Efficient MIMO Detection by Successive Projection

Karen Su and Ian J. Wassell
Laboratory for Communication Engineering
Cambridge University Engineering Department
Trumpington Street, Cambridge, CB2 1PZ
Email: {ks349, ijw24}@eng.cam.ac.uk

***Abstract*— This paper presents a new MIMO detector based on successive projection of the received signal onto the faces of the lattice induced by the channel matrix. Analysis of the relationship between successive algorithms like V-BLAST and the Maximum Likelihood (ML) sphere decoder, which share a common underlying tree structure, leads to the design of the Successive Projection Algorithm (SPA). Although it is a suboptimal detector, we prove theoretically that when ML detection can be realized in a successive manner, i.e., without back-tracking, ML performance is also achieved by the SPA. The advantageous implications of this result are illustrated via simulation of the average bit error rates attained over a 4×4 MIMO flat fading channel. For instance, at a target error rate of 10^{-3} using 16-QAM modulation, a 2.8dB improvement over the popular V-BLAST detector is observed at a comparable complexity. We also demonstrate a parameterized SPA, which offers performance profiles approaching that of an ML detector.**

***Index Terms*— Nearest lattice point search, maximum-likelihood detection, suboptimal detection, (linear) MIMO systems, successive cancellation.**

I. Introduction

With the advent of the space-time communications paradigm, the detection of QAM-modulated signals observed at the output of Multiple Input Multiple Output (MIMO) channels has received a considerable amount of attention. Popular leading algorithms arising from this body of work include those based on *successive* strategies, e.g., the suboptimal V-BLAST scheme [1], [2], the ML sphere decoder [3]–[6], and their variants, those based on *parallel* strategies [7], and hybrid techniques such as the Chase detector [8].

Underlying successive approaches is the decomposition of the overall detection problem into a set of smaller subproblems of decreasing dimension. For instance, application of the QR factorization to the channel matrix leads to an upper triangular channel structure that enables the detector to achieve this desirable break-down. The decomposition of the overall detection problem can be concisely captured in a search tree structure, which is central to the current work and will be discussed in greater detail in Section II.

The order of symbol detection is also known to play a key role in the performance and/or complexity of successive schemes [2], [4], [9]. For instance, in V-BLAST, poor ordering leads to degraded preformance due to error propagation, and in sphere decoding, to greatly increased decoding times. In this work, we present new results on the impact of ordering on successive detection algorithms. Drawing on our findings, we develop an efficient new detector that is capable of attaining performance profiles ranging from that offered by the popular V-BLAST variant, using MMSE nulling with cancellation and ordering, to that of the ML sphere decoder. Our flexible new detector is governed by a single parameter.

We begin our presentation in Section II with some mathematical preliminaries. Next, in Section III we study the behaviour of successive detection schemes within the context of the tree structure. Specifically we consider the distinct instances of the tree induced by different symbol detection orderings and identify desirable properties for achieving ML or near-ML performance. This analysis leads to the design of the Successive Projection Algorithm (SPA) as well as an iterative variant, both formalized in Section IV. Section V evaluates the performance of the new detectors and concludes with brief comments on their complexities.

II. Preliminaries

In this paper we are concerned with problems that can be modelled as the minimization of the squared Euclidean distance metric to a target point $\mathbf{r}$ over an M-dimensional finite discrete search set $\mathcal{X}^M$ with $|\mathcal{X}| = B$:

$$\mathbf{s}_* = \underset{\mathbf{s} \in \mathcal{X}^M}{\operatorname{argmin}} |\mathbf{r} - \mathbf{H}\mathbf{s}|^2, \tag{1}$$

where $\mathbf{r} \in \mathbb{R}^N$, $\mathbf{H} \in \mathbb{R}^{N \times M}$, and the optimization variables are the elements of $\mathbf{s}$.[1] Examples of such problems include ML detection of lattice coded signals and QAM modulated signals transmitted over MIMO flat fading channels, frequency selective fading channels, or multi-user channels. Both suboptimal and optimal solutions to (1) are considered.

We assume that $M \leq N$, and that $\mathbf{H}$ is of full rank M. For communication over MIMO flat fading channels, this assumption means that there are at least as many receive (N) as transmit (M) antennas. The following notational conveniences are applied: Let $\mathbf{a}_i$ denote the i^{th} column vector of matrix $\mathbf{A}$ and z_i the i^{th} element of vector $\mathbf{z}$. We denote by $\mathcal{I}$ the index set $\{1, 2, \ldots, M\}$. To distinguish between a variable itself and

This work was supported by Universities UK, the Cambridge Commonwealth Trust, the Natural Sciences and Engineering Research Council of Canada and Trinity Hall.

[1] The complex case where $\mathbf{s} \in (\mathcal{X}^2)^M$ is a vector of M QAM modulated signals, $\mathbf{r} \in \mathbb{C}^N$ and $\mathbf{H} \in \mathbb{C}^{N \times M}$ can be written as an equivalent problem in twice the number of real dimensions, i.e., with $\mathbf{r} \in \mathbb{R}^{2N}$ and $\mathbf{H} \in \mathbb{R}^{2N \times 2M}$.

its value, we use the underline notation $\underline{s}_i$ or $\underline{\mathbf{s}}$ to refer to a variable, and $s_i \in \mathcal{X}$ or $\mathbf{s} \in \mathcal{X}^M$ to indicate a particular value.

Given matrix $\mathbf{H}$ and alphabet $\mathcal{X}$, we define the *finite lattice* of points in the search set as

$$\mathcal{L} \triangleq \left\{ \mathbf{z} \,\middle|\, \mathbf{z} = \mathbf{H}\mathbf{s},\, \mathbf{s} \in \mathcal{X}^M \right\}. \tag{2}$$

There are B^M lattice points in $\mathcal{L}$. It can be decomposed into any one of M collections of B sub-lattices, each comprised of B^{M-1} lattice points embedded in one of B parallel *affine sets*.[2] For instance, given $i \in \mathcal{I}$, the i^{th} collection $\{\mathcal{F}_i(s_i) \mid s_i \in \mathcal{X}\}$ contains B affine sets defined as

$$\mathcal{F}_i(s_i) \triangleq \left\{ \mathbf{z} \,\middle|\, \left\langle \mathbf{z} - \mathbf{h}_i s_i, (\mathbf{H}^{-1})_i^T \right\rangle = 0 \right\}, \tag{3}$$

with $\langle \mathbf{a}, \mathbf{b} \rangle$ denoting the inner product.

We denote the *orthogonal projection* of a vector $\mathbf{y}$ onto affine set $\mathcal{F}_i(s_i)$ as $\text{proj}_{\mathcal{F}_i(s_i)}(\mathbf{y})$. and the corresponding *orthogonal distance* is then

$$d(\mathbf{y}, \mathcal{F}_i(s_i)) \triangleq d\left(\mathbf{y}, \text{proj}_{\mathcal{F}_i(s_i)}(\mathbf{y})\right). \tag{4}$$

Note that $\text{proj}_{\mathcal{F}_i(s_i)}(\mathbf{y})$ is the point in the affine set that is closest in Euclidean distance to $\mathbf{y}$. Algebraically, $\mathcal{F}_i(s_i)$ contains the feasible set of lattice points satisfying constraint $\underline{s}_i = s_i$ and is of dimension $M-1$.

Extending (3), given *constraint index set* $\mathcal{J} = \{i_L, \ldots, i_1\}$ of size L, we define the $(M-L)$-dimensional affine sets containing the feasible sets of lattice points satisfying constraints $\underline{s}_{i_l} = s_{i_l}$, $l = 1, \ldots, L$ as

$$\mathcal{F}_{\mathcal{J}}(\mathbf{s}_{\mathcal{J}}) \triangleq \bigcap_{l=1}^{L} \mathcal{F}_{i_l}(s_{i_l}), \tag{5}$$

and we highlight a useful property of the resulting distances:

Lemma 1: Given index set $\mathcal{K} \subseteq \mathcal{J}$, the squared orthogonal distance from a point $\mathbf{y} \in \mathbb{R}^N$ to affine set $\mathcal{F}_{\mathcal{J}}(s_{\mathcal{J}}) \subset \mathbb{R}^N$ can be written as

$$d^2(\mathbf{y}, \mathcal{F}_{\mathcal{J}}(\mathbf{s}_{\mathcal{J}})) = d^2(\mathbf{y}, \mathbf{y}') + d^2\Big(\mathbf{y}', \text{proj}_{\mathcal{F}_{\mathcal{J}}(\mathbf{s}_{\mathcal{J}})}(\mathbf{y})\Big), \tag{6}$$

where $\mathbf{y}' = \text{proj}_{\mathcal{F}_{\mathcal{K}}(\mathbf{s}_{\mathcal{K}})}(\mathbf{y})$.

Proof: Because $\mathcal{K} \subseteq \mathcal{J}$, the corresponding affine sets satisfy $\mathcal{F}_{\mathcal{J}}(\mathbf{s}_{\mathcal{J}}) \subseteq \mathcal{F}_{\mathcal{K}}(\mathbf{s}_{\mathcal{K}})$. The result then follows directly by recalling that $\text{proj}_{\mathcal{F}_{\mathcal{K}}(\mathbf{s}_{\mathcal{K}})}(\mathbf{y}) \in \mathcal{F}_{\mathcal{K}}(\mathbf{s}_{\mathcal{K}})$. □

Lemma 1 states that the squared distance from a point to an affine set can be decomposed into two orthogonal components. Because distances are non-negative, we have that $d\left(\mathbf{y}, \mathcal{F}_{\mathcal{J}}(\mathbf{s}_{\mathcal{J}})\right) \geq d\left(\mathbf{y}, \mathcal{F}_{\mathcal{K}}(\mathbf{s}_{\mathcal{K}})\right)$ if $\mathcal{K} \subseteq \mathcal{J}$.

Since the cost function of (1) can be expressed as $|\mathbf{r} - \mathbf{H}\mathbf{s}|^2 = d^2\left(\mathbf{r}, \mathcal{F}_{\mathcal{I}}(\mathbf{s})\right)$, applying Lemma 1 recursively, with index sets $\{i_1\} \subset \{i_2, i_1\} \subset \ldots \subset \{i_{M-1}, \ldots, i_1\} \subset \{i_M, \ldots, i_1\}$, gives a representation of its values as the weights of the leaf nodes of a $(M+1)$-level B-ary tree structure as shown in Fig. 1.

The *nodes* of the tree are arranged in $M+1$ *levels*, numbered from the *root* at level 0 to the *leaves* at level M.

[2]Recall that an affine set $\mathcal{M} \subset \mathbb{R}^N$ is a set such that $\mathcal{M} = \mathcal{S} + \mathbf{a}$ for some subspace $\mathcal{S} \subset \mathbb{R}^N$ and offset $\mathbf{a} \in \mathbb{R}^N$. See [10, Sec. 1] for more details.

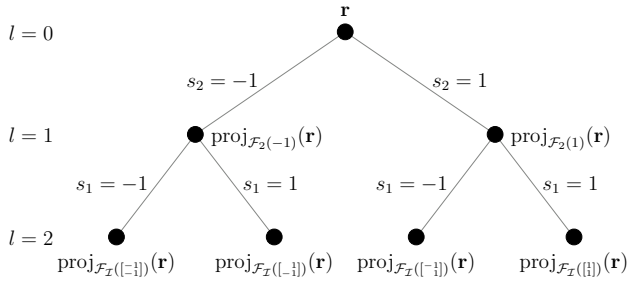


Fig. 1. A B-ary tree structure for $M = 2$ and $\mathcal{X} = \{-1, 1\}$ $(B = 2)$.

Associated with each is the projection of $\mathbf{r}$ onto an $(M-L)$-dimensional affine set defined by the constraint index set $\mathcal{K}^L = \{i_L, \ldots, i_1\}$, where L is its level. We define the *weight* of a node σ as the squared distance from $\mathbf{r}$ to this projection. Because the constraint index sets grow along all *paths* from the root, by Lemma 1 the corresponding sequence of node weights is non-decreasing.

Recall that the values of the cost function are represented as the weights of the leaf nodes. The leaf having the smallest weight then corresponds to an ML solution. We emphasize that index set $\{i_M, \ldots, i_1\}$ is one of $M!$ permutations of $\mathcal{I}$. More formally, we call $\Pi = \{i_M, \ldots, i_1\}$ an *ordering*[3] and say that each ordering *induces* an instance of the tree $\mathcal{T}(\Pi)$.

Finally, we make use of the following quantities: σ_*, the weight of the leaf node associated with an ML solution, and ν_*, the number of nodes in the search tree whose weights satisfy $\sigma \leq \sigma_*$. It is well-known that ν_* reflects the computational efficiency of optimal detectors [6], [11] and in this work we discuss its implications for sub-optimal schemes.

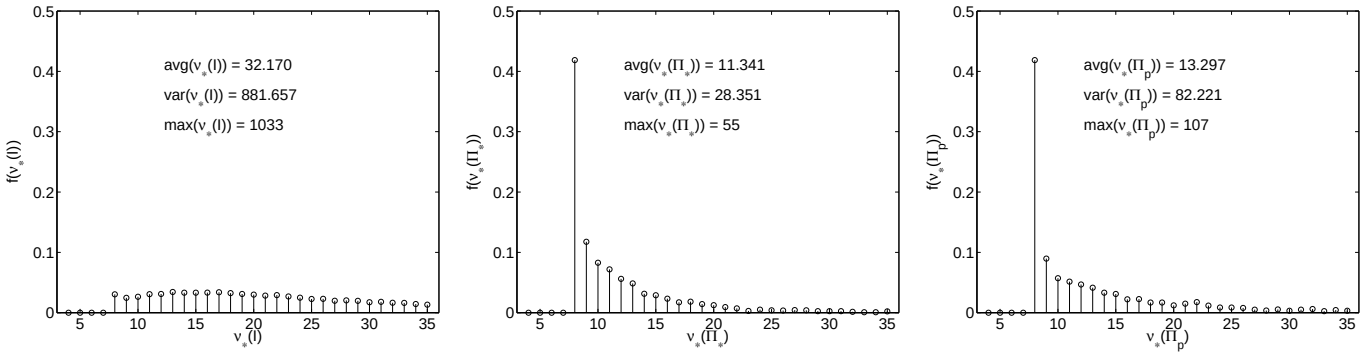
III. Achieving ML Detection Successively

Many representative MIMO detection techniques, e.g., [1], [3], seek to determine the nearest (or a nearby, in the case of sub-optimal approaches) lattice point to target $\mathbf{r}$ by successively exploring the affine sets containing lattice points of $\mathcal{L}$. At each stage L, given index set $\mathcal{K}^L$ of previously constrained variables, these detectors perform the following tasks:

1) Select an unconstrained variable $\underline{s}_{i_L}$, $i_L \in \mathcal{I} \setminus \mathcal{K}^L$.
2) Detect symbol i_L, i.e., assign a value $s_{i_L} \in \mathcal{X}$ to variable $\underline{s}_{i_L}$, or interpreted geometrically, constrain subsequent detection stages to operate within affine set $\mathcal{F}_{i_L}(s_{i_L})$.
3) Add i_L to the constraint index set, $\left(\mathcal{K}^{L+1} = \{i_L, \mathcal{K}^L\}\right)$.

Steps 1-3 correspond to descending a level in the tree, where $\mathcal{K}^L$ is the constraint index set associated with a node at level L. Thus the behaviour of a *single-pass* successive algorithm that detects each of the symbols in some order, e.g., V-BLAST, can be visualized as a simple path from the root ($\mathcal{K}^0 = \emptyset$) to a leaf ($\mathcal{K}^M = \Pi$). ML algorithms like the sphere decoder perform the same set of operations at each stage. However they are permitted to back-track and therefore may potentially explore more than one path to more than one leaf.

[3]We index the elements of Π in this way because many MIMO detectors apply a QR factorization and as a result detect symbols in the reverse order of the underlying index set. Therefore under Π the order becomes $s_{i_1}, \ldots, s_{i_M}$.



(a) Random selection strategy. (b) Optimal selection strategy. (c) Proposed selection strategy.

Fig. 2. Empirical estimates of the probability density functions $f_{\nu_*}(\nu_*)$ for various selection strategies over a 4:4 MIMO flat fading channel (complex, $M = 8$) at an SNR of 5dB using 16-QAM modulation, based on 2,500 independent realizations of received signal vectors $\mathbf{r}$ and channel matrices $\mathbf{H}$. The optimal selection strategies were determined by brute force simulation of $\nu_*(\Pi)$ over all $M!$ possible orderings.

To investigate performance limits of single-pass successive detectors, we conducted the following study: Assuming that Step 2 is a linear detector, i.e., quantizes an appropriate decision statistic to the nearest element of alphabet $\mathcal{X}$, we then considered the effect of applying different strategies in Step 1. Since there is a one-to-one correspondence between the sets of selection strategies and orderings, we can uncover properties of a selection strategy via study of the tree $\mathcal{T}(\Pi)$ induced by its associated ordering.

To quantify the "goodness" of a particular selection strategy, we use the number of nodes in $\mathcal{T}(\Pi)$ whose weights satisfy $\sigma \leq \sigma_*$, denoted $\nu_*(\Pi)$. The rationale behind this metric is that if for a given target vector $\mathbf{r}$ and channel matrix $\mathbf{H}$ there exists Π such that $\nu_*(\Pi) = M$, then there also exists an ML detector that behaves as a single-pass detector. (See Appendix, Lemma 2 and [11] for proofs.) Therefore, as we present in Section IV and show in the Appendix, under these conditions a single-pass successive scheme can achieve ML performance.

We report three interesting findings: First, that applying a random strategy does not yield good values of $\nu_*(\Pi)$ (Fig. 2(a)). Secondly, that the distribution realized by an optimal selection strategy is favourable, i.e., the optimal ordering

$$\Pi_* \triangleq \underset{\Pi}{\operatorname{argmin}}\, \nu_*(\Pi) \tag{7}$$

often satisfies $\nu_*(\Pi_*) = M$ (Fig. 2(b)). Finally, that the distribution arising from the strategy to be proposed is near to that observed in the best case (Fig. 2(c)). The selection strategy used by the V-BLAST detector [2] was also simulated and it produced a distribution near to that of the random case.

IV. The Successive Projection Algorithm (SPA)

The preceding investigation inspires us to ask what it means for there to exist an ordering such that ML detection is achievable in a single-pass through $\mathcal{T}(\Pi)$. The set of ordering-induced trees share a common root, having weight 0, as well as common leaf nodes, whose weights correspond to the values of the cost function of (1). However, they may differ significantly in the intermediate weights of non-leaf nodes, which leads to widely varying values of $\nu_*(\Pi)$.

When ML detection can be achieved in a single-pass, the weights of all non-leaf nodes that do not lie along the solution path have weights satisfying $\sigma > \sigma_*$. For instance, a single-pass successive detector traverses a simple path from the root node to a leaf in the order specified by Π. Determining which branch to take from level $L-1$ to L is equivalent to assigning one of B values to variable $\underline{s}_{i_L}$. Given constraint index set $\mathcal{K}^L = \{i_L, \mathcal{K}^{L-1}\}$ and previously applied constraint values $\mathbf{s}_{\mathcal{K}^{L-1}}$, let α_{i_L} and β_{i_L} be elements of $\mathcal{X}$ such that

$$d^2\Bigg(\mathbf{r}, \mathcal{F}_{\mathcal{K}^L}\left(\begin{bmatrix}\alpha_{i_L}\\ \mathbf{s}_{\mathcal{K}^{L-1}}\end{bmatrix}\right)\Bigg) \leq d^2\Bigg(\mathbf{r}, \mathcal{F}_{\mathcal{K}^L}\left(\begin{bmatrix}\beta_{i_L}\\ \mathbf{s}_{\mathcal{K}^{L-1}}\end{bmatrix}\right)\Bigg) \tag{8}$$

$$\leq d^2\Bigg(\mathbf{r}, \mathcal{F}_{\mathcal{K}^L}\left(\begin{bmatrix}\gamma_{i_L}\\ \mathbf{s}_{\mathcal{K}^{L-1}}\end{bmatrix}\right)\Bigg), \tag{9}$$

for all $\gamma_{i_L} \in \mathcal{X} \setminus \{\alpha_{i_L}, \beta_{i_L}\}$. Again assuming a linear detector, α_{i_L} is assigned to $\underline{s}_{i_L}$ and the corresponding branch is traversed. In addition, we call $d^2\Bigg(\mathbf{r}, \mathcal{F}_{\mathcal{K}^L}\left(\begin{bmatrix}\beta_{i_L}\\ \mathbf{s}_{\mathcal{K}^{L-1}}\end{bmatrix}\right)\Bigg)$ the *base weight* of the most favourable *path not taken* at level L. It can then be shown that to achieve ML detection in a single-pass, it suffices that these base weights be greater than σ_* at all levels of the tree.

Therefore intuition suggests that selection strategies where the base weights of the best paths not taken are large at all levels are most favourable for use with successive detectors. In theory, it would be best to maximize the minimum base weight over all levels. However because this global optimization problem can not be efficiently solved, we propose the following local selection criterion:

$$i_L = \underset{i \in \mathcal{I} \setminus \mathcal{K}^L}{\operatorname{argmax}}\, d^2\Bigg(\mathbf{r}, \mathcal{F}_{\mathcal{K}^L}\left(\begin{bmatrix}\beta_{i_L}\\ \mathbf{s}_{\mathcal{K}^{L-1}}\end{bmatrix}\right)\Bigg). \tag{10}$$

Pseudocode for the basic SPA detector is given in Algorithm 1.

Applying selection criterion (10) enables us to prove the following important result on its performance:

Theorem 1: If there exists an ordering Π such that $\nu_*(\Pi) = M$, then the SPA returns an ML solution.

Proof: See Appendix. □

Algorithm 1 SPA$(\mathbf{r}, \mathbf{H}, M, \mathcal{X})$

```
1:  I ⟵ {1, 2, ..., M}                                  Initialize index set
2:  y ⟵ r                                               Initialize target
3:  G ⟵ (H^{-1})^T                                      Compute inverse
4:  for each level L from 1 to M do
5:      for each index i in set I do
6:          α_i ⟵ argmin_{x∈X} |⟨y, g_i⟩ − x|           Find nearest sets
7:          β_i ⟵ argmin_{x∈X\α_i} |⟨y, g_i⟩ − x|       Second nearest
8:          δ_i ⟵ d^2 (y, F_i(β_i))                     Compute distances
9:      end for
10:     i_L ⟵ argmax_{i∈I} δ_i                          Select variable index
11:     ŝ_{i_L} ⟵ α_{i_L}                               Detect symbol
12:     I ⟵ I \ i_L                                     Remove from index set
13:     y ⟵ proj_{F_{i_L}(α_{i_L})}(y) − h_{i_L} α_{i_L}   Project and shift
14:     for each index i in set I do                    Project inverse
15:         g_i ⟵ proj_{F_{i_L}(0)}(g_i)
16:     end for
17: end for
18: Return ŝ
```

We refer to this algorithm as the basic SPA because it is a single-pass detector. A natural extension is to keep track of the ζ best paths not taken, i.e., those having the smallest base weights. The extended SPA is then a ζ-pass approach, where the main path and ζ paths not (initially) taken are traversed in search of the ML solution. It should be clear that in the limit as $\zeta \longrightarrow B^M$, all paths and nodes in the search tree are explored and therefore the performance of the extended SPA approaches that of an ML detector.

V. Performance Evaluation and Conclusions

Fig. 3 illustrates the performance profiles offered by the basic ($\zeta = 0$) and extended SPA for $\zeta = 1, 2, 4$. These are shown alongside those of an ML detector and of the V-BLAST scheme with MMSE nulling, cancellation and ordering. The system simulated is a 4×4 (complex) MIMO spatially-uncorrelated flat fading channel with inputs drawn from the 16-QAM symbol alphabet. We assume perfect channel state information at the receiver. Observe that as the complexity of the SPA is increased via parameter ζ, i.e., by exploring more paths not taken, its performance improves. For the system considered here at a target bit error rate of 10^{-3}, there is a 7.5dB gap between the V-BLAST and ML performance profiles. Setting ζ to 0, 1, 2 and 4 allows the SPA to reduce this gap to 4.7dB, 2.4dB, 1.2dB and 0.6dB, respectively.

The time complexity of Algorithm 1 with $\zeta = 0$ is $\mathcal{O}(M^3)$, roughly comparable to one matrix inversion and a few QR factorizations. For the extended SPA, this one matrix inversion is still required, and the complexity of the remaining operations is linear in ζ.

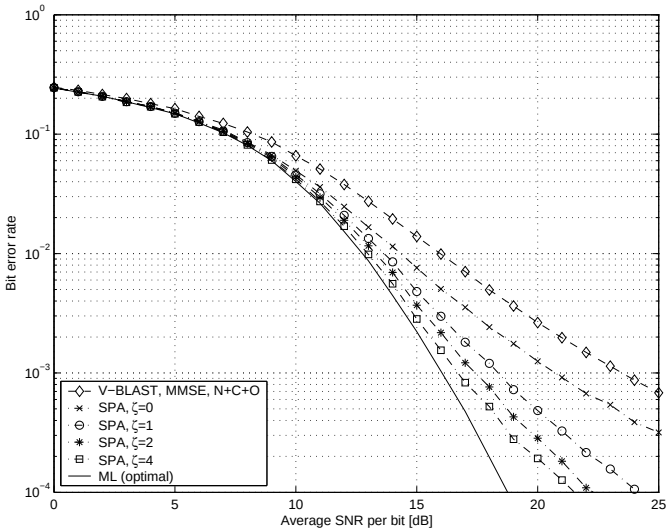


Fig. 3. Bit error rate vs. average SNR per bit for 4:4 systems in independent flat Rayleigh fading using uncoded coherent 16-QAM modulation. The performance profiles of the proposed decoders (SPA with $\zeta = 0, 1, 2, 4$) are compared to those of the V-BLAST scheme (with MMSE detection, nulling, cancellation and ordering) and of the ML sphere decoder.

Bit error rate performance is an essential validator of sub-optimal detection schemes. In this work, we also consider how their behaviour with respect to the underlying search trees relates to that of ML detectors. Our analysis leads to an advantageous and efficient new detector called the SPA. In its development, we apply a general detection framework and prove a useful theoretical result on the ability of single-pass techniques to achieve ML detection. We simulated the bit error rates attained by the SPA for several values of parameter ζ, and demonstrated that all variants offer improved performance compared to the popular V-BLAST scheme. In practice, the gap from ML performance is observed to shrink significantly even for very small values of ζ.

References

[1] G. J. Foschini, "Layered space-time architecture for wireless communication in a fading environment when using multiple antennas," *Bell Labs Technical Journal*, vol. 1, no. 2, pp. 41–59, September 1996.

[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 *International Symposium on Signals, Systems, and Electronics*, September 1998, pp. 295–300.

[3] E. Viterbo and J. Boutros, "A universal lattice code decoder for fading channels," *IEEE Transactions on Information Theory*, vol. 45, no. 5, pp. 1639–1642, July 1999.

[4] M. O. Damen, H. E. Gamal, and G. Caire, "On maximum-likelihood detection and the search for the closest lattice point," *IEEE Transactions on Information Theory*, vol. 49, no. 10, pp. 2389–2402, October 2003.

[5] A. M. Chan and I. Lee, "A new reduced-complexity sphere decoder for multiple antenna systems," in *IEEE International Conference on Communications*, vol. 1, April 2002, pp. 460–464.

[6] B. Hassibi and H. Vikalo, "On the sphere decoding algorithm: Part I, The expected complexity," *To appear in IEEE Transactions on Signal Processing*, 2004.

[7] Y. Li and Z. Luo, "Parallel detection for V-BLAST system," in *IEEE Int'l Conf. on Communications*, May 2002.

[8] D. W. Waters and J. R. Barry, "The Chase family of detection algorithms for MIMO channels," in *IEEE Global Communications Conf. (to appear)*, December 2004.

[9] K. Su and I. J. Wassell, "An enhanced ordering for efficient sphere decoding," in *IEEE Int'l Conf. on Communications (to appear)*, May 2005.
[10] R. T. Rockafellar, *Convex analysis*. Princeton University Press, 1970.
[11] K. Su, C. N. Jones, and I. J. Wassell, "An automatic sphere decoder," *Submitted to IEEE Transactions on Information Theory*, 2004.

APPENDIX

To prove Thm. 1, we begin by defining the central notion of *uniquely determinable* variables.

Definition 1: Given received signal vector $\mathbf{r}$, channel matrix $\mathbf{H}$, finite alphabet $\mathcal{X}$, and σ_*, the squared distance from $\mathbf{r}$ to the nearest lattice point in $\mathcal{L}$, let the first and second nearest $(M-1)$-dimensional affine sets to $\mathbf{r}$ in collection i be given by $\mathcal{F}_i(\alpha_i)$ and $\mathcal{F}_i(\beta_i)$, respectively, where α_i and β_i are elements of $\mathcal{X}$. Then we call variable $\underline{s}_i$ *uniquely determinable (u.d.)* if

$$\sigma_* < d^2\left(\mathbf{r}, \mathcal{F}_i(\beta_i)\right). \tag{11}$$

Because the squared distance to the second nearest affine set in collection i is greater than σ_*, that to all lattice points *except* those satisfying constraint $\underline{s}_i = \alpha_i$ are also greater than σ_*. Therefore $\underline{s}_i$ can be determined uniquely, i.e., its optimal value is α_i, regardless of decisions made for the other symbols.

In the derivations to follow, let $\mathcal{D}$ denote the set comprising the indicies of the u.d. variables and let an ordering Π be called *ensemble optimal (e.o.)* if $\nu_*(\Pi) = M$. Note that given problem parameters $\mathbf{r}$ and $\mathbf{H}$, an e.o. ordering may not exist, i.e., if $\nu_*(\Pi) > M$ for all Π. However, over the entire ensemble of problem realizations $\min_{\mathbf{r},\mathbf{H},\Pi} \nu_*(\Pi) = M$. The term ensemble optimal refers to those orderings that enable this realization-independent optimum to be achieved.

Lemma 2: There exists an ensemble optimal ordering $\Pi = \{i_M, \ldots, i_1\}$ if and only if exactly one node on each non-leaf level of the induced search tree satisfies $\sigma \leq \sigma_*$.

Proof: Because the leaf associated with the ML solution always satisfies $\sigma \leq \sigma_*$, so must its M ancestors. These lie along a path, one on each non-leaf level of the tree. Also, since $\nu_*(\Pi) = M$, no other nodes may satisfy $\sigma \leq \sigma_*$. □

Lemma 2 captures the relationship between the existence of an ensemble optimal ordering and the operation of an ML detector called the Automatic Sphere Decoder (ASD) [11]. In the context of the search tree, the ASD achieves ML detection while only processing those nodes whose weights satisfy $\sigma \leq \sigma_*$. Thus, when an e.o. ordering exists, the ASD behaves as a single-pass detector. Next we consider the implication of an optimal ordering on the existence of u.d. variables.

Lemma 3: If there exists an ensemble optimal ordering $\Pi = \{i_M, \ldots, i_1\}$, then $\mathcal{D} \neq \emptyset$.

Proof: From Lemma 2, we have that only one node on each non-leaf level of the tree satisfies $\sigma \leq \sigma_*$. In particular, at level 1, the squared distance to the second nearest affine set in collection i_1 must then satisfy $d^2\left(\mathbf{r}, \mathcal{F}_{i_1}(\beta_{i_1})\right) > \sigma_*$,. Therefore $i_1 \in \mathcal{D}$ and $\mathcal{D} \neq \emptyset$. □

Having established a sufficient condition for the existence of u.d. variables at the first stage of successive detection, we then observe the following about the behaviour of the SPA: Since it computes the index of the first symbol to detect according to $i_1 = \operatorname{argmax}_{i \in \mathcal{I}} d^2\left(\mathbf{r}, \mathcal{F}_{i_1}(\beta_{i_1})\right)$, as long as $\mathcal{D} \neq \emptyset$, it must be the case that $i_1 \in \mathcal{D}$. Thus, if an e.o. ordering exists, Algorithm 1 selects a u.d. variable for detection in its first stage.

At first glance, it may seem to follow immediately that in such cases Algorithm 1 selects a u.d. variable at all stages. However, there is a small wrinkle in that $\mathcal{D}$ may contain more than one index. Therefore, the last tool that we need in order to construct a recursive argument is to consider the consequence of selecting *any* u.d. variable at the first stage.

Lemma 4: If there exists an ensemble optimal ordering Π and a set of alternate orderings $\Pi^i = \{\Pi \setminus i, i\}$, $i \in \mathcal{D}$, then Π^i is ensemble optimal for all $i \in \mathcal{D}$.

Proof: It suffices to show that only one node on each non-leaf level of the trees $\mathcal{T}(\Pi^i)$, $i \in \mathcal{D}$ satisfies $\sigma \leq \sigma_*$.

Let j denote the position of i in Π, then the alternate orderings are given by $\Pi^i = \{\Pi_1 \ldots, \Pi_{j-1}, \Pi_{j+1}, \ldots, \Pi_M, i\}$. Since $\underline{s}_i$ is a u.d. variable, only one node on level 1 satisfies $\sigma \leq \sigma_*$. To consider nodes on levels $L = 2, \ldots, M-1$, first recall that weights are non-decreasing along paths from the root, therefore only the descendants of the one level 1 node satisfying $\sigma \leq \sigma_*$ need be considered.

For levels $L \leq M - j$, we apply Lemma 1 with index sets $\mathcal{K}^L \setminus i \subset \mathcal{K}^L$ to obtain that

$$d^2\left(\mathbf{r}, \mathcal{F}_{\mathcal{K}^L}\left(\mathbf{s}_{\mathcal{K}^L}\right)\right) \geq d^2\left(\mathbf{r}, \mathcal{F}_{\mathcal{K}^L \setminus i}\left(\mathbf{s}_{\mathcal{K}^L \setminus i}\right)\right). \tag{12}$$

Since the smaller constraint index set can be written as $\mathcal{K}^L \setminus i = \{\Pi^i_{M-L}, \ldots, \Pi^i_{M-1}\} = \{\Pi_{M-L+1}, \ldots, \Pi_M\}$, $\operatorname{proj}_{\mathcal{F}_{\mathcal{K}^L \setminus i}\left(\mathbf{s}_{\mathcal{K}^L \setminus i}\right)}(\mathbf{r})$ is a level $L-1$ node in tree $\mathcal{T}(\Pi)$, only one of which satisfies $\sigma \leq \sigma_*$. Combining this observation with (12), we have that the node weights on these levels of $\mathcal{T}(\Pi^i)$ are at least as large as their counterparts in $\mathcal{T}(\Pi)$. Thus at most one node per level can satisfy $\sigma \leq \sigma_*$.

For levels $L > M - j$, the constraint index sets induced by Π^i and Π are identical. Therefore the node weights on these levels of $\mathcal{T}(\Pi^i)$ are equal to their counterparts on the same levels of $\mathcal{T}(\Pi)$, only one of which per level satisfies $\sigma \leq \sigma_*$.

Finally, since the leaf node associated with the ML solution (and its ancestors) must meet the required condition, it follows that exactly one node on each non-leaf level satisfies $\sigma \leq \sigma_*$ and therefore Π^i is ensemble optimal. □

We can now prove the main result:

Theorem 1: If there exists an ensemble optimal ordering Π, then Algorithm 1 returns an ML solution.

Proof: Since an ensemble optimal ordering Π exists, Lemmas 3 and 4 establish that the index of the first symbol s_{i_1} detected by Algorithm 1 matches with the first index of an e.o. ordering $\Pi^p = \{\Pi \setminus i_1, i_1\}$. It then follows that for all nodes along the associated path to the ML solution, there exists an e.o. ordering over the remaining variables.

The preceding argument can be repeated recursively, i.e., to the selection of symbols $s_{i_2}, \ldots, s_{i_M}$ detected by Algorithm 1, until it reaches the ML solution and terminates. □

In other words, when an ensemble optimal ordering exists, Algorithm 1 always detects symbols associated with uniquely determinable variables and therefore its decisions are optimal.