P = 4$ boundary lines. However, these points are known *a priori*, and the following algorithm can easily be extended to this situation.

where the vector $\Delta \mathbf{d}_{m',p'}^{(i)}$ is zero except for the m' th component. That component is given by an update value $\Delta d_{m',p'}^{(i)}$ that depends on the m' th component of $\mathbf{d}^{(i)}$ and on the p' th boundary. For example, for 4-QAM, we have

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

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

D. Efficient Calculation of Distances

According to Theorem 1, we have to minimize the distance $\psi_i \triangleq \|\mathbf{P}_{\mathbf{v}_{M_T}}^\perp (\mathbf{d}^{(i)} - \mathbf{y}_{ZF})\|$ over all $\mathbf{d}^{(i)} \in \tilde{\mathcal{D}}$. In the previous subsection, we have shown how to determine $\tilde{\mathcal{D}}$ efficiently. We now present an efficient recursive algorithm for calculating the squared distances ψ_i^2 .

Again, we move along $\mathcal{B}^{(m,p)}$. Suppose that ψ_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 $\Delta \mathbf{d}_{m',p'}^{(i)}$. Thus, the distance for $\mathbf{d}^{(i+1)}$ is

$$\begin{aligned} \psi_{i+1}^2 &= \left\| \mathbf{P}_{\mathbf{v}_{M_T}}^\perp \left(\mathbf{d}^{(i)} + \Delta \mathbf{d}_{m',p'}^{(i)} - \mathbf{y}_{ZF} \right) \right\|^2 \\ &= \left\| \mathbf{P}_{\mathbf{v}_{M_T}}^\perp \left(\mathbf{d}^{(i)} - \mathbf{y}_{ZF} \right) \right\|^2 + \left\| \mathbf{P}_{\mathbf{v}_{M_T}}^\perp \Delta \mathbf{d}_{m',p'}^{(i)} \right\|^2 \\ &\quad + 2 \operatorname{Re} \left\{ \left(\mathbf{d}^{(i)} - \mathbf{y}_{ZF} \right)^H \mathbf{P}_{\mathbf{v}_{M_T}}^{\perp H} \mathbf{P}_{\mathbf{v}_{M_T}}^\perp \Delta \mathbf{d}_{m',p'}^{(i)} \right\} \\ &= \psi_i^2 + \Delta_i \end{aligned} \quad (16)$$

with

$$\begin{aligned} \Delta_i &= \left(1 - |v_{M_T, m'}|^2 \right) \left| \Delta d_{m',p'}^{(i)} \right|^2 \\ &\quad + 2 \operatorname{Re} \left\{ \left(\tilde{d}_{m'}^{(i)} - v_{M_T, m'}^* \tilde{\mathbf{d}}^{(i)H} \mathbf{v}_{M_T} \right) \Delta d_{m',p'}^{(i)} \right\} \end{aligned} \quad (17)$$

where $v_{M_T, m'}$ is the m' th component of \mathbf{v}_{M_T} and $\tilde{d}_{m'}^{(i)}$ is the m' th component of $\tilde{\mathbf{d}}^{(i)} \triangleq \mathbf{d}^{(i)} - \mathbf{y}_{ZF}$. This recursion is initialized by calculating $\psi_1^2 = \|\mathbf{P}_{\mathbf{v}_{M_T}}^\perp (\mathbf{d}^{(1)} - \mathbf{y}_{ZF})\|^2$ in a straightforward manner.

E. Summary of the ML Detection Algorithm

We have now developed the main elements of our efficient ML detection algorithm for the IBCM. In an actual implementation of this algorithm, the recursive procedure described in Sections IV-C and D has to be performed for all $M_T P$ boundary lines except for the last one (whose data vectors have all been processed before) so that all data vectors $\mathbf{d} \in \tilde{\mathcal{D}}$ are taken into account and the ML solution according to (11) is obtained. For each boundary line, at most $2M_T P$ different data vectors are considered, which results in a total maximum of $2M_T P(M_T P - 1) \approx 2M_T^2 P^2$ checked data vectors. Compared with $|\tilde{\mathcal{D}}|_{\max} = (M_T P)^2/2 + M_T P/2 + 1$, we see that most of the data vectors $\mathbf{d} \in \tilde{\mathcal{D}}$ are checked several times. (Of course, the distances for data vectors that have been checked before

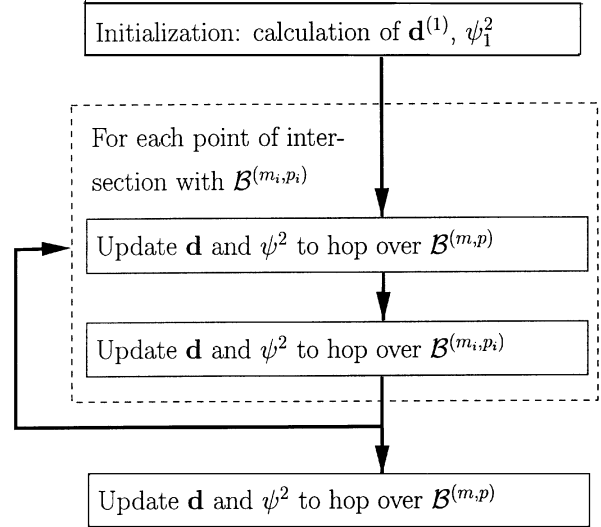


Fig. 8. Processing associated with the (m, p) th boundary line $\mathcal{B}^{(m,p)}$.

need not be calculated anew.) For example, for an (8,8) channel and a 4-QAM alphabet, we have a maximum of 448 checked data vectors, of which maximally 137 are different. Note, however, that most of the multiply checked data vectors are obtained through efficient single symbol updates as described above.

The principal steps of our algorithm can finally be summarized as follows.

- Determine all $M_T P$ boundary lines $\mathcal{B}^{(m,p)}$ in \mathcal{P} [see (14)].
- Calculate all intersection points (15) and order them for each $\mathcal{B}^{(m,p)}$, e.g., according to increasing real part coordinates.
- For each $\mathcal{B}^{(m,p)}$, determine the associated data vectors $\mathbf{d}^{(i)} \in \tilde{\mathcal{D}}$ and their distances ψ_i^2 , as discussed in Sections IV-C and D. The processing associated with $\mathcal{B}^{(m,p)}$ is summarized in Fig. 8.

V. EXTENSION TO ARBITRARY CHANNELS

The algorithm presented in the last section performs ML detection for the IBCM. Because actual channel realizations will not conform to the IBCM, we will now extend our algorithm to an arbitrary MIMO channel \mathbf{H} . Specifically, we propose to use an IBCM approximation of \mathbf{H} to find the reduced search set $\tilde{\mathcal{D}}$ and then to minimize $\|\mathbf{r} - \mathbf{H}\mathbf{d}\|$ over $\tilde{\mathcal{D}}$, i.e.,

$$\hat{\mathbf{d}}_{\text{ML}} \triangleq \arg \min_{\mathbf{d} \in \tilde{\mathcal{D}}} \{ \|\mathbf{r} - \mathbf{H}\mathbf{d}\|^2 \}.$$

Note that $\hat{\mathbf{d}}_{\text{ML}}$ is *not* guaranteed to be equal to the ML decision $\tilde{\mathbf{d}}_{\text{ML}}$ since we minimize $\|\mathbf{r} - \mathbf{H}\mathbf{d}\|^2$ not over the whole data set \mathcal{D} of size $|\mathcal{A}|^{M_T}$ (which would have a computational complexity that is exponential in M_T) but over the reduced set $\tilde{\mathcal{D}}$. On the other hand, the complexity of our algorithm is only $\mathcal{O}(M_T^3)$, as will be shown presently. The algorithm consists of the following steps.

- Calculate $(\mathbf{H}^H \mathbf{H})^{-1}$ and $\mathbf{y}_{ZF} = (\mathbf{H}^H \mathbf{H})^{-1} \mathbf{H}^H \mathbf{r}$.
- Determine the dominant eigenvector \mathbf{v}_{M_T} of $(\mathbf{H}^H \mathbf{H})^{-1}$.
- Using an IBCM approximation of \mathbf{H} , find the reduced search set $\tilde{\mathcal{D}}$ as explained in Section IV-C.

- Search $\tilde{\mathcal{D}}$ for the data vector minimizing $\|\mathbf{r} - \mathbf{H}\mathbf{d}\|$. The squared distances $\psi_i^2 = \|\mathbf{r} - \mathbf{H}\mathbf{d}^{(i)}\|^2$ can again be calculated recursively; the update term Δ_i in (17) is replaced by

$$\Delta_i = \|\mathbf{h}_m\|^2 \left| \Delta d_{m',p'}^{(i)} \right|^2 - 2 \operatorname{Re} \left\{ \left(\mathbf{r} - \mathbf{H}\mathbf{d}^{(i)} \right)^H \mathbf{h}_m \Delta d_{m',p'}^{(i)} \right\} \quad (18)$$

where \mathbf{h}_m denotes the m th column of \mathbf{H} . Here, the term $(\mathbf{r} - \mathbf{H}\mathbf{d}^{(i)})^H$ can be calculated recursively as well, and the $\|\mathbf{h}_m\|^2$ can be precalculated.

Two cases where $\hat{\mathbf{d}}_{\text{ML}}^{\sim}$ does coincide with $\hat{\mathbf{d}}_{\text{ML}}$ are i) an orthogonal channel (here, $\hat{\mathbf{d}}_{\text{ML}} = \hat{\mathbf{d}}_{\text{ZF}}$ and $\hat{\mathbf{d}}_{\text{ML}}^{\sim} = \hat{\mathbf{d}}_{\text{ZF}}$ because $\hat{\mathbf{d}}_{\text{ZF}} \in \tilde{\mathcal{D}}$) and ii) an IBCM (here, $\hat{\mathbf{d}}_{\text{ML}}^{\sim} = \hat{\mathbf{d}}_{\text{IML}} = \hat{\mathbf{d}}_{\text{ML}}$). We can thus expect near-ML performance for very good channels and for bad channels with a single dominant noise axis.

For $M_T = M_R$, we now study how the algorithm's complexity depends on the parameters $|\mathcal{A}|$ and M_T that determine the transmission rate. We first list the preparatory steps that have to be performed once for each data block over which the channel is assumed constant, together with their complexity:

- calculation of $(\mathbf{H}^H \mathbf{H})^{-1}$ and $\mathbf{G} = (\mathbf{H}^H \mathbf{H})^{-1} \mathbf{H}^H$: complexity $\mathcal{O}(M_T^3)$;
- calculation of \mathbf{v}_{M_T} : complexity $\mathcal{O}(M_T^2)$ if the power method [7] is used;
- 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 have to be performed at each time instant (i.e., for each transmitted data vector):

- determination of all $M_T P$ boundary lines in \mathcal{P} and their intersection points: complexity $\mathcal{O}(M_T^2 P^2)$;
- calculation of $M_T P$ initial data vectors and their associated distances: complexity $\mathcal{O}(M_T^3 P)$;
- calculation of roughly $2M_T P(M_T P - 1)$ distance updates according to (18): complexity $\mathcal{O}(M_T^3 P^2)$.

Hence, the *overall complexity* of our algorithm 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 $|\mathcal{A}|$.

VI. SPHERE-PROJECTION ALGORITHM

In this section, we will achieve another substantial reduction of computational complexity through a further reduction of the data search set. The resulting sphere projection algorithm (SPA) can be viewed as a simple nonlinear add-on to an existing suboptimal scheme such as ZF, MMSE, or NC detection. This add-on improves the error-rate and diversity-gain performance of the suboptimal detector by making it robust to the bad channel effects discussed in Section II.

The add-on construction of the SPA is as follows. Let $\hat{\mathbf{d}}_{\text{sub}}$ denote the result of the suboptimal detector. This result can be expected to be reasonably good for a good channel. In order to improve the performance for bad channels, we additionally consider a suitably chosen set $\mathcal{D}_+ \subset \mathcal{D}$ of valid data vectors

that are potentially better than $\hat{\mathbf{d}}_{\text{sub}}$ in the sense of smaller $\|\mathbf{r} - \mathbf{H}\mathbf{d}\|^2$. We then minimize $\|\mathbf{r} - \mathbf{H}\mathbf{d}\|^2$ over the search set \mathcal{D}_{SP} that consists of $\hat{\mathbf{d}}_{\text{sub}}$ and all data vectors in \mathcal{D}_+ :

$$\hat{\mathbf{d}}_{\text{SP}} \triangleq \arg \min_{\mathbf{d} \in \mathcal{D}_{\text{SP}}} \{\|\mathbf{r} - \mathbf{H}\mathbf{d}\|^2\}, \quad \text{with } \mathcal{D}_{\text{SP}} \triangleq \{\hat{\mathbf{d}}_{\text{sub}}\} \cup \mathcal{D}_+. \quad (19)$$

The SPA is an add-on to the given suboptimal detector because $\hat{\mathbf{d}}_{\text{sub}}$ is calculated and included in the total search set \mathcal{D}_{SP} .

To keep our discussion simple, we consider only alphabets with constant modulus σ_d (i.e., arbitrary PSK constellations, which includes 4-QAM and binary antipodal signaling). Here, all data vectors \mathbf{d} are located on an M_T -dimensional “data hypersphere” \mathcal{H} about the origin, with radius $R = \sigma_d \sqrt{M_T}$. This geometry will allow a simple construction of \mathcal{D}_+ that uses a projection onto the data hypersphere \mathcal{H} (this explains the name SPA). The SPA can be extended to symbol alphabets that do not satisfy the constant-modulus property by using several hyperspheres; however, this results in increased complexity.

A. Construction of the Additional Search Set \mathcal{D}_+

The additional search set \mathcal{D}_+ has to be constructed such that it improves the detector performance in the bad channel case. Therefore, let us consider a bad channel whose dominant noise component in the direction of \mathbf{v}_{M_T} (i.e., of the reference line \mathcal{L}) is much larger than all other noise components. This channel can again be approximated by an IBCM. Because the ML detector for the IBCM chooses the data vector with minimum distance from \mathcal{L} (see Section III-B), it makes sense to construct \mathcal{D}_+ as a set of data vectors that are close to \mathcal{L} . On the other hand, we know that all data vectors are located on the data hypersphere \mathcal{H} . With the SPA, for reasons of algorithmic simplicity, we attempt to combine these two properties by choosing for \mathcal{D}_+ data vectors that are close to the intersection $\mathcal{L} \cap \mathcal{H}$. However, this intersection does not always exist. Therefore, two cases will be distinguished.

- *Case 1:* If \mathcal{L} intersects \mathcal{H} , we choose \mathcal{D}_+ to consist of data vectors $\mathbf{d} \in \mathcal{H}$ located at or at least close to the intersection.
- *Case 2:* If \mathcal{L} does not intersect \mathcal{H} , we choose \mathcal{D}_+ to consist of data vectors $\mathbf{d} \in \mathcal{H}$ that are close to \mathcal{L} .

In the following, we will elaborate on both cases.

Case 1: Suppose \mathcal{L} intersects \mathcal{H} . Using the decomposition $\mathbf{y}_{\text{ZF}} = \mathbf{y}_{\mathbf{v}_{M_T}} + \mathbf{y}_{\mathbf{v}_{M_T}}^\perp$, where $\mathbf{y}_{\mathbf{v}_{M_T}} = (\mathbf{v}_{M_T}^H \mathbf{y}_{\text{ZF}}) \mathbf{v}_{M_T}$ is colinear with \mathbf{v}_{M_T} and $\mathbf{y}_{\mathbf{v}_{M_T}}^\perp = \mathbf{P}_{\mathbf{v}_{M_T}}^\perp \mathbf{y}_{\text{ZF}}$ is orthogonal to \mathbf{v}_{M_T} , \mathcal{L} can be rewritten as

$$\mathbf{y}_{\text{ref}}(\tilde{k}) = \tilde{k} \mathbf{v}_{M_T} + \mathbf{y}_{\mathbf{v}_{M_T}}^\perp, \quad \text{with } \tilde{k} = k + \mathbf{v}_{M_T}^H \mathbf{y}_{\text{ZF}} \in \mathbb{C}. \quad (20)$$

The intersection $\mathcal{L} \cap \mathcal{H}$ corresponds to the equation $\|\mathbf{y}_{\text{ref}}(\tilde{k})\|^2 = R^2$ or, equivalently

$$|\tilde{k}|^2 + \|\mathbf{y}_{\mathbf{v}_{M_T}}^\perp\|^2 = R^2. \quad (21)$$

In the real reference plane \mathcal{P} , \tilde{k} is represented by the 2-D real vector $\tilde{\mathbf{k}}$, and $|\tilde{k}|^2 = \|\tilde{\mathbf{k}}\|^2$. We can then write (21) as

$$\|\tilde{\mathbf{k}}\|^2 = \tilde{R}^2 \quad (22)$$

with radius $\tilde{R} \triangleq \sqrt{R^2 - \|\mathbf{y}_{\mathbf{v}_{M_T}}^\perp\|^2}$ (note that $R^2 - \|\mathbf{y}_{\mathbf{v}_{M_T}}^\perp\|^2 \geq 0$ because we assumed that an intersection $\mathcal{L} \cap \mathcal{H}$ exists). Equa-

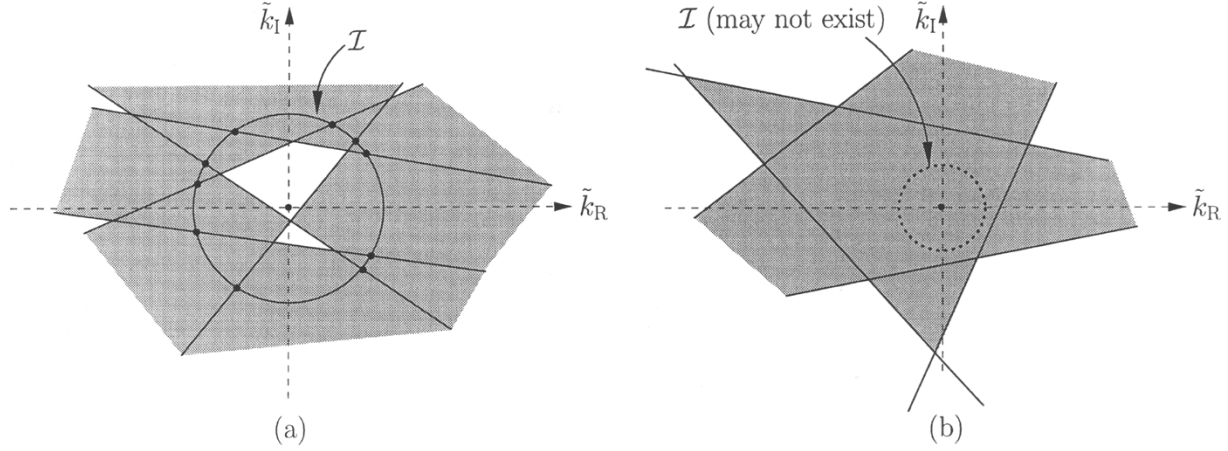


Fig. 9. Geometry underlying the SPA. (a) Some boundary lines intersect the intersection circle \mathcal{I} (only possible for Case 1). (b) No boundary line intersects \mathcal{I} (possible for Case 1 and Case 2). The shaded regions indicate (some of) the cells corresponding to data vectors in \mathcal{D}_+ .

tion (22) defines an *intersection circle* \mathcal{I} that represents $\mathcal{L} \cap \mathcal{H}$ in the reference plane \mathcal{P} . Because $\tilde{k} = k + \mathbf{v}_{M_T}^H \mathbf{y}_{ZF}$, the center of \mathcal{I} is at $k = -\mathbf{v}_{M_T}^H \mathbf{y}_{ZF}$.

Next, we find all cells $\mathcal{C} \subset \mathcal{P}$ that are pierced by the intersection circle \mathcal{I} . We first calculate all $M_T P$ boundary lines $\mathcal{B}^{(m,p)}$. We then calculate the intersection points—if they exist—of each $\mathcal{B}^{(m,p)}$ [see (14)] with \mathcal{I} [see (22)], i.e., $|k^{(m,p)}(k_b) + \mathbf{v}_{M_T}^H \mathbf{y}_{ZF}|^2 = \hat{R}^2$, where $k^{(m,p)}(k_b)$ is given by (14). These intersection points are illustrated in Fig. 9(a). (Here, we assume that at least one boundary line intersects \mathcal{I} . If no such intersection exists although \mathcal{L} intersects \mathcal{H} —cf. Fig. 9(b)—the processing for Case 2 described further below has to be performed.) The intersection points are sorted according to their angle. Starting with angle $\phi = 0$, the first data vector in \mathcal{D}_+ is obtained as $\mathbf{d}^{(1)} = Q\{\mathbf{y}_{\text{ref}}(\hat{R})\}$. Now, we move along \mathcal{I} from one intersection point to the next one. In this process, we apply the data component update procedure from Section IV-C to obtain the remaining data vectors $\mathbf{d}^{(i)} \in \mathcal{D}_+$, and we use the recursion (18) to efficiently calculate the distances $\psi_i^2 = \|\mathbf{r} - \mathbf{H}\mathbf{d}^{(i)}\|^2$ in (19).

The size of the resulting \mathcal{D}_+ is bounded as $|\mathcal{D}_+| \leq 2M_T P$ because each one of the $M_T P$ boundary lines $\mathcal{B}^{(m,p)}$ has zero or two intersection points with \mathcal{I} . The worst case $|\mathcal{D}_+| = 2M_T P$ occurs if and only if all boundary lines intersect \mathcal{I} .

Case 2: If $R^2 \leq \|\mathbf{y}_{\mathbf{v}_{M_T}}^\perp\|^2$, then \mathcal{L} and \mathcal{H} do not intersect [see Fig. 9(b)]. It can here be shown that $\mathbf{y}_{\mathbf{v}_{M_T}}^\perp$ [which, according to (20), corresponds to $\tilde{k} = 0$] is the point of \mathcal{L} with minimum distance from \mathcal{H} . We thus take $\mathbf{d}^{(1)} = Q\{\mathbf{y}_{\mathbf{v}_{M_T}}^\perp\}$ as the first data vector in \mathcal{D}_+ . Simulation results indicate that it is advantageous to include also the nearest neighbors of $\mathbf{d}^{(1)}$ in \mathcal{D}_+ . These additional data vectors can easily be found by substituting the nearest-neighbor symbols for the individual components of $\mathbf{d}^{(1)}$. For PSK constellations, we obtain two nearest neighbors for each data vector component, yielding $|\mathcal{D}_+| = 2M_T + 1$. The distances $\psi_i^2 = \|\mathbf{r} - \mathbf{H}\mathbf{d}^{(i)}\|^2$ in (19) can again be calculated recursively according to (18).

The same procedure is used if \mathcal{L} and \mathcal{H} intersect but no boundary line intersects the intersection circle \mathcal{I} (cf. the discussion of Case 1 above).

B. Discussion of the Sphere-Projection Algorithm

Having explained the efficient construction of the additional search set \mathcal{D}_+ , we now discuss the use of the SPA⁵ as an add-on to a suboptimal detector.

Algorithm Summary and Complexity: The principal steps of the SPA can be summarized as follows.

- Calculate $\hat{\mathbf{d}}_{\text{sub}}$ (the result of the given suboptimal detector) and the associated distance $\|\mathbf{r} - \mathbf{H}\hat{\mathbf{d}}_{\text{sub}}\|^2$.
- Calculate $(\mathbf{H}^H \mathbf{H})^{-1}$ and $\mathbf{y}_{ZF} = (\mathbf{H}^H \mathbf{H})^{-1} \mathbf{H}^H \mathbf{r}$.
- Calculate the dominant eigenvector \mathbf{v}_{M_T} of $(\mathbf{H}^H \mathbf{H})^{-1}$.
- Calculate all boundary lines $\mathcal{B}^{(m,p)}$ in \mathcal{P} .
- Determine the additional search set \mathcal{D}_+ and calculate the corresponding distances
 - by considering all data vectors whose cells are pierced by the intersection circle \mathcal{I} ;
 - or, if \mathcal{I} does not exist or if no boundary line intersects \mathcal{I} , by considering $\mathbf{d}^{(1)} = Q\{\mathbf{y}_{\mathbf{v}_{M_T}}^\perp\}$ and its next neighbors.
- Find the minimal distance (including the distance obtained for $\hat{\mathbf{d}}_{\text{sub}}$).

Here, $\hat{\mathbf{d}}_{\text{sub}}$ may be the result of ZF detection, MMSE detection, ZF-based NC detection, or MMSE-based NC detection; the resulting SPA variants will be referred to as ZF/SP, MMSE/SP, NC-ZF/SP, and NC-MMSE/SP detection, respectively.

The computational complexity of the SPA add-on (not counting calculation of $\hat{\mathbf{d}}_{\text{sub}}$) can be estimated as follows. Among the preparatory steps that have to be performed once for each transmitted data block during which the channel is constant, calculation of $(\mathbf{H}^H \mathbf{H})^{-1} \mathbf{H}^H$ has the dominant complexity of $\mathcal{O}(M_T^3)$. Additionally, at each time instant, we have to compute two distances with complexity $\mathcal{O}(M_T^2)$ and, in the worst case, $2M_T P - 1$ efficient distance updates with total complexity $\mathcal{O}(M_T^2 P)$. An experimental assessment of complexity will be provided in Section VII-C.

⁵In both Case 1 and Case 2, an important aspect of the SPA is that we look for the point(s) on the data hypersphere \mathcal{H} that is/are closest to the reference line \mathcal{L} . In a certain sense, this corresponds to a projection onto the hypersphere \mathcal{H} , which explains the name “sphere-projection algorithm.” In Case 1, where \mathcal{L} and \mathcal{H} intersect, the projection points are given by the intersection circle \mathcal{I} .

SPA Variants: To use the SPA as an add-on to ZF detection, we choose $\hat{\mathbf{d}}_{\text{sub}} = Q\{\mathbf{y}_{\text{ZF}}\}$. The resulting *ZF/SP algorithm* is the most efficient SPA variant because calculation of \mathbf{y}_{ZF} is a part of the SPA.

To use the SPA as an add-on to MMSE detection, we choose $\hat{\mathbf{d}}_{\text{sub}} = Q\{\mathbf{y}_{\text{MMSE}}\}$, where $\mathbf{y}_{\text{MMSE}} = (\mathbf{H}^H \mathbf{H} + (\sigma_w^2/\sigma_d^2)\mathbf{I})^{-1} \mathbf{H}^H \mathbf{r}$ is the result of MMSE equalization. Furthermore, we propose to replace the reference line \mathcal{L} in (20) by the “MMSE-based reference line” \mathcal{L}' : $\mathbf{y}_{\text{ref}}(\hat{k}) = \hat{k} \mathbf{v}_{M_T} + \mathbf{P}_{\mathbf{v}_{M_T}}^\perp \mathbf{y}_{\text{MMSE}}$. \mathcal{L}' is again parallel to the dominant principal axis \mathbf{v}_{M_T} , but its offset is the component of \mathbf{y}_{MMSE} (rather than of \mathbf{y}_{ZF}) perpendicular to \mathbf{v}_{M_T} . The resulting *MMSE/SP algorithm* can be obtained simply by replacing in the ZF/SP algorithm $(\mathbf{H}^H \mathbf{H})^{-1}$ with $(\mathbf{H}^H \mathbf{H} + (\sigma_w^2/\sigma_d^2)\mathbf{I})^{-1}$. The rationale for the MMSE/SP algorithm is that on average, \mathbf{y}_{MMSE} will be closer to the transmitted data vector \mathbf{d}_0 than \mathbf{y}_{ZF} . Note that applying the power method to $(\mathbf{H}^H \mathbf{H} + (\sigma_w^2/\sigma_d^2)\mathbf{I})^{-1}$ instead of $(\mathbf{H}^H \mathbf{H})^{-1}$ requires a few more iterations to obtain accurate results for \mathbf{v}_{M_T} . This is because the “regularization term” $(\sigma_w^2/\sigma_d^2)\mathbf{I}$ decreases the ratio of the largest to second largest singular value of $(\mathbf{H}^H \mathbf{H} + (\sigma_w^2/\sigma_d^2)\mathbf{I})^{-1}$, as compared with $(\mathbf{H}^H \mathbf{H})^{-1}$, which slows down the convergence of the power method [7].

Finally, to combine the SPA with NC detection (either ZF-based or MMSE-based), we propose to execute the ZF/SP or MMSE/SP algorithm, respectively, except for the use of the NC detection result for $\hat{\mathbf{d}}_{\text{sub}}$. As we will demonstrate in Section VII, the NC-MMSE/SP algorithm is able to yield very-near-to-ML performance at a fraction of the computational cost of Fincke and Phost’s sphere-decoding algorithm.

VII. SIMULATION RESULTS

Finally, we present simulation results in order to assess the error-rate performance and computational complexity of the proposed detection algorithms in comparison to the ZF, MMSE, NC, and ML schemes. In our simulations, we used 4-QAM modulation and MIMO channels with iid Gaussian entries. The dominant noise axis \mathbf{v}_{M_T} was computed by means of the power method [7].

A. SER versus Number of Power Method Iterations

First, we study the dependence of the symbol error rate (SER) on the number of power method iterations, which is denoted by K . Fig. 10 shows the SER versus K for ZF/SP and MMSE/SP detection for a (6,6) channel at an SNR⁶ of 20 dB. For comparison, the SER achieved with ZF and MMSE detection is indicated by horizontal lines. It is seen that for $K \geq 1$, the SPA add-on yields significant performance improvements. For $K = 0$, we used a randomly chosen \mathbf{v}_{M_T} ; as can be expected, in this case, the performance improvements yielded by the SPA add-on are negligible. For both ZF/SP and MMSE/SP, increasing K beyond 4 does not yield any additional performance improvement. Thus, we chose $K = 4$ in all simulations presented below.

⁶The SNR is defined as $\mathbb{E}\{||\mathbf{H}\mathbf{d}||^2\}/\mathbb{E}\{||\mathbf{w}||^2\} = M_T \sigma_d^2/\sigma_w^2$.

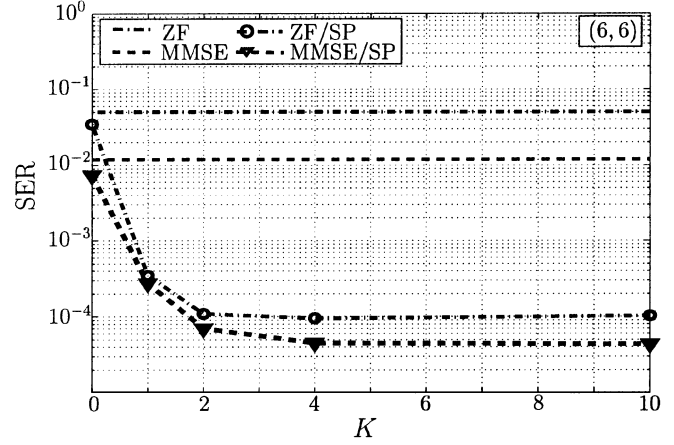
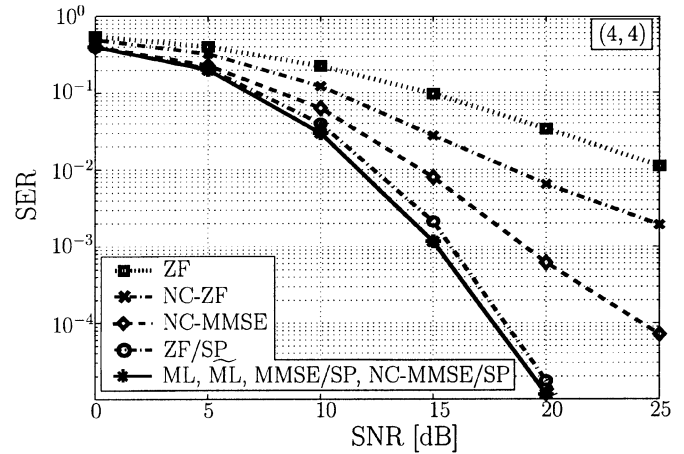
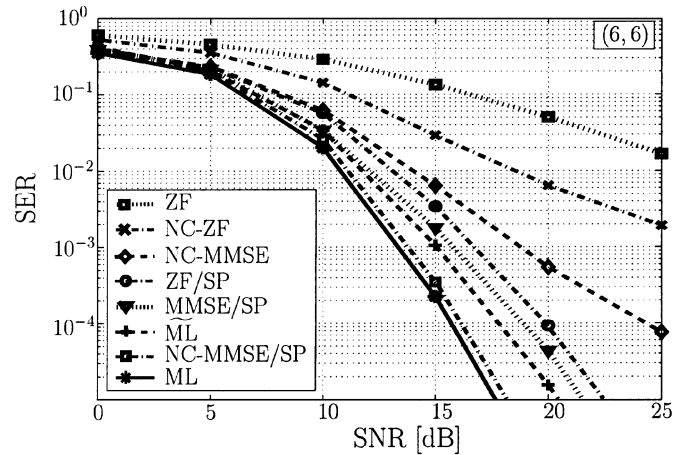


Fig. 10. SER versus the number of power method iterations K for a (6,6) channel and 4-QAM modulation at an SNR of 20 dB. $K = 0$ corresponds to a randomly chosen axis \mathbf{v}_{M_T} .



(a)



(b)

Fig. 11. SER versus SNR performance of the various proposed and standard detectors using 4-QAM modulation. (a) (4,4) channel. (b) (6,6) channel.

B. SER versus SNR

Figs. 11(a) and (b) show the SER versus SNR performance of the various proposed and standard detectors for a (4,4) channel and a (6,6) channel, respectively. As in Section V, $\widetilde{\text{ML}}$ denotes the extension of the IML algorithm to arbitrary channels. For the NC algorithms, we used the layer ordering maximizing the post-

TABLE I
MEASURED COMPLEXITY IN KFLOPS FOR (a) PREPARATORY (OVERHEAD) OPERATIONS PERFORMED ONCE PER DATA BLOCK AND
(b) DATA DETECTION OPERATIONS PERFORMED ONCE PER DATA VECTOR

measured kflops – block operations						
channel	FPSD	$\widetilde{\text{ML}}$	lin.	NC	lin./SP	NC/SP
(4, 4)	2.1	2.8	2.1	4.2	2.8	4.9
(6, 6)	6.4	7.8	6.4	16.2	7.8	17.6
(8, 8)	14.2	16.8	14.2	45.1	16.8	47.6

(a)

measured kflops – vector operations									
channel	FPSD		$\widetilde{\text{ML}}$	lin.	NC	lin./SP		NC/SP	
	av.	max.				av.	max.	av.	max.
(4, 4)	32	82	13	0.18	0.4	1.3	2.4	1.8	2.8
(6, 6)	68	300	41	0.37	1.1	2.5	4.7	3.6	5.7
(8, 8)	122	671	93	0.62	1.9	3.7	7.6	5.6	9.5

(b)

detection signal-to-noise ratio at each detection step as proposed in [2]. The following conclusions can be drawn from the results shown in Fig. 11.

- All proposed detectors perform substantially better than the standard suboptimal detectors. In particular, even the simplest of the proposed detectors (ZF/SP) outperforms the best standard suboptimal detector (NC-MMSE with optimal layer ordering).
- All proposed detectors achieve higher diversity orders than the standard suboptimal detectors.
- For the (4,4) channel, the MMSE/SP and NC-MMSE/SP detectors perform practically as well as the ML detector [the corresponding SER curves in Fig. 11(a) are indistinguishable], and the ZF/SP detector achieves near-ML performance.
- For the (6,6) channel, the NC-MMSE/SP detector achieves near-ML performance, and the performance of the other proposed detectors is intermediate between that of the ML detector and that of the best standard suboptimal detector (NC-MMSE with optimal layer ordering).
- For the (6,6) channel, the performance of the ZF/SP and MMSE/SP detectors is close to that of the $\widetilde{\text{ML}}$ detector. This shows that our strategy for constructing \mathcal{D}_+ works well.
- A comparison of the results obtained for the (4,4) and (6,6) channels suggests that for increasing channel size, the performance of the proposed algorithms degrades (compared with ML performance). This is due to the IBCM approximation underlying our algorithms. Specifically, for increasing channel size, the probability that two or more principal axes are dominant (rather than just one as assumed in the IBCM) increases so that the IBCM approximation becomes less accurate.

C. Computational Complexity

A rough picture of the computational complexity of the various detectors is provided in Table I for three different channels [(4,4), (6,6), and (8,8)] and 4-QAM modulation. The k flop estimates displayed in Table I were measured using MATLAB V5.3. Even though these complexity measurements depend on the specific algorithm implementations used, they provide more insight than the asymptotic $\mathcal{O}(\cdot)$ results presented in earlier sections. The ML detector was implemented by means of Fincke

and Phost's sphere-decoding (FSPD) algorithm [4]. The complexity of FPSD was measured at an SNR of 10 dB. The complexity of the other schemes is independent of the SNR.

Table I(a) shows the complexity of the operations that have to be performed once for an entire data block during which the channel is constant (i.e., "overhead" operations like calculation of $\mathbf{H}^\# = (\mathbf{H}^H \mathbf{H})^{-1} \mathbf{H}^H$ and \mathbf{v}_{M_T}). Table I(b) shows the complexity of the operations that have to be performed once for each time instant or data vector (i.e., data detection operations like determination of \mathcal{D}_+ and ψ_i^2 , not including the overhead computations considered in Table I(a)). Each table presents only one value for both linear detectors (ZF and MMSE; denoted as "lin.") and only one value for both linear detectors combined with the SPA (ZF/SP and MMSE/SP; denoted as "lin./SP") because there is virtually no difference in complexity. Note that in Table I(a), the values for FPSD and lin. are equal since both algorithms have the same overhead complexity (computation of $\mathbf{H}^\#$). Similarly, in addition, the values for $\widetilde{\text{ML}}$ and lin./SP are equal (same overhead of calculating $\mathbf{H}^\#$ and \mathbf{v}_{M_T}).

The complexity of FPSD strongly depends on the channel realization and the SNR; for a bad channel realization, it exceeds the average complexity by a large amount. Thus, in addition to the average FPSD complexity, Table I(b) shows the maximum FPSD complexity obtained during 10 000 simulation runs at an SNR of 10 dB. A maximum complexity is also provided for lin./SP and NC/SP; it refers to the case where all boundary lines intersect the intersection circle \mathcal{I} (cf. Case 1 in Section VI-A).

From Table I, the following conclusions can be drawn.

- The maximum complexity of FPSD is much larger than its average complexity. For practical system design, a limitation of the maximum complexity is desirable. However, FPSD with limited maximum complexity is no longer an exact implementation of ML decoding.
- In contrast to FPSD, the complexity of $\widetilde{\text{ML}}$ detection is independent of the channel realization and the SNR. According to Table I(b), the complexity of $\widetilde{\text{ML}}$ detection is smaller than the average complexity of FPSD and only a fraction of the maximum complexity of FPSD.
- For both FPSD and $\widetilde{\text{ML}}$, the complexity of the operations to be performed in each time instant is much larger than for the other algorithms.
- For the lin./SP detectors, the complexity of the computations performed once per data block is lower than for the

NC detectors but slightly higher than for the linear detectors and FPSD (due to the additional calculation of \mathbf{v}_{MT}).

- For the lin./SP detectors, the average complexity of the computations performed once per data vector is about twice that of the NC detectors but significantly lower (by a factor of about 25...33) than that of FPSD.
- The overall complexity of the SPA detectors is higher than that of the standard suboptimal detectors but much lower than that of FPSD and ML. In particular, the complexity of NC-MMSE/SP is just a fraction of the complexity of both FPSD and ML, even though NC-MMSE/SP achieves near-ML performance, as shown in Section VII-B.

VIII. SUMMARY AND CONCLUSIONS

The starting point of this paper was an analysis of the effects of bad (poorly conditioned) channels on suboptimal detectors for a MIMO spatial multiplexing system. The performance of all standard suboptimal detection schemes severely degrades compared to the performance of the maximum-likelihood (ML) detector when bad channel realizations occur. We found that this inferior performance is due to the inability of linear detectors to properly adapt their decision regions to the noise statistics. In addition, bad channels lead to a high computational complexity of Fincke and Phost's sphere-decoding algorithm for ML detection.

Based on an idealized approximate model for bad channels that captures the bad channel effects in a simplified form, we then presented new detection methods that are robust to bad channels. The initial form of the new detection approach was an efficient ML detector for the idealized bad channel model. Subsequently, we extended this detector to be suitable for arbitrary (nonidealized) channels. Finally, we developed the *sphere-projection algorithm* (SPA) that is a computationally efficient, nonlinear add-on to standard suboptimal detectors. This add-on improves the error-rate and diversity-gain performance of suboptimal detectors by making them robust to bad channel realizations.

Simulations showed that the SPA outperforms ZF-based and MMSE-based nulling and cancelling with optimal layer ordering by achieving higher diversity gains at a comparable computational complexity. The performance of the SPA is best for systems of moderate size (which are of greatest practical interest), whereas for increasing system size, it degrades compared with the performance of ML detection. For example, for a spatial multiplexing system with six transmit antennas and six receive antennas, the SPA is able to yield near-ML performance at just a fraction of the computational complexity of Fincke and Phost's sphere-decoding algorithm for ML detection.

REFERENCES

- [1] V. Tarokh, N. Seshadri, and A. R. Calderbank, "Space-time codes for high data rate wireless communications: performance criterion and code construction," *IEEE Trans. Inform. Theory*, vol. 44, pp. 744–765, Mar. 1998.
- [2] P. W. Wolniansky, G. J. Foschini, G. D. Golden, and R. A. Valenzuela, "V-BLAST: an architecture for realizing very high data rates over the rich-scattering wireless channel," in *Proc. URSI Int. Symp. Signals, Syst., Electron.*, Sept. 1998, pp. 295–300.

- [3] W.-J. Choi, R. Negi, and J. Cioffi, "Combined ML and DFE decoding for the V-BLAST system," in *Proc. IEEE Int. Conf. Commun.*, New Orleans, LA, June 2000, pp. 18–22.
- [4] U. Fincke and M. Phost, "Improved methods for calculating vectors of short length in a lattice, including a complexity analysis," *Math. Comput.*, vol. 44, pp. 463–471, Apr. 1985.
- [5] L. Zheng and D. Tse, "Diversity and freedom: a fundamental tradeoff in multiple antenna channels," *IEEE Trans. Inform. Theory*, submitted for publication.
- [6] S. M. Kay, *Fundamentals of Statistical Signal Processing: Detection Theory*. Upper Saddle River, NJ: Prentice-Hall, 1998.
- [7] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed. Baltimore: Johns Hopkins Univ. Press, 1996.
- [8] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. Englewood Cliffs, NJ: Prentice-Hall, 1993.
- [9] B. Hassibi and H. Vikalo, "On the expected complexity of sphere decoding," in *Proc. 35th Asilomar Conf. Signals, Syst., Comput.*, Pacific Grove, CA, Nov. 2001, pp. 1051–1055.
- [10] A. Edelman, "Eigenvalues and condition numbers of random matrices," Ph.D. dissertation, Mass. Inst. Technol., Cambridge, MA, 1989.
- [11] M. Damen, A. Chkeif, and J. Belfiore, "Lattice code decoder for space-time codes," *IEEE Commun. Lett.*, vol. 4, pp. 161–163, May 2000.
- [12] C. W. Therrien, *Discrete Random Signals and Statistical Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1992.



Harold Artés (S'98) received the Dipl.-Ing. degree in electrical engineering from Vienna University of Technology, Vienna, Austria, in 1997, where he is currently pursuing the Ph.D. degree.

He is a Research (and partly Teaching) Assistant with the Institute of Communications and Radio-Frequency Engineering, Vienna University of Technology. He participated in the European Commission funded IST project ANTIUM and co-authored two patents. His research interests include MIMO wireless communications, blind estimation and equalization of time-varying channels, and multiuser techniques.



Dominik Seethaler (S'03) received the Dipl.-Ing. degree in electrical engineering from Vienna University of Technology, Vienna, Austria, in 2002, where he is pursuing the Ph.D. degree.

He currently holds a Research and Teaching Assistant position with the Institute of Communications and Radio-Frequency Engineering, Vienna University of Technology. His research interests are in wireless communications with emphasis on MIMO and multiuser techniques.



Franz Hlawatsch (S'85–M'88–SM'00) received the Dipl.-Ing., Dr. techn., and habilitation degrees in electrical engineering/signal processing from Vienna University of Technology, Vienna, Austria, in 1983, 1988, and 1996, respectively.

Since 1983, he has been with the Institute of Communications and Radio-Frequency Engineering, Vienna University of Technology, where he currently holds an Associate Professor position. He was a consultant for Schrack AG from 1983 to 1988 and for AKG GesmbH from 1984 to 1985. From

1991 to 1992, he spent a sabbatical year with the Department of Electrical Engineering, University of Rhode Island, Kingston. From 1999 to 2001, he held one-month Visiting Professor positions with INP/ENSEEIH, Toulouse, France, and IRCCyN, Nantes, France. He authored the book *Time-Frequency Analysis and Synthesis of Linear Signal Spaces* (Boston, MA: Kluwer, 1998) and co-edited the book *The Wigner Distribution—Theory and Applications in Signal Processing* (Amsterdam, The Netherlands: Elsevier, 1997). His research interests are in nonstationary statistical and time-frequency signal processing and in wireless communications.

Dr. Hlawatsch is currently serving as an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING.