1

MIMO Receive Algorithms

T. Kailath*, H.Vikalo‡, and B. Hassibi‡

*Stanford University

‡California Institute of Technology

Abstract

The optimal detection problem in multi-antenna wireless communication systems often reduces to the problem of finding the least-squares solution to a system of linear equations, where the unknown vector is comprised of integers, but the matrix coefficients and the given vector are real-valued. The problem is equivalent to finding the closest lattice point to a given point and is known to be NP-hard. We review the most commonly used solution techniques, and discuss their computational complexity. Among heuristic algorithms, we focus on the nulling and cancelling techniques, and their fast implementations based on linear estimation theory. We then show that an exact method, the sphere decoding algorithm, often has expected complexity implementable in practical systems. We also describe extensions of sphere decoding techniques to receivers that make use of the so-called soft information.

Keywords – wireless communications, multi-antenna systems, integer least-squares problems, lattice problems, maximum-likelihood detection, NP hard, nulling and cancelling, sphere decoding

†In multi-antenna wireless communication systems, data is transmitted across channels that can often be modeled as linear and time-invariant. The received signal in such systems is given by a linear combination of the transmitted data symbols, corrupted by an additive Gaussian noise,

$$\mathbf{x} = \mathbf{H}\mathbf{s} + \mathbf{v}, \tag{1.1}$$

† This work is supported in part by the NSF under grant no. CCR-0133818, by the ONR under grant no. N00014-02-1-0578, and by Caltech's Lee Center for Advanced Networking

where $\mathbf{H}$ is an $N \times M$ complex valued channel whose realization is known to the receiver (and is estimated, for instance, by means of sending a known training sequence), $\mathbf{s}$ is an M-dimensional transmitted symbol, and $\mathbf{v}$ is an N-dimensional noise with $\mathcal{C}(0, \sigma^2)$ Gaussian entries. Furthermore, we will assume that the entries in the transmitted symbol vector $\mathbf{s}$ in (1.1) are points in a QAM constellation.

For computational reasons, we shall replace the complex-valued model (1.1) by its real-valued equivalent in the usual way. To this end, we define the $m = 2M$ dimensional vector s, and the $n = 2N$ dimensional vectors x and v, composed of the real and imaginary parts of $\mathbf{s}$, $\mathbf{x}$, and $\mathbf{v}$, respectively, as

$$s = \left[\mathcal{R}(\mathbf{s})^T \quad \mathcal{I}(\mathbf{s})^T\right]^T, \quad x = \left[\mathcal{R}(\mathbf{x})^T \quad \mathcal{I}(\mathbf{x})^T\right]^T, \quad v = \left[\mathcal{R}(\mathbf{v})^T \quad \mathcal{I}(\mathbf{v})^T\right]^T,$$

and the $n \times m$ matrix H

$$H = \begin{bmatrix} \mathcal{R}(\mathbf{H}) & \mathcal{I}(\mathbf{H}) \\ -\mathcal{I}(\mathbf{H}) & \mathcal{R}(\mathbf{H}) \end{bmatrix}.$$

Then the real-valued equivalent of the model (1.1) is given by

$$x = Hs + v. \tag{1.2}$$

At a receiver, a detector forms an estimate of the transmitted symbol, $\hat{s}$. The optimal detector minimizes the average probability of error, i.e., it minimizes $P(\hat{s} \neq s)$. This is achieved by the maximum-likelihood (ML) design which, under the previous assumptions, performs the non-linear optimization

$$\min_{s \in \mathcal{D}_L^m} \|x - Hs\|^2, \tag{1.3}$$

where $\mathcal{D}_L^m$ denotes the m-dimensional square lattice spanned by an L-PAM constellation in each dimension. Furthermore, to obtain the soft decisions required by iterative decoding schemes in systems employing space-time or error-correcting codes, MIMO soft-decoding algorithms also often have to solve (1.3) or its modifications.

Problem (1.3), typically referred to as an *integer least-squares* problem, has a simple geometric interpretation. As the entries of s run over the points in the L-PAM constellation, s spans the "rectangular" m-dimensional lattice, $\mathcal{D}_L^m$. However, for any given *lattice-generating matrix* H, the n-dimensional vector Hs spans a "skewed" lattice. Thus, given the skewed lattice Hs and the vector $x \in \mathcal{R}^n$, the integer least-squares problem is to find the "closest" lattice point (in a Euclidean sense) to x, as illustrated in Figure 1.1.

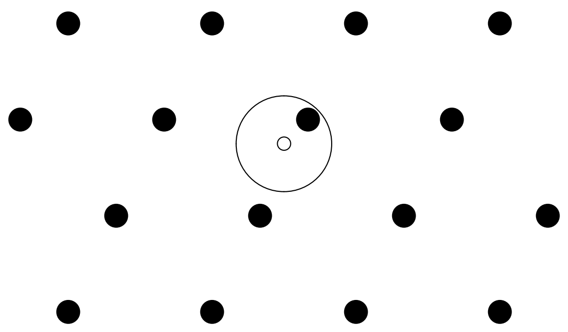

Fig. 1.1. Geometric interpretation of the integer least-squares problem

Problem (1.3) is, for a general H, known to be exponentially complex both in the worst-case sense [Grotschel et al., 1993] as well as in the average sense [Ajtai, 1998].

Optimal detection reduces to solving integer least-squares problems not only for the simple uncoded transmission problem modeled by (1.2), but also in the context of lattice codes [Banihashemi and Khandani, 1998; Agrell et al., 2002], CDMA systems [Brutel and Boutros, 1999; Viterbo and Boutros, 2000], multi-antenna systems employing space-time codes [Foschini, 1996; Damen et al., 2000; Hassibi and Hochwald, 2002], etc. Many of these applications are characterized by an affine mapping between the transmitted and the received signal, and thus again allow for the use of the model (1.2), where H now represents an *equivalent* channel.

In this chapter, we review the MIMO receiver algorithms for solving (1.3). In particular, the solution techniques that we discuss are the following:

- heuristic techniques, which provide approximate but readily implementable low-complexity solutions to the integer least-squares problem, and
- exact methods that, by exploiting the structure of the lattice, generally obtain the solution faster than a straightforward exhaustive search.

It is a pleasantly surprising fact that the exact techniques turn out to have complexity comparable to that of the heuristic techniques over a useful range of channel signal to noise ratios (SNR) (see Section 1.2).

1.1 Heuristic Techniques

Finding the exact solution of (1.3) is, in general, NP hard. Therefore, many wireless communication systems employ some approximations, heuristics or combinations thereof in often a manageable computational complexity. We briefly discuss some of these techniques.

Zero-forcing:

Solve the unconstrained least-squares problem to obtain $\hat{s} = H^{\dagger}x$, where $H^{\dagger}$ denotes the pseudo-inverse of H. Since the entries of $\hat{s}$ will not necessarily be integers, round them off to the closest integer (a process referred to as slicing) to obtain

$$\hat{s}_B = \left[H^{\dagger}x\right]_{\mathcal{Z}}. \tag{1.4}$$

The above $\hat{s}_B$ is often called a Babai estimate [Grotschel et al., 1993]. In the communications literature, this procedure is referred to as *zero-forcing equalization.*

The complexity of finding the Babai estimate is essentially determined by the complexity of finding the pseudo-inverse of the matrix H in (1.4). The simplest way of calculating the pseudo-inverse is by means of QR factorization, $H = QR$. It can also be calculated in a more stable way (which avoids inverting the upper triangular matrix R) by means of singular value decomposition (SVD) of H. In either case, assuming that H is square (i.e, $n = m$), the complexity of finding the Babai estimate is of cubic order, $O(m^3)$.

Nulling and cancelling:

In this method, the Babai estimate is used for only one of the entries of s, say the first. Then this entry, s_1, is assumed to be known and its effect is cancelled out to obtain a reduced-order integer least-squares problem with $m-1$ unknowns. The process is repeated to find s_2, etc. In communications parlance this is known as *decision-feedback equalization.*

We shall find it convenient to denote the partition of the channel matrix H into rows and columns as

$$H = [\underline{h}_1 \;\; \underline{h}_2 \;\; \ldots \;\; \underline{h}_m] = \begin{bmatrix} H_1 \\ \vdots \\ H_n \end{bmatrix}.$$

The nulling and cancelling algorithm can be stated by the following pseudocode:

$$
\begin{aligned}
&y_1 := x \\
&\text{for } k = 0 \text{ to } m-1 \\
&\quad \text{find weighting vector } w_{m-k} \\
&\quad \hat{s}_{m-k} := \text{slice}(w_{m-k} y_{k+1}) \\
&\quad y_{k+2} = y_{k+1} - \underline{h}_{m-k}\hat{s}_{m-k} \\
&\text{end}
\end{aligned}
$$

In the algorithm, for each value of the index k, the entries of the auxiliary vector y_{k+1} are weighted by the components of the weight vector w_{m-k} and linearly combined to account for the effect of the interference. Depending on the criterion chosen for the design of w_{m-k} (i.e., for performing the nulling operation), we can distinguish between the following cases:

(i) *Zero-forcing (ZF) nulling*

In this case, interference from the yet undetected symbols is nulled. Denoting

$$H_{m-k} = \left[\underline{h}_1 \;\; \underline{h}_2 \;\; \ldots \;\; \underline{h}_{m-k}\right],$$

this condition can be stated as

$$H^*_{m-k} w_{m-k} = e_{m-k},$$

where e_{m-k} is a $(m-k) \times 1$ column vector that consists of all zeros except for the $(m-k)$-th entry whose value is 1. The weighting vector is then given by the least-norm solution of the form

$$w_{m-k} = H^\dagger_{m-k} e_{m-k},$$

where $(\cdot)^\dagger$ denotes the pseudo-inverse, i.e., $H^\dagger_{m-k} = H_{m-k}(H^*_{m-k}H_{m-k})^{-1}$.

(ii) *Minimum mean-square error (MMSE) nulling*

The objective in MMSE nulling is to minimize the expected mean-square error between the receiver's estimate and the transmitted symbol. This can be expressed as

$$E\left[s_{m-k} y^*_{k+1}\right] = w^*_{m-k} E\left[y_{k+1} y^*_{k+1}\right].$$

Furthermore, we shall assume that the previous decisions made by the detector were correct, i.e., $\hat{s}_{m-k} = s_{m-k}$. Defining

$$s_{1:m-k} = [s_1 \;\; s_2 \;\; \ldots \;\; s_{m-k}],$$

we can write

$$y_{k+1} = H_{m-k} s_{1:m-k} + v,$$

where v is the noise vector in (1.2). Furthermore, assuming that the transmitted symbol sequence is spatially white and has variance $\mathcal{E}_s$, we readily find that the MMSE nulling vector is given by

$$w_{m-k} = \left(H_{m-k}^* H_{m-k} + \frac{1}{\rho} I \right)^{-1} \underline{h}_{m-k},$$

where $\rho = \frac{\mathcal{E}_s}{\sigma^2}$ denotes the signal-to-noise ratio (SNR).

The computational complexity is again determined by the complexity of solving the underlying unconstrained least-squares problem, i.e., calculating the pseudo-inverse at each step of the algorithm. When $m = n$, we need to evaluate the pseudo-inverse of a series of matrices with dimensions $m \times (m-k)$, where $k = m, m-1, \ldots, 1$. The computational complexity of performing this series of operations is clearly of the fourth order, i.e., $O(m^4)$, an order of magnitude higher than the complexity of finding the Babai estimate.

Nulling and cancelling with optimal ordering.

The nulling and cancelling algorithm can suffer from *error-propagation*: if s_1 is estimated incorrectly it can have an adverse effect on estimation of the remaining unknowns s_2, s_3, etc. To minimize the effects of error propagation, it is advantageous to perform nulling and cancelling from the "strongest" to the "weakest" signal. This is the method proposed for V-BLAST [Foschini, 1996].

Consider, for instance, the MMSE nulling and cancelling algorithms. To perform optimal ordering, we consider the covariance matrix of the estimation error $s - \hat{s}$,

$$P = E(s - \hat{s})(s - \hat{s})^* = \left(H^* H + \frac{1}{\rho} I \right)^{-1}.$$

Consider the entries of the estimated symbol $\hat{s}$, i.e., $\{\hat{s}_i, i = 1, 2, \ldots, m\}$. The "strongest" signal, corresponding to the "best" estimate, is the one with the smallest variance, i.e., s_i for which P_{ii} is the smallest. If we reorder the entries of s so that the strongest signal is s_m, then the estimate $\hat{s}_m$ is going to be better than it would be for any other ordering of the entries in s. This ordering we perform at each step of the nulling and cancelling algorithm.

The computational complexity of the algorithm is the same as the complexity of the standard nulling and cancelling algorithm, namely $O(m^4)$, augmented by the complexity of the ordering operation, which, for the set of k elements, is $O(k^3)$.

Square-root algorithm for nulling and cancelling:

The increased complexity of the nulling and cancelling algorithm as compared to the zero-forcing algorithm is that the former requires repeated evaluation of the pseudo-inverse for each deflated channel matrix. It is of interest to seek cost effective implementations of the algorithm so that a pseudo-inverse at a particular stage can be efficiently deduced from the pseudo-inverse computed at the previous stage. To this end, using the array algorithm ideas of linear estimation theory (see, e.g., [Kailath et al., 2000]), a so-called square-root algorithm was proposed in [Hassibi, 1999].

Such algorithms are characterized by numerical stability and robustness achieved by increasing the dynamic range of the quantities involved, their condition numbers, etc. In particular, this is obtained by insisting to

- Avoid squaring objects, such as computation of H^*H.
- Avoid inverting objects.
- Make as much use as possible of unitary transformations.

For the following discussion, it will be convenient to write the basic linear least-mean-squares estimate of s, given the observation $x = Hs + v$, as

$$\hat{s} = (H^*H + \frac{1}{\rho}I)^{-1}H^*x = \begin{bmatrix} H \\ \frac{1}{\sqrt{\rho}}I_m \end{bmatrix}^{\dagger} \begin{bmatrix} x \\ 0 \end{bmatrix} = H_1^{\dagger}x, \tag{1.5}$$

where $H_1^{\dagger}$ denotes the first n columns of the pseudo-inverse of the augmented channel matrix in (1.5).

To start the use of unitary transformations, and to avoid squaring H, consider the QR decomposition of the augmented channel matrix

$$\begin{bmatrix} H \\ \frac{1}{\sqrt{\rho}}I_m \end{bmatrix} = QR = \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} R,$$

where Q is an $(n+m) \times m$ matrix with orthonormal columns, and R is $m \times m$ non-singular and upper-triangular. Note then that

$$P = (H^*H + \frac{1}{\rho}I_m)^{-1} = (R^*R)^{-1} = R^{-1}R^{-*}.$$

Thus we can identify R^{-1} as a square-root of P, say,

$$R^{-1} = P^{1/2}, \quad P^{1/2}P^{*/2} = P.$$

The pseudo-inverse of the augmented channel matrix now becomes

$$\begin{bmatrix} H \\ \frac{1}{\sqrt{\rho}} I_m \end{bmatrix}^{\dagger} = R^{-1} Q^* = P^{1/2} Q^*,$$

and thus

$$H_1^{\dagger} = P^{1/2} Q_1^*.$$

Therefore, given $P^{1/2}$ and Q_1, we can compute both the pseudo-inverse and the error covariance matrix, required for the nulling operation and the optimal ordering. The problem becomes one of finding the best way to compute $P^{1/2}$ and Q_1.

Recall that the optimal ordering is performed according to the values of the diagonal entries of P. This information can also be deduced from $P^{1/2}$. Since the diagonal entries of P are simply the squared length of the rows of $P^{1/2}$, the minimum diagonal entry of P corresponds to the minimum length row of $P^{1/2}$.

Now assume that the entries of the transmitted signal s have been reordered so that the m-th diagonal entry is the smallest. Consider a unitary transformation Σ that rotates (or reflects) the m-th row of $P^{1/2}$ to lie along the direction of the m-th unit vector, i.e.,

$$P^{1/2}\Sigma = \begin{bmatrix} P^{(m-1)/2} & P_m^{(m-1)/2} \\ 0 & p_m^{1/2} \end{bmatrix}, \tag{1.6}$$

where $p_m^{1/2}$ is a scalar. It was shown in [Hassibi, 1999] that the block upper triangular square-root factor of P in (1.6), $P^{(m-1)/2}$, is a square-root factor of P^{m-1}. Therefore, to find the square-root factor of $P^{(m-1)}$, one needs to make P block upper triangular. The next signal to be detected is selected by finding the minimum length row of $P^{(m-1)/2}$. The rows of $P^{(m-1)/2}$ are then reordered so that this minimum length row corresponds to the last $(m-1)$-th row, and the upper block triangularization of $P^{(m-1)/2}$ gives the next square-root factor, $P^{(m-2)/2}$, and so on.

The process described above results in upper triangularization of the square root matrix $P^{1/2}$. Let $\underline{q}_{1,i}$, $i = 1, \ldots, m$, denote the resulting columns of Q_1, i.e.,

$$Q_1 = \left[\underline{q}_{1,1} \cdots \underline{q}_{1,m} \right].$$

It was shown in [Hassibi, 1999] that the nulling vectors for the signals s_1 to s_m are given by

$$H_{1,i}^{\dagger} = p_i^{1/2} \underline{q}_{1,i}^*,$$

where $p_i^{1/2}$ is the i-th diagonal entry of $P^{1/2}$. Therefore, the nulling vectors are simply found by scaling the columns of Q_1 by the diagonals of $P^{1/2}$. Moreover, there is no need for recomputing $P^{1/2}$ and Q_1 for the deflated matrices $H^{(m-k)}$, $k = 1, \ldots, m-1$. The information needed for the optimal ordering and finding nulling vectors is already implicitly contained in $P^{1/2}$ and Q_1.

What remains to be specified is the computation of $P^{1/2}$ and Q_1. Note that we can write

$$P = (\sum_{j=1}^{n} H_j^* H_j + \frac{1}{\rho} I)^{-1}.$$

Denoting

$$P_{|i} \triangleq \left(\sum_{j=1}^{i} H_j^* H_j + \frac{1}{\rho} I \right)^{-1}, \quad P_{|n} = P,$$

and using a matrix inversion lemma, we obtain the so-called Ricccati recursion of the RLS (recursive-least-squares) algorithm ([Kailath et al., 2000], Section 2.6),

$$P_{|i} = P_{|i-1} - \frac{P_{|i-1} H_i^* H_i P_{|i-1}}{r_{e,i}}, \quad r_{e,i} = 1 + H_i P_{|i-1} H_i^*, \quad P_{|0} = \rho I. \tag{1.7}$$

On the other hand, to find a recursion for $H_1^\dagger$, note that the least-mean-square estimate of the signal, $\hat{s} = P^{1/2} Q_1^* x = H_1^\dagger x$, satisfies the recursion ([Kailath et al., 2000], Lemma 2.6.1)

$$\hat{s}_{|i} = \hat{s}_{|i-1} + \bar{K}_{p,i} r_{e,i}^{-1/2} (x_i - H_i \hat{s}_{|i-1}), \quad \bar{K}_{p,i} = P_{|i-1} H_i^* r_{e,i}^{-*/2}, \quad \hat{s}_{|0} = 0.$$

Then the recursion for the pseudoinverse $H_1^\dagger = P^{1/2} Q_1$ can be written as

$$H_{1|i}^\dagger = H_{1|i-1}^\dagger + \bar{K}_{p,i} r_{e,i}^{1/2} (e_i^* - H_i H_{1|i-1}^\dagger), \quad \bar{K}_{p,i} = P_{|i-1} H_i^* r_{e,i}^{-*/2}, \quad H_{1|0}^\dagger = 0_{m \times n}. \tag{1.8}$$

Note that $H_1^\dagger = H_{1|n}^\dagger$.

One can further improve (1.7), (1.8), by ensuring direct propagation of $P^{1/2}$. Incorporating these improvements, the algorithm of [Hassibi, 1999] can be summarized as follows:

(i) Compute $P^{1/2}$ and Q_1.

Propagate a square-root algorithm of the following form:

$$\begin{bmatrix} 1 & H_i P_{|i-1}^{1/2} \\ 0 & P_{|i-1}^{1/2} \\ -e_i & B_{i-1} \end{bmatrix} \Theta_i = \begin{bmatrix} r_{e,i}^{1/2} & 0 \\ \bar{K}_{p,i} & P_{|i}^{1/2} \\ A_i & B_i \end{bmatrix}, \quad P_{|0}^{1/2} = \sqrt{\rho} I, B_0 = 0_{n \times m},$$

where e_i is the i-th unit vector of dimension n, and Θ_i is any unitary transformation that block lower triangularizes the pre-array. After n steps, we obtain

$$P^{1/2} = P_{|n}^{1/2} \text{ and } Q_1 = B_n.$$

(ii) Find the minimum length row of $P^{1/2}$ and permute it to be the last (m-th) row. Permute s accordingly.

(iii) Find a unitary Σ that makes $P^{1/2}\Sigma$ block upper triangular,

$$P^{1/2}\Sigma = \begin{bmatrix} P^{(m-1)/2} & P_m^{(m-1)/2} \\ 0 & p_m^{1/2} \end{bmatrix}.$$

(iv) Update Q_1 to $Q_1\Sigma$.

(v) The nulling vector for the m-th signal is given by $p_m^{1/2} q_{1,m}^*$, where $q_{1,m}$ denotes the m-th column of Q_1.

(vi) Go back to step 3, but now with $P^{(m-1)/2}$ and $Q_1^{(m-1)}$, (the first $m-1$ columns of Q_1).

The square-root algorithm achieves all the desired computational objectives. In particular, it avoids computing the pseudo-inverse for each deflated channel matrix, avoids squaring or inverting any quantities, and makes extensive use of unitary transformations. Using the algorithm, the computational complexity of the nulling and cancelling receiver can be reduced from $O(m^4)$ to $O(m^3)$, which is the complexity of the simple zero-forcing algorithm discussed earlier.

Solving relaxed convex optimization problems:

Another heuristic approach to maximum-likelihood detection is via convex optimization techniques. The integer least-squares problem is essentially transformed into an optimization problem with both objective and constraint being convex functions. To illustrate the technique, we consider the detection problem where the entries in the symbol s are chosen from $4 - QAM$ constellations, i.e., for each entry in the symbol vector s it holds that $s_i^2 = 1$.

Since

$$\begin{aligned}\|x - Hs\|^2 &= s^T H^T H s - 2x^T H^T s + x^T x \\ &= \mathrm{Tr} H^T H S - 2x^T H^T s + x^T x,\end{aligned}$$

where $S = ss^T$, the integer least-squares problem can be expressed as

$$\min \mathrm{Tr} H^T H S - 2x^T H^T s + x^T x$$

$$\text{subject to } S_{ii} = 1, S \succeq ss^T,\ \mathrm{rank}(S) = 1$$

Using

$$S \succeq ss^T \Leftrightarrow \begin{bmatrix} S & s \\ s^T & 1 \end{bmatrix} \succeq 0$$

and relaxing the rank one constraint, one can obtain a semi-definite program (with variables S, s) of the form

$$\min \mathrm{Tr} H^T H S - 2x^T H^T s + x^T x$$

$$\text{subject to } S_{ii} = 1, \begin{bmatrix} S & s \\ s^T & 1 \end{bmatrix} \succeq 0.$$

Solving this SDP for s, an approximate solution to the detection problem can be found as $\hat{s} = \mathrm{sgn}(s)$ (recall that the algorithm is only for $s_i^2 = 1$). The complexity of solving the SDP is roughly cubic, $O(m^3)$.

1.2 Exact Methods: Sphere Decoding

With an abundance of heuristic methods presented in the previous section, it is natural to ask how close they come to the optimal solution? In Figure 1.2, the bit-error rate (BER) performance of an exact solution is compared with the ordered nulling and cancelling (N/C) for a multi-antenna system with $M = 8$ transmit and $N = 12$ receive antennas employing 16-QAM modulation scheme. Clearly, the ML receiver significantly outperforms N/C; thus there is merit in studying exact solutions. The most obvious one is to search over the entire lattice which invariably requires an exponential search. There do, however, exist exact methods that are more sophisticated than exhaustive search and can be employed for an arbitrary H. Such are Kannan's algorithm [Kannan, 1983] (which searches only over restricted parallelograms), the KZ algorithm [Lagarias et al., 1990] (based on the Korkin-Zolotarev reduced basis [Korkin and Zolotarev, 1873]) and the sphere decoding algorithm of Fincke and Pohst [Pohst, 1981; Fincke and Pohst, 1985]. We will focus on the latter, i.e., on solving (1.3) with the

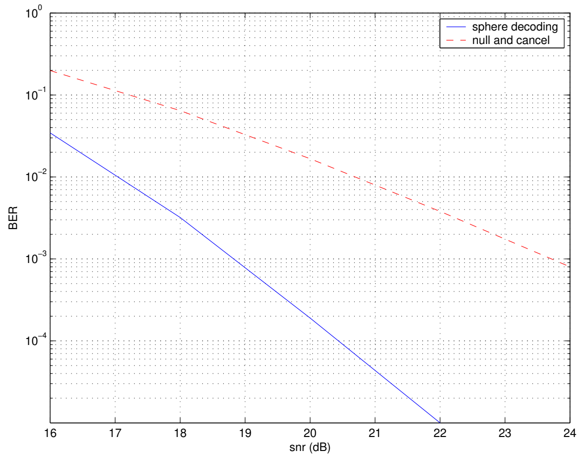

Fig. 1.2. Bit error performance of a sphere decoding vs. nulling and cancelling with optimal ordering, $M = 8$, $N = 12$, 16-QAM.

sphere decoding algorithm. [For the system in Figure 1.2, finding the exact solution by means of exhaustive search requires testing 4.3×10^9 points and is thus practically infeasible. The exact performance curve in Figure 1.2 is obtained with the sphere decoding algorithm, which on the other hand requires computational effort implementable in practice.]

The basic premise in sphere decoding is rather simple: attempt to search over only lattice points $s \in \mathcal{D}_L^m$ that lie in a certain sphere of radius d around the given vector x, thereby reducing the search space and hence the required computational effort (see Figure 1.3). Clearly, the closest lattice point inside the sphere will also be the closest lattice point for the whole lattice. However, closer scrutiny of this basic idea leads to two key questions.

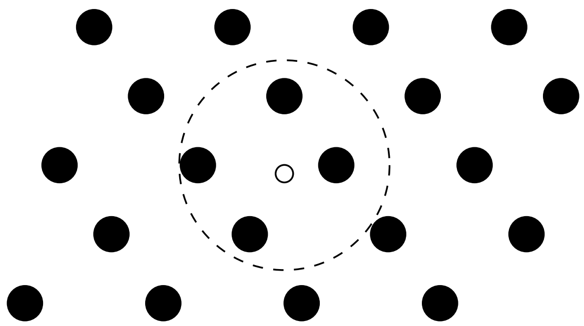

Fig. 1.3. Idea behind the sphere decoder

(i) *How to choose d?* Clearly, if d is too large, we may obtain too many points and the search may remain exponential in size, whereas if d is too small, we may obtain no points inside the sphere.

A natural candidate for d is the *covering radius* of the lattice, defined to be the smallest radius of spheres centered at the lattice points that cover the entire space. This is clearly the smallest radius that guarantees the existence of a point inside the sphere for any vector x. The problem with this choice of d is that determining the covering radius for a given lattice is itself NP hard [Conway and Sloane, 1993].

Another choice is to use d as the distance between the Babai estimate and the vector x, i.e., $d = \|x - H\hat{s}_B\|$, since this radius guarantees the existence of at least one lattice point (here the Babai estimate) inside the sphere. However, it may happen that this choice of radius will yield too many lattice points lying inside the sphere.

(ii) *How can we tell which lattice points are inside the sphere?* If this requires testing the distance of each lattice point from x (to determine whether it is less than d), then there is no point in sphere decoding as we shall still need an exhaustive search.

Sphere decoding does not really address the first question. [We shall address it later by exploiting statistical assumptions in our model.] However, it does propose an efficient way to answer the second one. The basic observation is the following. Although it is difficult to determine the lattice points inside a general m-dimensional sphere, it is trivial to do so in the (one-dimensional) case of $m = 1$. The reason is that a one-dimensional sphere reduces to the endpoints of an interval and so the desired lattice points will be the integer values that lie in this interval. We can use this observation to go from dimension k to dimension $k + 1$. Suppose we have determined all k-dimensional lattice points that lie in a sphere of radius d. Then for any such k-dimensional point, the set of admissible values of the $k + 1$-th dimensional coordinate that lie in the higher dimensional sphere of the *same* radius d forms an interval.

The above means that we can determine all lattice points in a sphere of dimension m and radius d by successively determining all lattice points in spheres of lower dimensions $1, 2, \ldots, m$ and the same radius d. Such an algorithm for determining the lattice points in an m-dimensional sphere essentially constructs a tree where the branches in the k-th level of the tree correspond to the lattice points inside the sphere of radius d and dimension k—see Figure 1.4. Moreover, the complexity of such an algorithm will depend on the *size* of the tree, i.e., on the number of lattice points visited by the algorithm in different dimensions.

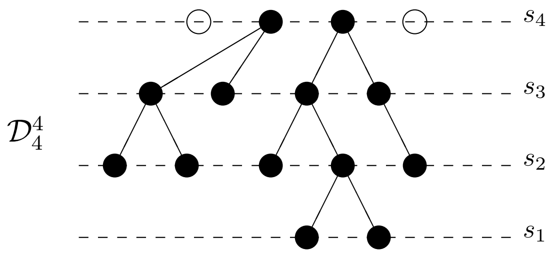

Fig. 1.4. Sample tree generated to determine lattice points in a 4-dimensional sphere.

With this brief discussion we can now be more specific about the problem at hand. To this end, we shall assume that $n \geq m$, i.e., that there are at least as many equations as unknowns in $x \approx Hs$. Note that the lattice point Hs lies inside a sphere of radius d centered at x if, and only if,

$$d^2 \;\geq\; \|x - Hs\|^2 . \tag{1.9}$$

In order to break the problem into the subproblems described above, it is useful to introduce the QR factorization of the matrix H

$$H = Q \begin{bmatrix} R \\ 0_{(n-m)\times m} \end{bmatrix}, \tag{1.10}$$

where R is an $m \times m$ upper triangular matrix and $Q = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix}$ is an $n \times n$ orthogonal matrix. The condition (1.9) can then be written as

$$d^2 \;\geq\; \left\| x - \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} R \\ 0 \end{bmatrix} s \right\|^2 = \|Q_1^* x - Rs\|^2 + \|Q_2^* x\|^2 ,$$

where $(\cdot)^*$ here denotes Hermitian matrix transposition. Or in other words,

$$d^2 - \|Q_2^* x\|^2 \;\geq\; \|Q_1^* x - Rs\|^2 . \tag{1.11}$$

Defining $y = Q_1^* x$ and $d'^2 = d^2 - \|Q_2^* x\|^2$ allows us to rewrite this as

$$d'^2 \geq \sum_{i=1}^{m} \left(y_i - \sum_{j=i}^{m} r_{i,j} s_j \right)^2 , \tag{1.12}$$

where $r_{i,j}$ denotes an (i,j) entry of R. Here is where the upper triangular property of R comes in handy. The right-hand side (RHS) of the above inequality can be expanded as

$$d'^2 \geq (y_m - r_{m,m}s_m)^2 + (y_{m-1} - r_{m-1,m}s_m - r_{m-1,m-1}s_{m-1})^2 + \ldots \quad (1.13)$$

where the first term depends only on s_m, the second term on $\{s_m, s_{m-1}\}$ and so on. Therefore a necessary condition for Hs to lie inside the sphere is that $d'^2 \geq (y_m - r_{m,m}s_m)^2$. This condition is equivalent to s_m belonging to the interval

$$\left\lceil \frac{-d' + y_m}{r_{m,m}} \right\rceil \leq s_m \leq \left\lfloor \frac{d' + y_m}{r_{m,m}} \right\rfloor, \quad (1.14)$$

where $\lceil \cdot \rceil$ denotes rounding to the nearest larger element in the L-PAM constellation which spans the lattice. Similarly, $\lfloor \cdot \rfloor$ denotes rounding to the nearest smaller element in the L-PAM constellation which spans the lattice.

Of course, (1.14) is by no means sufficient. For every s_m satisfying (1.14), defining $d'^2_{m-1} = d'^2 - (y_m - r_{m,m}s_m)^2$ and $y_{m-1|m} = y_{m-1} - r_{m-1,m}s_m$, a stronger necessary condition can be found by looking at the first two terms in (1.13), which leads to s_{m-1} belonging to the interval

$$\left\lceil \frac{-d'_{m-1} + y_{m-1|m}}{r_{m-1,m-1}} \right\rceil \leq s_{m-1} \leq \left\lfloor \frac{d'_{m-1} + y_{m-1|m}}{r_{m-1,m-1}} \right\rfloor. \quad (1.15)$$

One can continue in a similar fashion for s_{m-2}, and so on until s_1, thereby obtaining all lattice points belonging to (1.9).

We can now formalize the algorithm.

Input: $Q = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix}$, R, x, $y = Q_1^* x$, d.

1. Set $k = m$, $d'^2_m = d^2 - \|Q_2^* x\|^2$, $y_{m|m+1} = y_m$
2. (Bounds for s_k) Set $UB(s_k) = \lfloor \frac{d'_k + y_{k|k+1}}{r_{k,k}} \rfloor$, $s_k = \lceil \frac{-d'_k + y_{k|k+1}}{r_{k,k}} \rceil - 1$
3. (Increase s_k) $s_k = s_k + 1$. If $s_k \leq UB(s_k)$ go to 5, else go to 4.
4. (Increase k) $k = k + 1$; if $k = m + 1$ terminate algorithm, else go to 3.
5. (Decrease k) If $k = 1$ go to 6. Else $k = k - 1$, $y_{k|k+1} = y_k - \sum_{j=k+1}^{m} r_{k,j}s_j$, $d'^2_k = d'^2_{k+1} - (y_{k+1|k+2} - r_{k+1,k+1}s_{k+1})^2$, and go to 2.
6. Solution found. Save s and its distance from x, $d'^2_m - d'^2_1 + (y_1 - r_{1,1}s_1)^2$, and go to 3.

Note that the subscript $k|k+1$ in $y_{k|k+1}$ above is used to denote the received signal y_k adjusted with the already estimated symbol components $s_{k+1}, \ldots, s_m$.

We also need a method to determine the desired radius d. Here is where our statistical model of the communication system helps. Note that $\frac{1}{\sigma^2} \cdot$

$\|v\|^2 = \frac{1}{\sigma^2} \cdot \|x - Hs\|^2$ is a χ^2 random variable with n degrees of freedom. Thus we may choose the radius to be a scaled variance of the noise,

$$d^2 = \alpha n \sigma^2,$$

in such a way that with a high probability we find a lattice point inside the sphere,

$$\int_0^{\alpha n/2} \frac{\lambda^{n/2-1}}{\Gamma(n/2)} e^{-\lambda} d\lambda = 1 - \epsilon,$$

where the integrand is the probability density function of the χ^2 random variable with n degrees of freedom, and where $1-\epsilon$ is set to a value close to 1, say, $1-\epsilon = 0.99$. [If the point is not found, we can increase the probability $1-\epsilon$, adjust the radius, and search again.]

With the above choice of the radius, and because of the random nature of H and v, the computational complexity of the sphere decoding algorithm is clearly a random variable. Moreover, and rather strikingly, we can compute the mean and the variance of the complexity. We omit the details, but note that in [Hassibi and Vikalo, 2005], the mean value is calculated

(i) for a 2-PAM constellation to be

$$C(m, \rho, d^2) = \sum_{k=1}^{m} f_p(k) \sum_{l=0}^{k} \binom{k}{l} \gamma \left(\frac{\alpha n}{2(1 + \frac{12\rho l}{m(L^2-1)})}, \frac{n-m+k}{2} \right) \tag{1.16}$$

(ii) for a 4-PAM constellation to be

$$C(m, \rho, d^2) = \sum_{k=1}^{m} f_p(k) \sum_{q} \frac{1}{2^k} \sum_{l=0}^{k} \binom{k}{l} g_{kl}(q) \gamma \left(\frac{\alpha n}{2(1 + \frac{12\rho q}{m(L^2-1)})}, \frac{n-m+k}{2} \right), \tag{1.17}$$

where $g_{kl}(q)$ is the coefficient of x^q in the polynomial

$$(1 + x + x^4 + x^9)^l (1 + 2x + x^4)^{k-l}.$$

The number of elementary operations per visited point in (1.16)-(1.17) is $f_p(k) = 2k+9+2L$, and $\gamma(\cdot,\cdot)$ denotes an incomplete gamma function.

Similar expressions can be obtained for 8-PAM, 16-PAM, etc., constellations.

Let $C(m, \rho)$ denote the expected complexity of actually finding the solution, i.e., the expected complexity of the search where we keep increasing radii until finding a lattice point. Figure 1.5 shows the expected complexity exponent defined as $e_c = \log_m(C(m,\rho))$. For a wide range of SNR, $e_c \le 4$, and thus in such SNR regions the expected complexity of the sphere decoding is comparable with the complexity of the heuristic techniques.

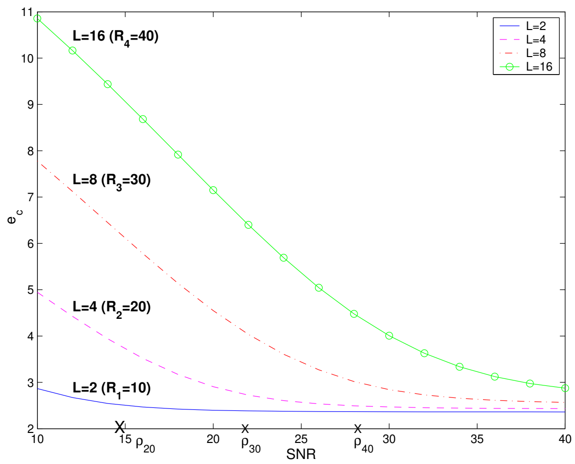

Fig. 1.5. The complexity exponent as a function of ρ for $m = n = 10$ and $L = 2, 4, 8, 16$

Figure 1.5 shows the complexity as a function of SNR for $m = 10$ and L^2-QAM constellations with $L = 2, 4, 8, 16$. A particular modulation scheme can be used only in the range of SNRs that supports transmission at the rate corresponding to that modulation scheme, i.e., the rate has to be smaller than the ergodic capacity of the MIMO channel,

$$C_{\text{erg}} = E\left\{\log\det\left(I_M + \mathbf{H}^*\mathbf{H}\right)\right\}.$$

On the other hand, the complexity of the sphere decoding algorithm for such SNRs is practically feasible (as noted above, it is often $e_c \le 4$). For instance, although the complexity for $L = 16$ appears to be high over a wide range of SNR, it is only for $\rho > \rho_{40} = 27.9dB$ that this modulation scheme can be employed (ρ_{40} is the SNR for which the capacity $C_{\text{erg}} = 40 = R_4(L = 16)$). The complexity exponent at ρ_{40} and $L = 16$ is $e_c \approx 4.4$. The other SNRs marked on Figure 1.5, $\rho_{30} = 21.6dB$, and $\rho_{20} = 14.9dB$, have similar meanings (only for $L = 8$ and $L = 4$, respectively).

On another note, the expected complexity above accounts for finding all the lattice points in the sphere. The point among those found that is closest to x is the solution to (1.3). There are some more efficient variations on the basic sphere decoding algorithm that potentially avoid having to search over all the lattice points inside the sphere. We briefly mention two of them here. In both cases, of course, the expected complexity will be no greater than that of the basic sphere decoding algorithm; however, exact calculation of the expected complexity appears to be difficult.

- *Sphere decoding with radius update.*

Whenever the algorithm finds a point s_{in} inside the sphere (note that Hs_{in} is not necessarily the closest point to x), we set the new radius of the sphere $d^2 = \|x - Hs_{in}\|^2$ and restart the algorithm. Such radius update may be particularly useful at lower SNRs, where the number of points in the initial sphere is relatively large.

- *Schnorr-Euchner version of sphere decoding.*

This strategy was proposed in [Schnorr and Euchner, 1994]. The likelihood that the point will be found early is maximized if the search at each dimension k is performed from the middle of the allowed interval for s_k, and if the radius update strategy (as described above) is used. More details about the Schnorr-Euchner version of the sphere decoding, and some improvements thereof, can be found in [Agrell et al., 2002; Damen et al., 2003].

1.3 Soft MIMO Receive Algorithms

Multi-antenna wireless communication systems that protect transmitted data by either imposing error-correcting or space-time codes require probabilistic (soft) information at the MIMO receiver. This soft information is typically used to iterate between the receiver and the inner decoder (which recovers information from the error-correcting or the space-time encoder).

In [Stefanov and Duman, 2001], turbo-coded modulation for multi-antenna systems was studied, and heuristics based on N/C employed to obtain soft channel information. It was also noted there that if the soft information is obtained by means of an exhaustive search, the computational complexity grows exponentially in the number of transmit antennas and in the size of the constellation. Hence, for high-rate systems with large number of antennas, the exhaustive search proves to be practically infeasible.

In [Vikalo et al., 2004; Hochwald and ten Brink], two variations of the sphere decoding algorithm were proposed for obtaining the soft information. Both variations reduce the complexity of estimating the soft information by employing sphere decoding ideas to constrain the number of lattice points used for computing the required likelihood ratios. In [Hochwald and ten

Brink], sphere decoding was employed to obtain a list of bit sequences that are "good" in a likelihood sense. This list is then used to generate soft information, which is subsequently updated by iterative channel decoder decisions. In [Vikalo et al., 2004], a MIMO detector based on a modification of the original Fincke-Pohst algorithm was proposed to efficiently obtain soft information for the transmitted bit sequence. This modified Fincke-Pohst algorithm essentially performs a maximum a posteriori (MAP) search, i.e., it solves

$$\min_{s \in \mathcal{D}_L^m} \left[\|x - Hs\|^2 - \sum_{k=1}^{m} \log p(s_k) \right],$$

where $p(s_k)$ are *a priori* information for each symbol in the transmitted sequence. The MAP search is used to obtain a set of lattice points that contribute significantly to the likelihood ratios (for more details, see [Vikalo et al., 2004]). These likelihood ratios (i.e., the required soft information) are then passed onto the channel decoder. The channel decoder's output is then fed back to the Fincke-Pohst MAP (FP-MAP) for the next iteration.

As discussed above, to obtain computationally efficient receiver schemes, the MIMO communication systems utilizing soft information may require modifications of the basic sphere decoding algorithm. Other MIMO systems may require such modifications as well. For FIR channels, the sphere decoding algorithm does not at all exploit the Markovian property of the channel, which is precisely what the Viterbi algorithm does. Practical algorithms that combine both structures (the lattice and the Markovian property) are highly desirable. On the other hand, when error-correcting codes are coupled with analog channels (through some modulation scheme) problems of joint detection and decoding arise. Some preliminary work addressing both these issues can be found in [Vikalo, 2003].

References

E. Agrell, T. Eriksson, A. Vardy, and K. Zeger. Closest point search in lattices. *IEEE Transactions on Information Theory*, 48(8):2001–2214, 2002.

M. Ajtai. The shortest vector problem in L_2 is NP-hard for randomized reductions. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, pages 10–19, 1998.

A.H. Banihashemi and A.K. Khandani. On the complexity of decoding lattices using the Korkin-Zolotarev reduced basis. *IEEE Transactions on Information Theory*, 44(2):162–171, 1998.

C. Brutel and J. Boutros. Euclidean space lattice decoding for joint detection in CDMA systems. In *Proceedings of the 1999 IEEE Information Theory and Communications Workshop*, page 129, 1999.

J.H. Conway and N.J. Sloane. *Sphere Packings, Lattices and Graphs.* Springer-Verlag, 1993.

M. O. Damen, A. Chkeif, and J. C. Belfiore. Lattice codes decoder for space-time codes. *IEEE Communications Letters*, 4:161–163, May 2000.

M. O. Damen, H. El Gamal, and G. Caire. On maximum-likelihood detection and the search for the closest lattice point. *IEEE Transactions on Information Theory*, 49(10):2389–2402, October 2003.

U. Fincke and M. Pohst. Improved methods for calculating vectors of short length in a lattice, including a complexity analysis. *Mathematics of Computation*, 44 (170):463–471, April 1985.

G. J. Foschini. Layered space-time architecture for wireless communication in a fading environment when using multi-element antennas. *Bell Labs Technical Journal*, 1(2):41–59, 1996.

M. Grotschel, L. Lovász, and A. Schriver. *Geometric Algorithms and Combinatorial Optimization.* Springer Verlag, 2nd edition, 1993.

B. Hassibi. An efficient square-root algorithm for blast. *Submitted to IEEE Transactions on Signal Processing. Available for download from http://mars.bell-labs.com*, 1999.

B. Hassibi and B. Hochwald. High-rate codes that are linear in space and time. *IEEE Transactions on Information Theory*, 48(7):1804–1824, July 2002.

B. Hassibi and H. Vikalo. On sphere decoding algorithm. I. Expected complexity. *to appear in IEEE Trans. on Signal Processing*, 2005.

B. M. Hochwald and S. ten Brink. Achieving near-capacity on a multiple-antenna channel. *IEEE Transactions on Communications.*

T. Kailath, A. H. Sayed, and B. Hassibi. *Linear Estimation.* Prentice-Hall, Englewood Cliffs, NJ, 2000.

R. Kannan. Improved algorithms on integer programming and related lattice problems. *Proceedings of the 15th Annual ACM Symposium on Theory of Computing*, pages 193–206, 1983.

A. Korkin and G. Zolotarev. Sur les formes quadratiques. *Math. Ann.*, 6:366–389, 1873.

J.C. Lagarias, H.W. Lenstra, and C.P. Schnorr. Korkin-Zolotarev bases and successive minima of a lattice and its reciprocal. *Combinatorica*, 10:333–348, 1990.

M. Pohst. On the computation of lattice vectors of minimal length, successive minima and reduced basis with applications. *ACM SIGSAM Bull.*, 15:37–44, 1981.

C. P. Schnorr and M. Euchner. Lattice basis reduction: improved practical algorithms and solving subset sum problems. *Mathematical Programming*, 66: 181–191, 1994.

A. Stefanov and T. M. Duman. Turbo-coded modulation for systems with transmit and receive antenna diversity over block fading channels: system model, decoding approaches, and practical considerations. *IEEE Journal on Selected Areas in Communications*, 19(5), May 2001.

H. Vikalo. *Sphere Decoding Algorithms for Digital Communications.* PhD thesis, Stanford University, 2003.

H. Vikalo, B. Hassibi, and T. Kailath. Iterative decoding for MIMO channels via modified sphere decoder. *IEEE Transactions on Wireless Communications*, 3 (6), November 2004.

E. Viterbo and J. Boutros. A universal lattice code decoder for fading channels. *IEEE Transactions on Information Theory*, 45:1639–1642, 7 2000.