A Hybrid Semidefinite Programming - Sphere Decoding Approach to MIMO Detection

DIPLOMA THESIS

By

Dimitris Evangelinakis

Submitted to the Department of Electronic & Computer Engineering in partial fulfillment of the requirements for the Diploma Degree.

Technical University of Crete, Chania

Advisor: Professor Sidiropoulos Nikolaos

Co-advisor: Associate Professor Liavas Athanasios

Co-advisor: Assistant Professor Karystinos Georgios

April 2007

Contents

Chapter 1

Introduction

This thesis considers the development of a hybrid MIMO detection algorithm using *Semidefinite Programming (SDP)* and *Sphere Decoding (SD)* techniques. The aforementioned algorithm can be applied for multiuser detection in DS-CDMA systems, and for low complexity Maximum Likelihood (ML) decoding in multi-antenna communication systems.

The MIMO communication model, shown in figure (1), is constituted by a transmitter with M transmit antennas and a receiver with N receive antennas. An M-dimensional symbol vector $\mathbf{s}$, whose elements are integers drawn from a Finite Alphabet ($\mathcal{F}A$) constellation, which can be either real or complex, is transmitted through a linear time invariant $N \times M$ block fading channel $\mathbf{M}$. The elements of the channel matrix $\mathbf{M}$ (mixing matrix) are the channel coefficients (channel gains) m_{ij} of the corresponding paths. At the receiver, a $N \times 1$ vector $\mathbf{d}$ is observed, which is corrupted by Additive White Gaussian Noise (AWGN)

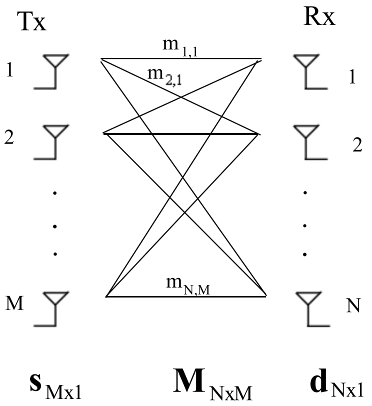

Figure 1.1: MIMO communication model architecture

vector of zero mean and covariance matrix $\sigma^2 I$, where I denotes the $N \times N$ identity matrix. Thus, this model, for a single time interval, can be formulated as:

$$\mathbf{d} = \mathbf{M}\mathbf{s} + \mathbf{n}, \quad \mathbf{s} \in \mathcal{F}\mathcal{A}^{\mathbf{M}} \tag{1.1}$$

where $\mathbf{s}$ is the $M \times 1$ transmitted symbol vector, $\mathbf{M}$ is the $N \times M$ baseband-equivalent channel matrix, $\mathbf{d}$ is the observed $N \times 1$ baseband-equivalent output vector and $\mathbf{n}$ is the $N \times 1$ Gaussian noise vector. Assume that the channel matrix $\mathbf{M}$ remains constant while transmitting a single block and is known at the receiver (full channel state information (full CSI-R)) but not at the transmitter. The goal at the receiver is to detect accurately the transmitted symbol vector from the received one, using the known channel matrix and the noise statistics. The Multiple-Input Multiple-Output (MIMO) model (1.1) is common in a number of modern communication systems. In addition to multi-antenna space-time systems, MIMO decoding is also encountered in multiuser spread-spectrum systems like synchronous Code Division Multiple Access (CDMA) receivers performing multiuser detection [1], [7].

1.1 ML Detection

For any constellation, ML detection in memoryless MIMO communication systems, in the presence of AWGN, boils down to the following problem:

$$\hat{\mathbf{s}}_{\mathbf{ML}} = \arg \min_{\mathbf{s}\in\mathcal{FA}^{\mathbf{M}}} ||\mathbf{d} - \mathbf{Ms}||_{\mathbf{2}}^{\mathbf{2}} \tag{1.2}$$

In this thesis, high-order QAM constellations (16-, 64-, 256-QAM etc.) are considered, so the variables are in general complex, $\mathbf{s} \in \mathcal{FA}^M$, $(\mathbf{M}, \mathbf{d}) \in (\mathcal{C}^{N\times M}, \mathcal{C}^{N\times 1})$ respectively. For separable, but not necessarily uniform, QAM constellations (squared PAM), we may define:

$$\begin{aligned} \mathbf{z} &:= [\Re\{d\}^T \ \Im\{d\}^T]^T, \\ \mathbf{H} &:= \begin{bmatrix} \Re\{\mathbf{M}\} & -\Im\{\mathbf{M}\} \\ \Im\{\mathbf{M}\} & \Re\{\mathbf{M}\} \end{bmatrix}, \\ \mathbf{r} &:= [\Re\{s\}^T \ \Im\{s\}^T]^T, \end{aligned} \tag{1.3}$$

and convert the problem to the real-valued form:

$$\hat{\mathbf{r}}_{\mathbf{ML}} = \arg \min_{\mathbf{r}\in\mathcal{FA}^{\mathbf{2M}}} ||\mathbf{z} - \mathbf{Hr}||_{\mathbf{2}}^{\mathbf{2}} \tag{1.4}$$

where $\mathbf{r} \in \mathcal{FA}^{2M}, \mathbf{H} \in \mathcal{R}^{2N\times 2M}, \mathbf{z} \in \mathcal{R}^{2N\times 1}$, where $\mathcal{FA}$ now denotes the PAM alphabet, employed for the real and imaginary part of the original, complex QAM constellation. In (1.4):

the vector $\mathbf{z}$ *and the matrix* $\mathbf{H}$ *are given, and the goal is to find the column vector* $\hat{\mathbf{r}} \in \mathcal{FA}^{2M}$ *that minimizes the squared Euclidean distance from* $\mathbf{z}$ *to* $\mathbf{Hr}$:

$$||\mathbf{z} - \mathbf{Hr}||_{\mathbf{2}}^{\mathbf{2}} \tag{1.5}$$

The above is variably known as lattice search, or *Integer Least Squares* (ILS) problem [15], because the elements of $\mathbf{r}$ are drawn from the integer lattice. The ML detector minimizes the probability of error for equiprobable symbol vectors. Viewing $\mathbf{z}$ as a point in the $2N$-dimensional space, (1.4) suggests searching exhaustively over all $|\mathcal{FA}|^{2M}$ candidate vectors $\mathbf{r}$ and selecting the one for which $\mathbf{Hr}$ lies closest to $\mathbf{z}$. Unfortunately, it is known that this solution requires exponential computation complexity in M, which is prohibitive in most practical scenarios. Specifically, problem (1.4) has been proved to be nondeterministic polynomial-time hard (NP-Hard) [1]. For this reason several computationally efficient algorithms have been developed in order to achieve (quasi-)optimal performance with relatively low computational cost. The current state-of-the-art includes two main families of quasi-ML detectors: Those based on *Semidefinite Relaxation* [7], [8], [9], [10], [11], and those based on *Sphere Decoding* [14], [18], [13], [19], [22].

1.1.1 Overview of common suboptimal methods

Apart from the SDR family of suboptimal algorithms, other heuristic approximation methods are employed to reduce the complexity of finding a (suboptimal in general) solution for the ILS problem (1.2). Some common approaches in the communication literature are discussed next [16].

1. Solve the unconstrained least-squares problem to obtain $\hat{\mathbf{s}} = \mathbf{pinv}(\mathbf{M})\mathbf{d}$, where $\mathbf{pinv}(\mathbf{M})$ denotes the pseudoinverse of $\mathbf{M} \quad ((\mathbf{M^T M})^{-\mathbf{1}}\mathbf{M^T})$. The entries of $\hat{\mathbf{s}}$ will not necessarily be integers, so a quantization to the nearest $\mathcal{FA}$ point is needed (a process referred to as slicing) to obtain

$$\hat{\mathbf{s}}_{\mathbf{B}} = \Big[(\mathbf{M^T M})^{-\mathbf{1}}\mathbf{M^T d}\Big]_{\mathcal{FA}^{\mathbf{M}}} \tag{1.6}$$

   The above estimate is also called the Babai estimate [28]. In communications theory, this procedure is also referred to as *zero-forcing equalization.*

2. *Nulling and cancelling.* In this method, the Babai estimate is used for only one of the entries of $\mathbf{s}$, say the first one (s_1). That entry is then assumed to be known and its effect is cancelled out to obtain a reduced-order ILS problem with $M-1$ unknowns. This process is repeated to find s_2, etc. This is similar to *decision-feedback equalization.*

3. *Nulling and cancelling with optimal ordering (Detection ordering).* Nulling and cancelling is sensitive in "error propagation"; if s_1 is estimated incorrectly it can have an adverse effect on the estimation of the remaining unknown symbols ($s_2, s_3 \ldots$). To minimize this effect, it is advantageous to perform nulling and cancelling from the "strongest" to the "weakest" symbol. This method is proposed in [29], [30].

The above heuristic methods all require $\mathcal{O}(MN^2)$ computations, essentially because they all first solve the unconstrained least-squares problem, assuming that the channel changes very slowly, so that the matrix inversion can be reused.

1.1.2 Measures of Performance

In this thesis, the state-of-the-art algorithms and the proposed one are compared, by means of three performance metrics; namely, *Symbol Error Rate (SER)*, *mean* and *worst-case execution time.*

The key detection performance indicator of a detector is the SER - the probability that a transmitted symbol has been mistaken for another one of the employed constellation.

Moreover, a detector has to be computationally efficient in order to be implementable in practical communication systems. For this reason, as measures of the computational cost of the various detection algorithms considered, we use the *mean* and the *worst-case execution time*[1].

1.1.3 Comparison of SDR and SD properties

The SD family of detectors [14], [18], [13], [19], [22], yields high quality (ML) solutions at low computational cost, provided that the Signal to Noise Ratio (SNR) is relatively high, and the aggregate transmission rate is relatively low.

[1] On an Intel Centrino Core Duo 1.83 GHz system, with 2GB RAM

On the other hand, SD cannot efficiently handle high problem dimensions (long symbol vectors) or high-order symbol constellations, especially at low SNR, and it has been recently shown that its expected complexity is exponential under mild conditions [26]. In practice, SD exhibits a phase transition: it either works very efficiently in terms of both complexity and detection performance, or it hopelessly "freezes".

SDR approaches [7], [8], [9], [10], [11], on the other hand, feature polynomial worst-case complexity and very competitive performance (quasi-ML). Initially, SDR multiuser / MIMO detection was developed for BPSK constellations, but the ideas were later extended to high-order QAM constellations [8]. The complexity of the SDR MIMO detector in [8], is nearly cubic in the dimension of the transmitted symbol vector, and independent of the constellation order for uniform QAM, affine in the constellation order for non-uniform QAM.

Still, when SD is operative, it often outperforms SDR in terms of complexity and SER performance. These observations motivate research in hybrid SDR-SD techniques, aiming to capture and leverage the best features of both families of detectors. Specifically, we may use SDR to speed up SD in difficult cases, as we will see in chapter 4.

1.2 Thesis Outline

The basic idea of this thesis is the description and implementation of SDR and SD algorithms in the MIMO detection problem with high-order QAM signaling and the development of a hybrid SDR-SD algorithm for the same reason. In the first chapter, the MIMO model and the detection in MIMO systems have been explained. Also, a brief reference to some of the simpler algorithms that have been developed for the detection procedure has been given. Chapter two is devoted to the convex optimization theory and semidefinite relaxation. In the first section of this chapter the basic concepts of convex optimization theory are reviewed. The family of convex optimization problems and its classes are introduced. The possibility of converting a non-convex engineering problem, like the one considered in this thesis, to a convex one by reformulating it or dropping / relaxing the non-convex constraints is explained, and how the resulting convex problem can be solved by efficient algorithms. Next, a link is drawn between convex relaxation the MIMO detection problem. The second part is devoted to the description of the semidefinite relaxation algorithm in [8] for solving the detection problem. The sphere decoder, with its variates and improvements, is explained in chapter three. A detailed description of this algorithm is given, analysing the way of choosing an appropriate radius and explaining how SD searches within a (hyper-)sphere to find the exact ML solution. In the sequel, the Schnorr-Euchner (SE) variate of SD is given, explaining how SE ordering increases the probability to find the exact ML solution earlier than the original SD. Also a short reference to improvements and heuristics such as the detection ordering (DO) preprocessing step is given. In the fourth chapter, the development of a hybrid SDR-SD algorithm is presented, which is the main contribution of this thesis. In chapter five, simulation results and figures are presented, along with a simple and efficient way of converting QAM to multidimensional BPSK constellations. In the figures, the SDR algorithm, SE-SD and the hybrid SDR-SD algorithm are compared regarding their error performance and their mean / worst-case execution time. Also a comparison with the fast SDR detector of Z. Luo and M. Kisialiou [27] is given. The aforementioned conversion from QAM to multidimensional BPSK constellations is employed to enable [27] to work with high-order QAM. Conclusions are presented along with the simulation results in chapter five.

Chapter 2

Convex Optimization and Semidefinite Relaxation

In the first section of this chapter the basic concepts of convex optimization theory are reviewed. The family of convex optimization problems and its classes are introduced. The conversion of a non-convex engineering problem, like the one considered in this thesis, to a convex one by reformulating it or dropping / relaxing the non-convex constraints is explained and how is being solved by efficient algorithms. Next, the meaning of convex relaxation is given and the link between this theory and the MIMO detection problem. The second part is devoted to the description of the semidefinite relaxation algorithm in [8] for solving the detection problem.

2.1 Convex Optimization Theory

2.1.1 Convex Problems

In general, an optimization problem with equality and inequality constraints can be expressed as follows :

$$\begin{aligned} \text{minimize}_x \quad & f_0(\mathbf{x}) & & \\ \text{subject to} \quad & f_i(\mathbf{x}) \leq 0 & & 1 \leq i \leq m \\ & h_i(\mathbf{x}) = 0 & & 1 \leq i \leq p \end{aligned} \tag{2.1}$$

where f_0 is the objective function, f_i are the inequality constraint functions, h_i are the equality constraint functions and $\mathbf{x} \in \mathcal{R}^n$ is the optimization variable [2]. When the objective and the inequality constraint functions are convex and the equality constraint functions are affine, then problem (2.1) is a *convex* optimization problem. A function $f : \mathcal{R}^n \to \mathcal{R}$ is said to be convex if for all $\mathbf{x_1}, \mathbf{x_2} \in \mathit{domf}$ (the domain of f) and $\lambda \in [0,1]$, $\lambda\mathbf{x_1} + (1-\lambda)\mathbf{x_2} \in \mathit{domf}$, and $f(\lambda\mathbf{x_1} + (1-\lambda)\mathbf{x_2}) \leq \lambda f(\mathbf{x_1}) + (1-\lambda)f(\mathbf{x_2})$. The set of all points $\mathbf{x}$ that satisfy the constraints is called *feasible set.* An optimization problem is said to be feasible if there exists at least one feasible solution, i.e. the feasible set is not empty; infeasible otherwise. For a convex optimization problem, the feasible set is convex. A set $S \subset \mathcal{R}^n$ is said to be convex if for any two points $\mathbf{x_1}, \mathbf{x_2} \in S$, the line segment joining $\mathbf{x_1}$ and $\mathbf{x_2}$ also lies in S. This property can be expressed mathematically as: $\lambda\mathbf{x_1} + (1-\lambda)\mathbf{x_2} \in S, \quad \forall\lambda \in [0,1] \quad \text{and} \quad \mathbf{x_1}, \mathbf{x_2} \in S$. Generaly, a convex set must be a solid body, containing no holes and always curve outward [6]. For example, convex sets are ellipsoids, hypercubes, polyhedrals etc. An important property of convex sets is that the intersection of any number of convex sets remains convex. On the other hand, the union of two convex sets is typically non-convex. The *optimal solution* of an optimization problem is the point $\mathbf{x}^*$ that belongs to the feasible set and yields the optimal (i.e minimum) value $f^* = f_0(\mathbf{x}^*)$.

2.1.2 Convex Classes and Problem Reformulation

Convex optimization problems do not entail local minima, which usually give rise to difficulties in optimization. Moreover, as mentioned earlier, convex problems can be solved very efficiently, in polynomial time, by powerful modern optimization algorithms like interior point methods [2], using software packages like SeDuMi [5].

There are various classes of convex problems, depending on the nature of the functions involved. Basic classes are:

1. *Linear Program* (LP): Where all functions are linear,
2. *Quadratic Program* (QP): Where f_0 is quadratic and f_i, h_i are linear,
3. *Quadratically Constrained Quadratic Program* (QCQP): Where f_0, f_i are quadratic and h_i are linear,
4. *Semidefinite Program* (SDP): Where f_0, h_i are linear and the f_i's are Linear Matrix Inequalities (LMIs).

We will focus our interest in the fourth class (SDP) because the MIMO detection problem can be easily relaxed to an SDP [3], which can be solved efficiently using interior point methods.

Unfortunately, not all problems are convex, as many engineering problems and MIMO detection is one such example. This is not necessarily bad in general. In many cases we may be able to properly reformulate a problem into a convex one by rewriting it, changing some variables or adding some others (auxiliary variables), unveiling its hidden convexity. It has to be noted that there is no systematic way to reformulate a problem in convex form, it is rather an art that can only be learned by examples.

2.1.3 Convex Relaxation

There are many engineering problems that cannot be converted to convex form - e.g. NP-hard problems. In such cases, the non-convex constraints can be dropped, resulting in a relaxed problem that is convex. Dropping constraints expands the feasible set. This means that more $\mathbf{x}$ points are allowed to solve the problem (minimize or maximize the objective function). Obviously, the optimal solution in this case will not always be the desirable one, due to the relaxation of the constraints, and cannot be directly used as an approximate solution of the original problem, because it may not lie in the original feasible set. Thus, we have to use some techniques to convert the solution of the relaxed problem to a respective solution of the original. A known and well working approximation technique is the Gaussian Randomization. Hence, a relaxation algorithm consists of two steps. First, solve the relaxed problem; then use an approximation algorithm in order to convert the relaxation solution to an approximate solution of the original problem.

In our case, reformulating the MIMO detection problem and dropping a non-convex constraint, we end up in a SDP and thus this method is called Semidefinite Relaxation. In the next paragraph the Semidefinite Relaxation method and the Gaussian Randomization are presented, and the SDR algorithm [8] for the MIMO detection problem is introduced.

2.2 Semidefinite Relaxation Algorithm

The ML detection problem in memoryless MIMO communication systems with AWGN, for any QAM constellation, can be expressed as the following optimization problem :

$$\begin{aligned} & \min_{s} ||\mathbf{d} - \mathbf{Ms}||_{\mathbf{2}}^{\mathbf{2}} \\ \text{subject to:} \quad & \Re\{\mathbf{s}(i)\} \in \mathcal{FA}_{real} \quad, \forall i \\ & \Im\{\mathbf{s}(i)\} \in \mathcal{FA}_{imag} \quad, \forall i \end{aligned} \tag{2.2}$$

where $\mathbf{d}$ is the complex baseband received vector, $\mathbf{M}$ is a known baseband-equivalent channel matrix, $\mathbf{s}$ the symbol vector and $\mathcal{F}A$ the symbol alphabet of the employed constellation. Converting problem (2.2) to a real valued form, as already described in (1.3) and (1.4), and assuming that $\mathcal{FA}_{real} = \mathcal{FA}_{imag} = \mathcal{FA}$

(i.e. “square” QAM constellations) in the sequel, problem (2.2) can be written as:

$$\begin{aligned} &\min_{r} ||\mathbf{z} - \mathbf{Hr}||_{\mathbf{2}}^{\mathbf{2}} \\ &\text{subject to:} \quad \mathbf{r}(i) \in \mathcal{FA} \quad ,\forall i \end{aligned} \tag{2.3}$$

Assume that the $\mathcal{F}A$ is symmetric about the origin (always valid for QAM constellations). If $\mathbf{r}$ satisfies the $\mathcal{F}A$ constraints then so does $t\mathbf{r}, t \in \{-1, 1\}$. Furthermore

$$||\mathbf{z} - \mathbf{Hr}||_{\mathbf{2}}^{\mathbf{2}} = \mathbf{r^T H^T H r} - \mathbf{2z^T H r} + \mathbf{z^T z} \tag{2.4}$$

It follows that the minimization in (2.3) subject to the corresponding constraints is equivalent to

$$\begin{aligned} &\min_{\mathbf{r}} (\mathbf{r^T H^T H r} - \mathbf{2z^T H}t\mathbf{r}) \\ &\text{subject to:} \quad \mathbf{r}(i) \in \mathcal{FA} \quad ,\forall i \\ &t \in \{-1, 1\} \end{aligned} \tag{2.5}$$

For brevity of exposition, assume that after the conversion of the problem to a real one, the transmitted symbol vector $\mathbf{r}$ is of dimension $M \times 1$, $\mathbf{z}$ is $N \times 1$ and $\mathbf{H}$ is $N \times M$. Moreover, by defining

$$\begin{aligned} &\mathbf{x} := [\mathbf{r^T} t]^{\mathbf{T}} \in \mathcal{R}^{\mathbf{M+1}} \quad \text{, and} \\ &\mathbf{Q} := \begin{bmatrix} \mathbf{H^T H} & -\mathbf{H^T z} \\ -\mathbf{z^T H} & 0 \end{bmatrix} \end{aligned} \tag{2.6}$$

the minimization problem (2.5) can be rewritten in homogeneous quadratic form

$$\begin{aligned} &\min_{\mathbf{x}} \mathbf{x^T Q x} \\ \text{s.t :} \quad &\mathbf{x}(i) \in \mathcal{FA}, \forall i \in \{1, \ldots, M\} \\ &\mathbf{x}(M+1) \in \{-1, 1\} \end{aligned} \tag{2.7}$$

Using the property $\mathbf{x^T Q x} = \mathit{Trace}(\mathbf{x^T Q x}) = \mathit{Trace}(\mathbf{Q x x^T})$, and denoting $\mathbf{X} := \mathbf{x x^T}$ (that means $\mathbf{X}$ is symmetric, positive semidefinite and of rank 1), problem (2.7) can be equivalently rewritten as:

$$\begin{aligned} &\min_{X} Trace(\mathbf{QX}) \\ \text{s.t :} \quad &\mathbf{X} \succeq \mathbf{0}, rank(\mathbf{X}) = 1, \\ &\mathbf{X}(i,i) \in \mathcal{FA}^2, \forall i \in \{1, \ldots, M\}, \mathbf{X}(M+1, M+1) = 1 \end{aligned} \tag{2.8}$$

Problem (2.8) is not a Convex Optimization Problem because it contains the following non-convex constraints :

1. $rank(\mathbf{X}) = 1$, (rank-one constraint)

2. $\mathbf{X}(i,i) \in \mathcal{FA}^2, \forall i \in \{1, \ldots, M\}$, (Finite Alphabet constraint)

Dropping the rank-one constraint, and relaxing the Finite Alphabet constraint $\mathbf{X}(i,i) \in \mathcal{FA}^2, \forall i \in \{1, \ldots, M\}$ to the convex half-space constraints $\mathbf{L} := \min_{a \in \mathcal{FA}} a^2 \leq \mathbf{X}(i,i) \leq \max_{a \in \mathcal{FA}} a^2 =: \mathbf{U}, \forall i \in \{1, \ldots, M\}$, the following convex relaxation problem is derived:

$$\begin{aligned} &\min_{X} Trace(\mathbf{QX}) \\ &\text{s.t}: \quad \mathbf{X} \succeq 0, \\ &\mathbf{L} \leq \mathbf{X}(i,i) \leq \mathbf{U}, \forall i \in \{1,\ldots,M\}, \\ &\mathbf{X}(M{+}1, M+1) = 1 \end{aligned} \tag{2.9}$$

The relaxed problem (2.9) can be solved by using any software SDP solver, such as SeDuMi [5], based on interior point methods. Note that, as already mentioned in section (2.1.3), the solution of (2.9) is not necessarily an appropriate solution to the original problem (2.8), due to the relaxation. After this step, an approximate solution to the original problem can be generated using *Gaussian Randomization*, as described in [8] (also in [7]). This simple method draws random vectors $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{X_0})$, where $\mathbf{X_0}$ denotes the solution of the relaxed problem (in our case, problem (2.9)). Afterwards, each element of $\mathbf{x}$ is quantized to the nearest point in $\mathcal{FA}$, $\mathbf{r}$ is reconstructed from the quantized $\mathbf{x}$ and the one that yields the smallest cost (after a fixed number of trials) in (2.2) is picked. Note that in general $\mathbf{X_0} \neq \mathbf{xx^T}$, where $\mathbf{X_0}$ denotes the relaxation solution and $\mathbf{x}$ the solution after the Gaussian Randomization step and before the reconstruction of $\mathbf{r}$. Note that in case that matrix $\mathbf{X_0}$ (the relaxed problem solution given by the SDP solver) is of rank 1, then the solution to the original problem is the principal component of matrix $\mathbf{X_0}$ and the randomization method is not needed.

The computation complexity of this algorithm is $\mathcal{O}(M^{3,5})$, where M is the dimension of the transmitted symbol vector $\mathbf{r}$.

Chapter 3

The Sphere Decoder

The Sphere Decoding (SD) Algorithm (Sphere Decoder), first introduced in [12] by Finke and Phost, has been applied for lattice decoding in [14], [13], multi-antenna systems [18], and MIMO decoding [19], [20], where it is used to compute the (quasi-) ML symbol vector estimate with moderate computational complexity. It has also been proposed for the multiuser detection problem in synchronous CDMA systems [21]. Since then, many improvements have been developed in order to reduce its computation complexity, and heuristics for returning the exact ML solution with high probability [22], [23], [24], [25]. In this chapter, a detailed description of SD and the Schnorr-Euchner (SE) variate of SD is given. In this thesis, the SE-SD is used for all simulations and comparisons, due to its improved complexity performance.

3.1 The Idea behind Sphere Decoding

Consider again the *real* MIMO model :

$$\mathbf{z} = \mathbf{H}\mathbf{r} + \mathbf{v}, \quad \mathbf{r} \in \mathcal{F}\mathcal{A}^{\mathbf{M}} \tag{3.1}$$

where $\mathbf{r}$ is the $M \times 1$ transmitted symbol vector, $\mathbf{H}$ is the $N \times M$ baseband-equivalent channel matrix, $\mathbf{z}$ is the observed $N \times 1$ baseband-equivalent output vector and $\mathbf{v}$ is the $N \times 1$ Gaussian noise vector. This model is derived from the concatenation of the real and imaginary parts of the complex MIMO model (1.1), using the simple method described in the first chapter (1.3). The dimensions of the complex problem are half of those of the real one, for notational brevity. As already explained in chapter 1, viewing $\mathbf{z}$ as a point in the N-dimensional space, the solution of problem (1.4) requires exhaustive search over all possible $|\mathcal{F}\mathcal{A}|^M$ candidate vectors $\mathbf{r}$ and selecting the one for which $\mathbf{H}\mathbf{r}$ lies closest to $\mathbf{z}$. This procedure requires exponential complexity in the dimension of the transmitted vector $\mathbf{r}$, a prohibitive process for embedded communication systems.

As its name indicates, SD searches within a (hyper-)sphere of radius C, centered at the received vector $\mathbf{z}$. This implies that SD accounts for all $\mathbf{H}\mathbf{r}$ candidates that lie inside the hyper-sphere of radius C. The noise vector $\mathbf{v}$ is the reason of the distance between $\mathbf{z}$ and $\mathbf{H}\mathbf{r_0}$, where $\mathbf{r_0}$ denotes the *true* transmitted symbol vector. The SD has three basic steps. First, find an appropriate radius that guarantees that at least one candidate solution lies inside the hyper-sphere. Next, initialize the search with the unconstrained least-squares solution and then

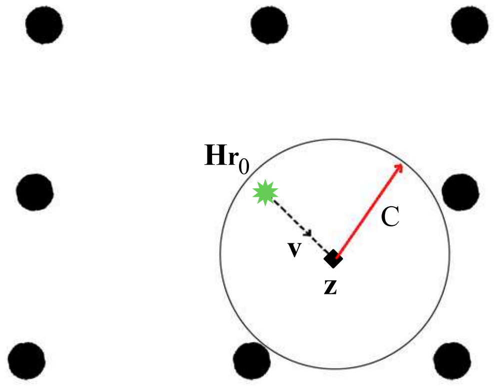

Figure 3.1: Illustration of sphere decoding

search exhaustively but efficiently within the sphere of fixed radius C.

3.2 Selecting a Radius and Initializing with the Unconstrained LS

If the radius C is large enough, i.e infinite, then there is no difference (in terms of the final estimate) between sphere decoding and searching exhaustively. On the other hand, if the radius is too small, the sphere is possibly empty (contains no **Hr** candidate vectors). The radius has to be appropriately chosen, i.e. small enough in order to search within the sphere with moderate complexity, and large enough to guarantee that the sphere is not empty of candidates. For AWGN, the probability that $\mathbf{Hr_0}$ lies within a sphere of fixed radius C, is given by $Pr(||\mathbf{z} - \mathbf{Hr_0}||_2^2 \leq C^2) = Pr(||\mathbf{v}||_2^2 \leq C^2)$. Under the AWGN model, $||\mathbf{v}||_\mathbf{2}^\mathbf{2}$ is chi-square distributed.

When n standard normal distributed random variables are squared and *summed, a chi-square random variable emerges, with probability density function (pdf):*

$$f_V(u) = \frac{1}{(2\sigma^2)^{\frac{M}{2}}\Gamma(\frac{M}{2})} u^{\frac{M}{2}-1} e^{-\frac{u}{2\sigma^2}}, \quad u \geq 0 \tag{3.2}$$

The point $\mathbf{Hr_0}$ is contained with probability p in a sphere centered at $\mathbf{z}$ and radius that can be calculated by solving the following equation with respect to C_p

$$Pr(||\mathbf{v}||_2^2 \leq C_p^2) = \int_0^{C_p^2} f_V(u)du = p \tag{3.3}$$

The physical meaning of this method is that the calculated radius generates a sphere that contains the ML solution with probability p. For instance, with $p = 0.99$, the engineer knows that if SD searches within a sphere of radius $C_{0.99}$, it will return the ML solution with probability 0.99. This method ensures that the ML solution will be found within the sphere with the given probability but does not guarantee that it will be done with polynomial complexity.

The next question is how SD searches within the sphere among all candidates to find the vector that minimizes the error norm (1.5). The starting point of SD is the unconstrained least-squares solution $\hat{\mathbf{r}} = (\mathbf{H^T H})^{-1}\mathbf{H^T z}$. The LS solution is obtained using the QR-decomposition of channel matrix $\mathbf{H}$. SD uses the last entry of the soft least-squares(LS) estimate to search backwards for the remaining candidate symbols.

Note that SD does not search within a sphere centered at $\mathbf{z}$ but within an equivalent one centered at $\mathbf{H\hat{r}}$, where $\mathbf{\hat{r}}$ denotes the soft LS estimate here [4]. Below follows a proof of the equivalence $||\mathbf{z}-\mathbf{Hr}||_{\mathbf{2}}^{\mathbf{2}} = ||\mathbf{R}(\mathbf{r}-\mathbf{\hat{r}})||_{\mathbf{2}}^{\mathbf{2}} + c$, where $\mathbf{R}$ denotes the $M \times M$ upper triangular matrix of the QR-decomposition of matrix $\mathbf{H}$, and c a constant that does not depend on $\mathbf{r}$.

Let $\mathbf{P} := \mathbf{H}(\mathbf{H}^{\mathbf{T}}\mathbf{H})^{-\mathbf{1}}\mathbf{H}^{\mathbf{T}}$ or $\mathbf{P} := \mathbf{H}\mathbf{H}^{\dagger}$, where $\mathbf{H}^{\dagger}$ denotes the pseudoinverse of matrix $\mathbf{H}$, $\quad (\mathbf{H}^{\mathbf{T}}\mathbf{H})^{-\mathbf{1}}\mathbf{H}^{\mathbf{T}}$. Also let $\mathbf{P}^{\perp} := \mathbf{I}{-}\mathbf{H}\mathbf{H}^{\dagger}$. $\mathbf{P}$ denotes the orthonormal matrix that when multiplied with a vector from left it projects it onto the column space of $\mathbf{H}$, and $\mathbf{H}^{\perp}$ denotes its orthogonal complement. Using the latter it holds:

$$\begin{aligned}
&||\mathbf{z}-\mathbf{Hr}||_{\mathbf{2}}^{\mathbf{2}} = ||\mathbf{Hr}-\mathbf{z}||_{\mathbf{2}}^{\mathbf{2}} = ||\mathbf{Hr}-(\mathbf{P}+\mathbf{P}^{\perp})\mathbf{z}||_{\mathbf{2}}^{\mathbf{2}} = \\
&||\mathbf{Hr}-\mathbf{Pz}-\mathbf{P}^{\perp}\mathbf{z}||_{\mathbf{2}}^{\mathbf{2}} = (*), \quad \mathbf{z} = \mathbf{Hr_0}+\mathbf{v}
\end{aligned} \tag{3.4}$$

$\mathbf{PHr_0} = \mathbf{Hr_0}$ because $\mathbf{Hr_0}$ already lies in the column space of $\mathbf{H}$ as a linear combination of the columns of $\mathbf{H}$. Thus :

$$\begin{aligned}
&(*) = ||\mathbf{Hr}-\mathbf{Hr_0}-\mathbf{Pv}-\mathbf{P}^{\perp}\mathbf{v}||_{\mathbf{2}}^{\mathbf{2}} = (**) \\
&\mathbf{\hat{r}} = \mathbf{H}^{\dagger}\mathbf{z} = \mathbf{H}^{\dagger}(\mathbf{Hr_0}+\mathbf{v}) = \mathbf{r_0}+\mathbf{H}^{\dagger}\mathbf{v} \Rightarrow \\
&\mathbf{H\hat{r}} = \mathbf{Hr_0}+\mathbf{HH}^{\dagger}\mathbf{v} \Leftrightarrow \mathbf{H\hat{r}} = \mathbf{Hr_0}+\mathbf{Pv} \\
&(**) = ||\mathbf{Hr}-\mathbf{H\hat{r}}-\mathbf{P}^{\perp}\mathbf{v}||_{\mathbf{2}}^{\mathbf{2}} = ||\mathbf{H}(\mathbf{r}-\mathbf{\hat{r}})-\mathbf{P}^{\perp}\mathbf{v}||_{\mathbf{2}}^{\mathbf{2}} \Rightarrow \\
&||\mathbf{z}-\mathbf{Hr}||_{\mathbf{2}}^{\mathbf{2}} = ||\mathbf{H}(\mathbf{r}-\mathbf{\hat{r}})||_{\mathbf{2}}^{\mathbf{2}} + ||\mathbf{P}^{\perp}\mathbf{v}||_{\mathbf{2}}^{\mathbf{2}}
\end{aligned} \tag{3.5}$$

The last equivalence is derived using the Pythagoras' Theorem - *the squared norm of the difference of two orthogonal vectors, is the sum of their squared norms.*

Using the orthonormality of factor $\mathbf{Q}$ of the QR-decomposition:

$$||\mathbf{z} - \mathbf{Hr}||_{\mathbf{2}}^{\mathbf{2}} = ||\mathbf{R}(\mathbf{r} - \hat{\mathbf{r}})||_{\mathbf{2}}^{\mathbf{2}} + c \tag{3.6}$$

Note that the constant c vanishes in the square case ($M = N$, equal number of transmit and receive antennas) because matrix $\mathbf{H}$ is square and of full column rank, thus its columns span the whole space and therefore, its orthogonal complement is zero. This implies that :

$$\hat{\mathbf{r}}_{\mathbf{ML}} = \arg \min_{\mathbf{r} \in \mathcal{FA}^{\mathbf{M}}} ||\mathbf{z} - \mathbf{Hr}||_{\mathbf{2}}^{\mathbf{2}} = \arg \min_{\mathbf{r} \in \mathcal{FA}^{\mathbf{M}}} ||\mathbf{R}(\mathbf{r} - \hat{\mathbf{r}})||_{\mathbf{2}}^{\mathbf{2}} \tag{3.7}$$

With the proof of (3.7) the equivalence between the two spheres is established. Moreover, since $\hat{\mathbf{r}}$ is a projection of $\mathbf{z}$ in the “signal range space”, searching within a sphere centered at $\mathbf{R}\hat{\mathbf{r}}$ or equivalently at $\mathbf{H}\hat{\mathbf{r}}$, instead of one centered at $\mathbf{z}$, results in noise mitigation by providing an initialization orthogonal to the “noise subspace” [4]. This mitigation increases with the difference $N - M$, where M, N denote the number of transmit and receive antennas respectively. Also, SD can search within the sphere recursively with respect to the dimension of $\mathbf{r}$, using the upper triangular structure of matrix $\mathbf{R}$.

3.3 Sphere Decoding

SD searches for the ML solution only between candidate vectors $\mathbf{r}$ that fulfill the condition $||\mathbf{R}(\mathbf{r} - \hat{\mathbf{r}})||_2^2 < C^2$. Expressing the norm $||\mathbf{R}(\mathbf{r} - \hat{\mathbf{r}})||_{\mathbf{2}}^{\mathbf{2}}$ component-wise, starting from the last (Mth) entry and moving backwards to the first:

$$||\mathbf{R}(\mathbf{r}-\hat{\mathbf{r}})||_2^2 = R_{M,M}^2(r_M-\rho_M)^2 + R_{M-1,M-1}^2(r_{M-1}-\rho_{M-1})^2 + \ldots + R_{1,1}^2(r_1-\rho_1)^2,$$
$$\rho_k := \hat{r}_k - \sum_{j=k+1}^{M}(r_j-\hat{r}_j)\frac{R_{k,j}}{R_{k,k}} \quad \text{for} \quad 1 \le k \le M. \tag{3.8}$$

In equation (3.8), the contribution of each symbol to the norm ($||\mathbf{R}(\mathbf{r}-\hat{\mathbf{r}})||_2^2$) increment is explicit. Furthermore, for each symbol, the partial norm increment, depends on the already decided symbols of higher dimensions. For example, symbol r_{M-2} increases the norm for $R_{M-2,M-2}^2(r_{M-2}-\hat{r}_{M-2}+\sum_{j=M-1}^{M}(r_j-\hat{r}_j)\frac{R_{M-2,j}}{R_{M-2,M-2}})$.

Now, let's explain how does SD choose a candidate symbol from the $\mathcal{FA}$ for each dimension. Remember that SD starts the search from the last entry of $\mathbf{r}$ and goes step by step to the top of the vector, if this is possible. Condition $||\mathbf{R}(\mathbf{r}-\hat{\mathbf{r}})||_2^2 < C^2$ explains how this works. This inequality constrains the last entry r_M of $\mathbf{r}$ to be such that $R_{M,M}^2(r_M-\rho_M)^2 < C^2$, or equivalently, by expanding the squares :

$$\lceil \hat{r}_M - \frac{C}{R_{M,M}} \rceil \le r_M \le \lfloor \hat{r}_M + \frac{C}{R_{M,M}} \rfloor, \tag{3.9}$$

where $\lceil . \rceil$, $\lfloor . \rfloor$ denote the ceiling and the floor operator, respectively. At this step, SD computes a lower and an upper bound for the candidate symbol and searches in the $\mathcal{FA}$ for symbols that lie inside the interval defined by the computed bounds. Then it selects a candidate symbol from the admissible list, say the one closest to the lower bound, which is denoted as $r_M^{(1)}$, and proceeds to the next dimension ($M-1$th entry of $\mathbf{r}$). If none of the $\mathcal{FA}'s$ symbols lie inside the interval (the number of the candidates (n_{r_M}) of the admissible list is zero), then SD declares infeasibility. The admissible list of candidate symbols per dimension can be empty ($n_{r_M} = 0$), or as large as $\mathcal{FA}$ is ($n_{r_M} = |\mathcal{FA}|$). The candidate symbols for the next $(M-1)$ dimension must satisfy the condition (inequality) $R^2_{M-1,M-1}(r_{M-1} - \rho_{M-1})^2 + R^2_{M,M}(r^{(1)}_M - \rho_M)^2 < C^2$, which constrains the candidate symbols of this entry of $\mathbf{r}$ to lie in an interval defined by the $r^{(1)}_M$'s value chosen at the previous step, the fixed radius C and the entries of $\mathbf{R}, \hat{\mathbf{r}}$ obtained from the initialization step.

Generally, for each entry, the condition that the candidate symbols must satisfy can be expressed as :

$$\begin{aligned} R^2_{k,k}(r_k - \rho_k)^2 + \sum_{i=k+1}^{M} R^2_{i,i}(r_i - \rho_i)^2 < C^2, or \\ R^2_{k,k}(r_k - \rho_k)^2 < C^2 - \sum_{i=k+1}^{M} R^2_{i,i}(r_i - \rho_i)^2. \end{aligned} \tag{3.10}$$

This expresses also that the sum of the partial norm increments of each symbol must not exceed C^2. Given the selected candidate of the previous dimensions $\{r_i\}^M_{i=k+1}$, the condition that constraints the admissible candidates of symbol r_k can be written as :

$$\lceil \rho_k - \frac{\tau_k}{R_{k,k}} \rceil \leq r_k \leq \lfloor \rho_k + \frac{\tau_k}{R_{k,k}} \rfloor, \quad k = M, \ldots, 1, \tag{3.11}$$

where $\tau_k := \sqrt{C^2 - \sum_{i=k+1}^{M} R^2_{i,i}(r_i - \rho_i)^2}, \quad k = M-1, \ldots, 1,$ represents the radius reduction after deciding for the candidates $\{r_i\}^M_{i=k+1}$.

If the search procedure flows normally, in the way described above, SD reaches the first entry (r_1) of $\mathbf{r}$ and decides for a candidate symbol. Then, stacking all collected candidates per dimension in a vector, yields an admissible candidate vector $\mathbf{r}^{(1)} = [r_1^{(1)}, r_2^{(1)}, \ldots, r_{M-1}^{(1)}, r_M^{(1)}]^T$ (Admissible here means that $\mathbf{r}^{(1)}$ obeys the $\mathcal{FA}$ constraints and $||\mathbf{z} - \mathbf{H}\mathbf{r}^{(\mathbf{1})}||_2^2 < C^2$). This does not imply that $\mathbf{r}^{(1)}$ is necessarily the ML solution. SD just found a lattice point that lies within the sphere of fixed radius C. SD can use this solution to search within a new sphere, centered at the same point $\mathbf{R}\hat{\mathbf{r}}$, but with a new, smaller radius $C^{(1)} := ||\mathbf{R}(\mathbf{r}^{(\mathbf{1})} - \hat{\mathbf{r}})|| < C$, for another solution, possibly the ML one. This procedure continues until SD searches within a sphere that is empty of candidate vectors. This implies that the last vector $\mathbf{r}$ found, is the ML solution.

While searching per dimension for candidate symbols, if SD reaches a dimension k for which the list of symbol candidates is empty, it returns back to the previous dimension $k + 1$, selects another symbol from the candidates list and proceeds again to dimension k. SD generates a new candidate symbol list for dimension k, selects one from the new list and proceeds toward dimension 1.

This kind of search is also called depth-first search. SD while searching generates a tree where the nodes (leafs) are the symbols picked from the candidates list. At each branch, SD computes the cost (norm increment) that is summed with the total cost of the previous branches. The algorithms that use this kind of search are also called *branch and bound* algorithms.

3.4 Sphere Decoder's Complexity

The complexity of the sphere decoder is proportional to the number of nodes visited by the algorithm. The complexity also depends on the value of the initial radius. In [26] it is shown that an exponential lower bound on the complexity of SD is given by

$$C(M) \geq \frac{|\mathcal{FA}|^{\eta M} - 1}{|\mathcal{FA}| - 1}, \quad \eta = \frac{1}{4\rho_r + 2} \tag{3.12}$$

where ρ_r is an upper bound for the SNR value, and SD's initial radius is a function of SNR (see [4]). This proves that the expected complexity of SD cannot be polynomial, since it is lower bounded by an exponential function in M [26]. Moreover, sphere decoder's expected complexity is a decreasing function of SNR and an increasing function of the data transmition rate.

3.5 SD with Detection Ordering

As already described in chapter's introduction, many preproccesing steps and / or heuristics have been developed in order to reduce SD's complexity and / or improve the algorithm's error performance. A simple method of detection ordering (DO), also described in paragraph (1.2.1, number 3), is based on the idea of detecting at first symbols with large SNR. DO sorts the columns of channel matrix **H** in ascending order according to their SNR values (the column with the largest SNR value is ordered last). The entries of each column represent the gain of each sub-channel. The last column will be the one with the largest SNR value. The SNR comparison between the columns of $\mathbf{H}$ is performed by computing the squared Euclidean norm of each column ($||\mathbf{h}_k||_2^2$). The ascending order of sorting is employed due to SD's property to search backwards in the dimension of $\mathbf{r}$.

3.6 Schnorr-Euchner variate of SD

The Schnorr-Euchner (SE) variate of SD differs from the original algorithm in the ordering of the admissible symbol candidates list. Remember that the original SD selects at each dimension the symbol closest to the lower bound every time and if this results in an empty candidate list for the next dimension, returns back and picks the next one. SE-SD on the other hand, applies a different, more sophisticated ordering for the candidates list. According to the SE ordering, the first candidate in the list of a dimension (say kth), is the r_k value from the $\mathcal{FA}$ that lies closest to ρ_k, namely $r_k^{(1)} = \lceil \rho_k \rfloor$. The r_k chosen is the one that lies in the middle of the interval defined by the admissible candidates (3.11). This choice also leads to the smallest increment of the squared distance $R_{k,k}^2(r_k - \rho_k)^2$ among all candidates in the list of dimension k. Note that, as in the original SD, the decision of r_k depends on the decisions of the previous dimensions. If all the previous symbols are perfectly detected, $\lceil \rho_k \rfloor$ is the most likely candidate at dimension k, since it is the one that results in the smallest increment of the LS error norm ($||\mathbf{R}(\mathbf{r} - \hat{\mathbf{r}})||_2^2$ or equivalently $||\mathbf{z} - \mathbf{H}\mathbf{r}||_2^2$) and under the AWGN assumption, minimum distance translates to ML. The second candidate in the list of the kth dimension is the second closest point to the middle of the interval defined by (3.11). This is either $\lceil \rho_k \rfloor - 1$, in case that $\rho_k \leq \lceil \rho_k \rfloor$, or $\lceil \rho_k \rfloor + 1$ in case that $\rho_k \geq \lceil \rho_k \rfloor$. Sorting all n_{r_k} symbol candidates according to SE ordering, SE-SD generates a list of admissible symbol candidates with order :

$$[r_k^{(1)}, r_k^{(2)}, r_k^{(3)}, r_k^{(4)}, \ldots] = [\lceil \rho_k \rfloor, \lceil \rho_k \rfloor - 1, \lceil \rho_k \rfloor + 1, \lceil \rho_k \rfloor - 2, \ldots], \tag{3.13}$$

in case $\rho_k \leq \lceil \rho_k \rfloor$, or exchanging the order of candidate $\lceil \rho_k \rfloor - n$ with the order of candidate $\lceil \rho_k \rfloor + n$ in case that $\rho_k \geq \lceil \rho_k \rfloor$. The improvement of SD with the SE ordering is that the the sorted list in (3.13) increases the likelihood to find early an admissible vector $\mathbf{r}$ for which $\mathbf{Rr}$ lies very close to $\mathbf{R\hat{r}}$, or equivalently $\mathbf{Hr}$ lies very close to $\mathbf{z}$.

It is obvious that SE-SD picks a symbol from the list for a dimension, without taking into consideration the fixed radius C. After deciding for a dimension, SE-SD computes the LS error norm increment of this symbol ($R_{k,k}^2(r_k - \rho_k)^2$), adds it to the total computed distance (the LS error norm increments of the previous dimensions) and checks if it exceeds the squared radius C^2. In case that the computed total distance $d^2 = d_M^2 + d_{M-1}^2 + d_{M-2}^2 + \ldots + d_k^2$, where $d_k^2 := R_{k,k}^2(r_k - \rho_k)^2$, exceeds C^2, SE-SD returns back to a dimension that has not already examined all the list's candidates (remember, the number of symbol trials per dimension is stored in n_{r_k}). Note that the algorithm does not pick another symbol from the list of the current dimension, because SE ordering sorts the candidate symbols in ascending order according to their squared-distance increment. According to counter n_{r_k}, SE-SD picks the next candidate from the list, computes its LS error norm increment, adds it to the total distance for the kth dimension, checks the radius constraint and proceeds backwards toward dimension 1. Note that SE-SD sets the counter n_{r_k} for the remaining dimensions to zero and starts to pick symbols from the beginning of their lists. Also note that the total distance for the kth dimension is recomputed, using stored value of the total distance for dimension $k-1$. Actually, SE-SD stores (and SD in general) for each dimension value d_k^2. Therefore, the memory that SE-SD needs to store the distances is only linear in the dimension of M.

When SE-SD reaches dimension 1, it constructs a vector $\mathbf{r} = [r_1, r_2, \ldots, r_M]^T$ using the symbol decisions for each dimension, stores its total squared distance $d^2 = \sum_{k=1}^{M} R_{k,k}^2 (r_k - \rho_k)^2 = ||\mathbf{R}(\mathbf{r} - \hat{\mathbf{r}})||_2^2$ and updates the fixed radius by setting its value to the reduced one of d^2. This means that SE-SD uses this estimate (vector $\mathbf{r}$), and more specifically its computed squared distance, to search again within a new sphere of reduced radius $C'^2 = d^2 = ||\mathbf{R}(\mathbf{r} - \hat{\mathbf{r}})||_2^2 < C^2$. After constructing and returning vector $\mathbf{r}$, SE-SD from dimension 1, returns back to a dimension that has not already examined all the list's candidates and picks another candidate symbol from its list, according to n_{r_k}. After that, SE-SD proceeds toward dimension 1 in the same way described in the previous paragraph. Reaching dimension 1 again, SE-SD constructs a new estimate for vector $\mathbf{r}$ that will be used for searching within a new sphere of reduced radius. This procedure continues until SE-SD searches within a sphere empty of candidate vectors. Then, the last estimate found, is the ML solution of the initial problem (1.4).

Pseudocode of SE-SD is provided in the following table. Preprocessing steps such as DO and QR decomposition applied to the channel matrix $\mathbf{H}$ are not included. Function *enum(.)* takes as arguments the values of $\rho_k, \mathcal{FA}$ and n_{r_k} (number of candidates that have been enumerated), returns a candidate symbol picked from the admissible list according to n_{r_k} and increases it by one. This algorithm reduces the radius whenever a candidate vector with smaller LS error is found.

Table 3.1: Schorr-Eucher Sphere Decoding Pseudocode

```
$[\mathbf{r}_{ML}, d] = \text{SE-SD}(\mathbf{R}, \hat{\mathbf{r}}, C, \mathcal{FA}, M)$

$d := C^2; k := M; d_k^2 := 0;$ calculate $\rho_k$;
$[r_k, n_{r_k}] :=$ enum$(\rho_k, \mathcal{FA}, 0)$;
while(1){
$w := d_k^2 + R_{k,k}^2(r_k - \rho_k)^2$
if$(w < d)${
    if$(k > 1)${
        $k := k - 1; d_k^2 := w; \text{calculate}\rho_k$
        $[r_k, n_{r_k}] :=$ enum$(\rho_k, \mathcal{FA}, 0)$;
    }else{$\mathbf{r}_{ML} := [r_1, r_2, \ldots, r_M]; d := w;$ goto Rollback;}
}else{ Rollback:
    while$(++k < M \quad \&\& \quad n_k \geq |\mathcal{FA}|)${if$(k == M)$ break;}
    if$(k == M)${
        if$(d < C^2)${return $[\mathbf{r}_{ML}, d]$;}else{return $[\mathbf{0}, -1]$;}}
    $[r_k, n_{r_k}] :=$ enum$(\rho_k, \mathcal{FA}, n_{r_k})$;
} }
```

Chapter 4

Hybrid Semidefinite Programming - Sphere Decoding Algorithm

In chapters 2 and 3, the Semidefinite Relaxation and the Schorr-Euchner Sphere Decoder algorithms were introduced and related to the MIMO detection problem. In this chapter, a hybrid algorithm, based on SDR preprocessing and sphere decoding, is introduced. The properties of the two state-of-the-art algorithms (SDR and SD) are described in paragraph (1.2.3). SD (even SE-SD) "freezes" at high problem dimensions (long transmitted vectors / high order constellations) especially at low SNR environments, but guarantees ML estimates fot sufficiently large radius; in contrast, SDR that has a fixed complexity behaviour at any system and environment, yields a suboptimal, albeit near-ML solution. Aiming

Table 4.1: Hybrid SDR-SD algorithm

1. (Preprocessing step) Run SDR for problem (1.4) to obtain $\mathbf{r}_{SDR}$
2. Run SE-SD with initial radius $C^2 := ||\mathbf{z} - \mathbf{H}\mathbf{r}_{SDR}||_2^2$ to obtain $\mathbf{r}_{ML}$

to speed up SE-SD in difficult cases and obtain the ML solution in any case, it is reasonable to perform a preprocessing step, applying SDR to the MIMO detection problem. The SDR algorithm will yield a quasi-ML solution $\mathbf{r}_{SDR}$ to the problem (1.4). This solution has a LS error given by $||\mathbf{z} - \mathbf{H}\mathbf{r}_{SDR}||_2^2$. This LS error can be interpreted as a distance of the SDR estimate to the received vector $\mathbf{z}$. This distance can be used as a radius to generate a hyper-sphere that SE-SD will search within, to yield the ML solution. The suboptimality of the SDR estimate guarantees that the ML solution of the MIMO detection problem is at a distance that is less than, or equal to the SDR one, and thus subsequently applying SE-SD with radius $C := ||\mathbf{z} - \mathbf{H}\mathbf{r}_{SDR}||_2$, will yield the ML estimate. The basic steps of the hybrid SDR-SD algorithm are shown in table (4.1).

The complexity of SE-SD is reduced using a small enough radius, which however is guaranteed to contain the exact ML solution. Note that the first stage of this algorithm (preprocessing step) runs the SDR detector. This implies that a $\mathcal{O}(M^{3.5})$ complexity is added to the overall complexity of the hybid algorithm. Therefore, the complexity of the hybrid SDR-SD algorithm is $\mathcal{O}(M^{3.5})+$ SE-SD's comlexity.

Chapter 5

Simulations and Results

In the first part of this chapter, the algorithms considered in this thesis are simulated in order to compare their error performance and their computational efficiency and applicability. The algorithms run for detecting the transmitted symbol vector over a MIMO memoryless block fading channel. Two communication scenarios were used. The first was a MIMO system with 16 transmit and 16 receive antennas ($M = N = 16$) using 64-QAM signaling, and the second a MIMO system with 8 transmit and 8 receive antennas ($M = N = 8$) using 256-QAM signaling. The Monte-Carlo (MC) simulation method was used for all scenarios, in order to estimate the average SER and mean/worst case execution time. For every MC trial, a new transmission block (symbol vector), channel matrix and noise vector realization, were generated. In all cases, the channel matrix has i.i.d elements drawn from a circulary symmetric zero-mean complex normal distribution of unit variance $\mathcal{CN}(0, 1)$. The simulations were conducted for an SNR range from $15dB$ to $35dB$ (with step 5) and for 1000 MC runs for every SNR value. The SNR is defined as $SNR := 10log_{10}(ME_s/N_0)$, where M is the length of the transmitted QAM symbol vector, E_s the the mean symbol energy of the QAM constellation and the noise vector is i.i.d $\mathcal{CN}(0, N_0)$.

In the second part of this chapter, the hybrid SDR-SD algorithm is compared with the fast SDR BPSK detector of Z.-Q. Luo and M. Kisialiou [27]. The simulation of these algorithms was made over a MIMO channel with 16 transmit and 16 receive antennas with 16-QAM signaling. Note that in order to use the fast SDR detector [27], the 16-QAM constellation has to be converted into a BPSK one, since [27] works only with BPSK symbols. This conversion can be made using a simple and efficient technique, discribed in the corresponding paragraph.

5.1 SDR, SE-SD, Hybrid SDR-SD simulations over MIMO channels and comparison

In this section, the SDR, the SD and the hybrid SDR-SD algorithm are compared. All the simulation scripts are written in MATLAB. The SDR algorithm is implemented using the general-purpose SeDuMi MATLAB toolbox [5]. SE-SD is implemented as a MATLAB executable (mex) compiled from optimized C code. The hybrid SDR-SD algorithm uses the implementations of SDR and SE-SD, integrated using a MATLAB simulation script.

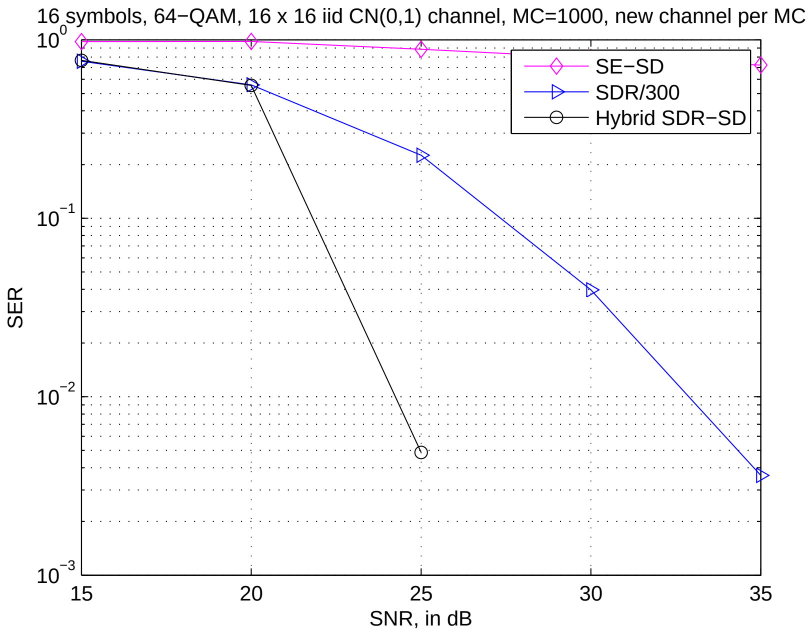

Figure 5.1: SER s SNR for 16x16 MIMO system using 64-QAM

5.1.1 16x16 MIMO system using 64-QAM constellation

In this scenario the number of randomization steps for SDR is set to 300. The initial radius for SE-SD is set to infinity, in order to obtain always the ML solution. Note that the simulation script performs a control in SE-SD's execution time and if it exceeds the value of 300 seconds for a single block detection, SE-SD is being deactivated for the rest of the MC runs for the corresponding SNR value, otherwise the simulations will take too much time. The SER of SE-SD in this case is set to the random choice: $\frac{|\mathcal{FA}|^2-1}{|\mathcal{FA}|^2}+$ until now errors in detection, and the mean execution time to infinity. The results of this simulation are presented in figures 5.1, 5.2, 5.3.

It is obvious that the hybrid SDR-SD algorithm outperforms SDR in terms

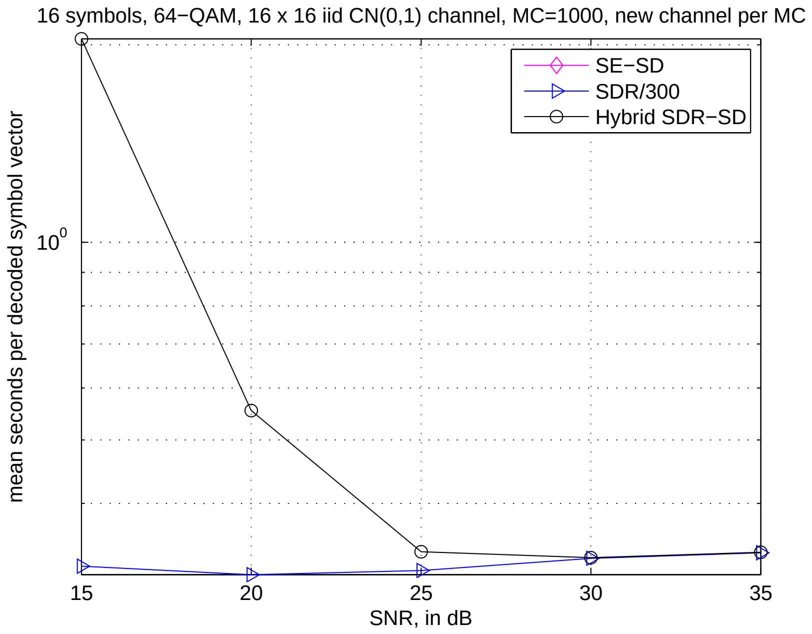

Figure 5.2: Mean execution time vs SNR for 16x16 MIMO system using 64-QAM

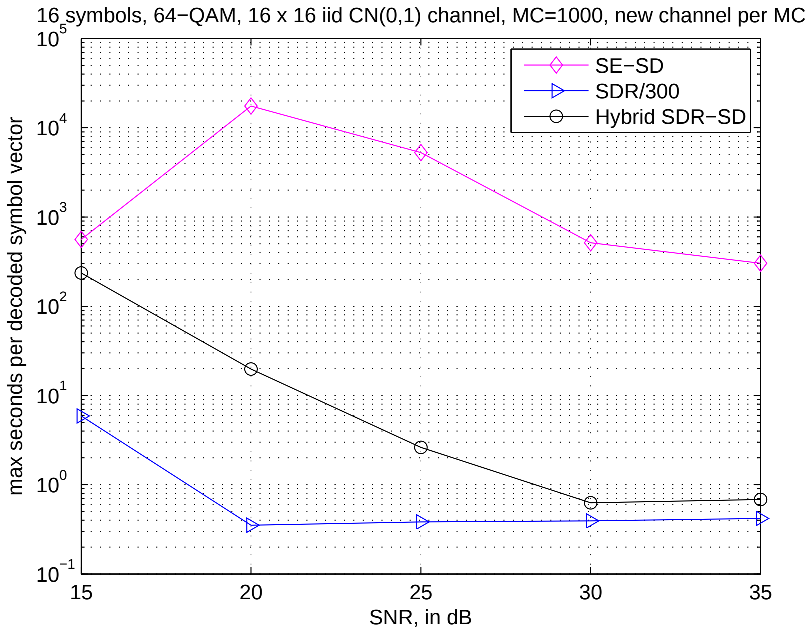

Figure 5.3: Worst case execution time vs SNR for 16x16 MIMO system using 64-QAM

of error performance. SE-SD could not return a solution in reasonable (¡ 300 sec) time at all SNR values. That is why it's SER is not the same with the hybrid SDR-SD's one (without the execution time control it had to be the same, since they both search for the ML solution in an appropriate sphere). The execution times of the hybrid SDR-SD algorithm (mean and as well worst case) are obviously reduced relative to those of SE-SD with infinity initial radius, by using the SDR estimate to initialize the radius. Hybrid SDR-SD's execution times can be analyzed in two parts. The first part contains an approximately constant execution time due to the SDR operation, in order to define SE-SD's radius, and the second part that is the execution of SE-SD with initial radius the one defined by the LS error of the SDR estimate. At low SNR values (15 to 23dB), the SE-SD execution is time consuming in contrast with high SNR's(25 to 35dB) where SE-SD finds the ML solution quickly and the execution time of the hybrid SDR-SD algorithm is almost the same with SDR's one. The key feature of hybrid SDR-SD algorithm in those cases (long symbol vectors and high order constellations) is that SDR can run and return a quasi-ML estimate in a constant low-order polynomial time, and by using it we can "pay" something more (in terms of execution time / complexity) to obtain the exact ML solution by running SE-SD with the SDR estimate LS error as it's initial radius.

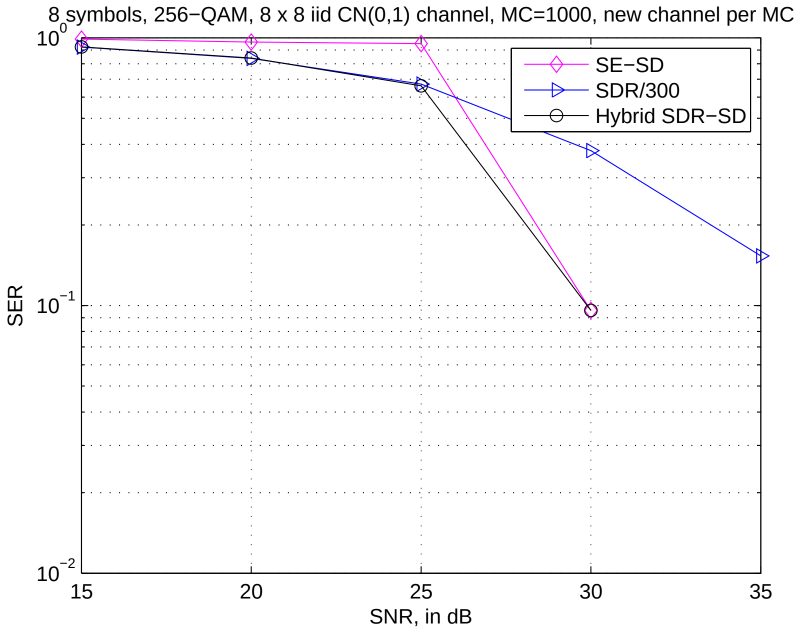

Figure 5.4: SER vs SNR for 8x8 MIMO system using 256-QAM

5.1.2 8x8 MIMO system using 256-QAM constellation

This scenario assumes a MIMO system with 8 transmit and receive antennas using 256-QAM constellation. The results are presented in figures 5.4, 5.5, 5.6.

As it is obvious from figure 5.1.2, SE-SD did not manage to yield the ML solution in reasonable time at low SNR values (15 to 30dB) in contrast with hybrid SDR-SD who did it at low computational cost. SE-SD exhibits the same error performance with hybrid SDR-SD at high SNR values (30 to 35dB) with less average execution time but much more worst case execution time. Hybrid SDR-SD, on the other hand, features a competitive mean and worst-case, execution time performance in addition to ML (optimal) error performance at all noise levels. It is reasonable to mention again that hybrid SDR-SD's execution time is

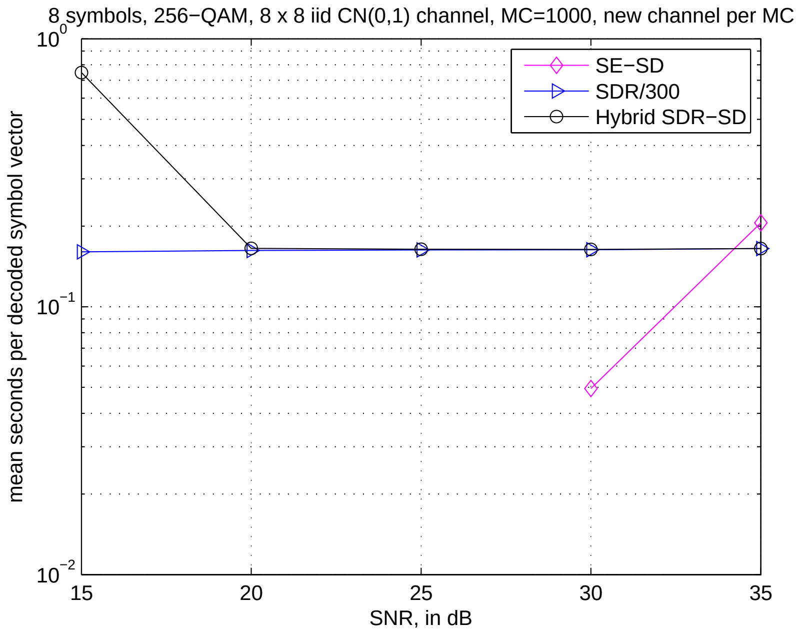

Figure 5.5: Mean execution time vs SNR for 8x8 MIMO system using 256-QAM

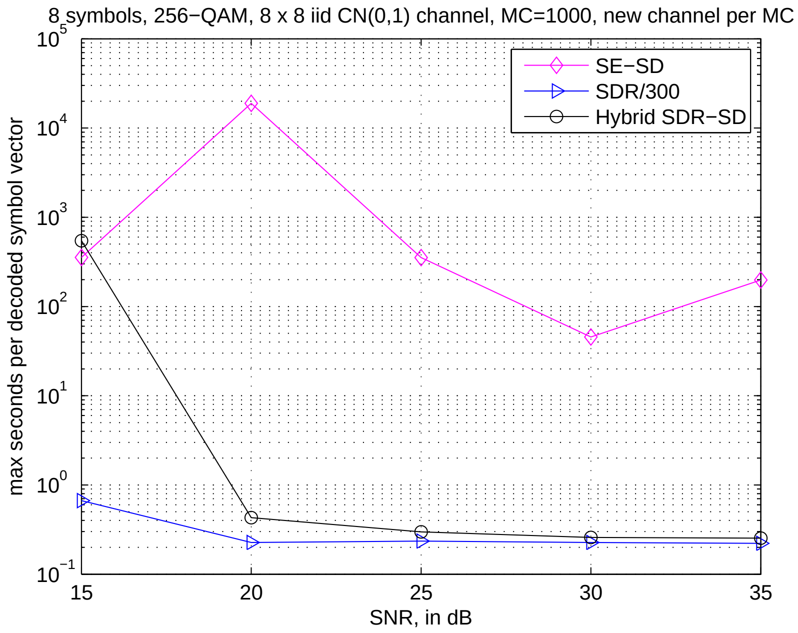

Figure 5.6: Worst case execution time vs SNR for 8x8 MIMO system using 256-QAM

a function (combination) of SDR's and SE-SD's (with SDR radius initialization) execution times. In most scenarios considered here (8x8 MIMO using 256-QAM and for 16x16 MIMO using 64-QAM from 25 to 35dB), the second stage of the algorithm (SE-SD execution) is so fast, that the mean execution time of the hybrid SDR-SD algorithm is almost the same with SDR's one. This means that the SE-SD's radius initialization using SDR, gives very good comlexity reduction for the SE-SD stage, without harming the ML error performance feature of SE-SD.

5.2 Hybrid SDR-SD and fast SDR detector comparison

In this section, the hybrid SDR-SD algorithm is compared with the fast SDR BPSK detector [27]. In order to compare these algorithms using a QAM constellation, the 2-dimensional signal (QAM) has to be transformed to a BPSK one, since fast SDR works only with BPSK signals. This method is introduced in the following section.

5.2.1 QAM to multidimensional BPSK transformation

An x-QAM signal can be written as a linear combination of BPSK signals. Consider a 4-QAM signal. It can be written as

$$\mathbf{s}_{4-QAM} = \begin{bmatrix} 1 & j \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}, \tag{5.1}$$

where $b_1, b_2 \in \{\pm 1\}$ are BPSK singnals. This can be expanded for a 16-QAM constellation as follows.

$$\begin{aligned} \mathbf{s}_{16-QAM} &= \begin{bmatrix} 1 & j \end{bmatrix} \begin{bmatrix} s_4 \\ t_4 \end{bmatrix} = (*), \quad s_4, t_4 \in 4 - PAM \\ (*) &= \begin{bmatrix} 1 & j \end{bmatrix} \begin{bmatrix} 1 & 2 & 0 & 0 \\ 0 & 0 & 1 & 2 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix} = \\ & \underbrace{\begin{bmatrix} 1 & 2 & j & 2j \end{bmatrix}}_{\mathbf{c}^T} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix}, \quad b_i \in BPSK \end{aligned} \tag{5.2}$$

The MIMO detection problem for $N = M$ can be written as

$$\min_{\mathbf{s} \in (16-QAM)^N} ||\mathbf{d} - \mathbf{M}\mathbf{s}||_2^2 \tag{5.3}$$

where $\mathbf{M} \in \mathcal{C}^{N\times N}$ and

$$\mathbf{s} = \underbrace{\begin{bmatrix} \mathbf{c}^T & 0 & \dots & 0 \\ 0 & \mathbf{c}^T & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \mathbf{c}^T \end{bmatrix}}_{\mathbf{A}_{N\times 4N}} \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \vdots \\ \mathbf{b}_N \end{bmatrix}_{4N\times 1} \tag{5.4}$$

So, the equivalent problem can be written as

$$\min_{\mathbf{b}\in\{\pm 1\}^{4N}} ||\mathbf{d} - \underbrace{\mathbf{MA}}_{N\times 4N} \mathbf{b}||_2^2 \tag{5.5}$$

Transformation (5.2) can be expanded for any high-order QAM constellation. The general case can be written as

$$\mathbf{s}_{x-QAM} = \begin{bmatrix} 1 & j \end{bmatrix} \underbrace{\begin{bmatrix} 2^0 & 2^1 & \dots & 2^{i-1} & 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & 2^0 & 2^1 & \dots & 2^{i-1} \end{bmatrix}}_{2\times 2i} \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{2i} \end{bmatrix} \tag{5.6}$$

where i is the column index and is the number of the bits per constellation symbol. For example, a 64-QAM constellation is the product of two 8-PAM constellations (one in each dimension). A 8-PAM symbol “carries” 3 bits. This number (3 in this case) is the value of i.

5.2.2 Simulation Results

The simulation of hybrid SDR-SD and fast SDR detector was made over a MIMO channel with 16 transmitt and 16 receive antennas using 16-QAM constellation. The algorithms are compared for their error performance, their mean and worst case execution time in a noise environment from 15 to 35dB. The implementation of the fast SDR detector (available online, see URL in [27]) is made as a MATLAB executable (mex) compiled from optimized C code, in contrast with hybrid SDR-SD where only the second stage of the algorithm (SE-SD execution) is implemented in optimized C code. As a result, the execution times (mean and worst-case) estimates, are somewhat biased in favor of fast SDR detector. The simulation script is written in MATLAB code, using the simulation setup described earlier (chapter introduction). The results are presented in the following figures.

As it is obvious in figure 5.2.2, hybrid SDR-SD outperforms fast SDR detector (with QAM to multidimensional BPSK transformation) in terms of error performance. Note here that due to the QAM to multidimensional BPSK transformation, the resulting channel matrix is fat an thus the warm-start procedure of [27] must be disabled. This takes away an important advantage of [27], however there was no way around it. Figures 5.2.2 and 5.2.2 show that the execution times of the algorithms, in this scenario, are almost equal, with the hybrid SDR-SD algorithm having less worst case execution time at high SNR values (23 to 35dB). In conclusion, hybrid SDR-SD is better than fast SDR detector in communication

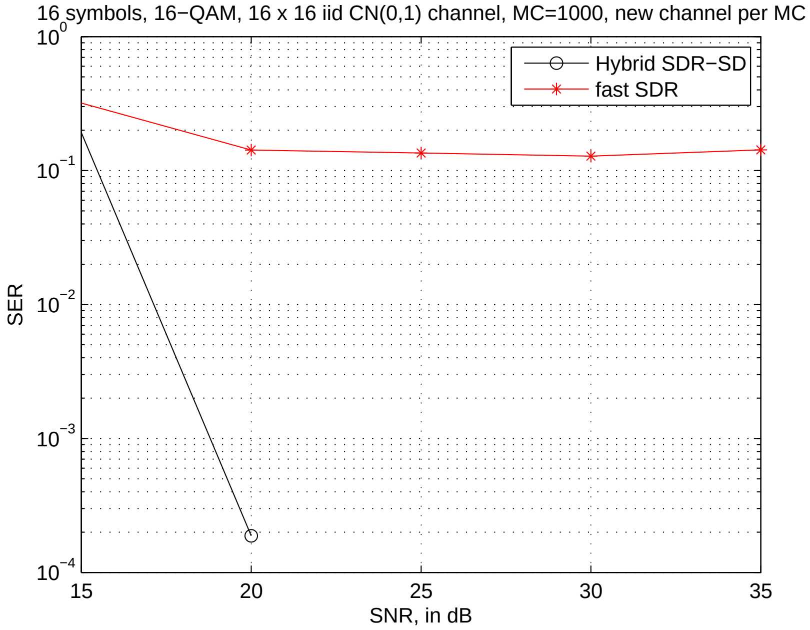

Figure 5.7: SER vs SNR for 16x16 MIMO system using 16-QAM

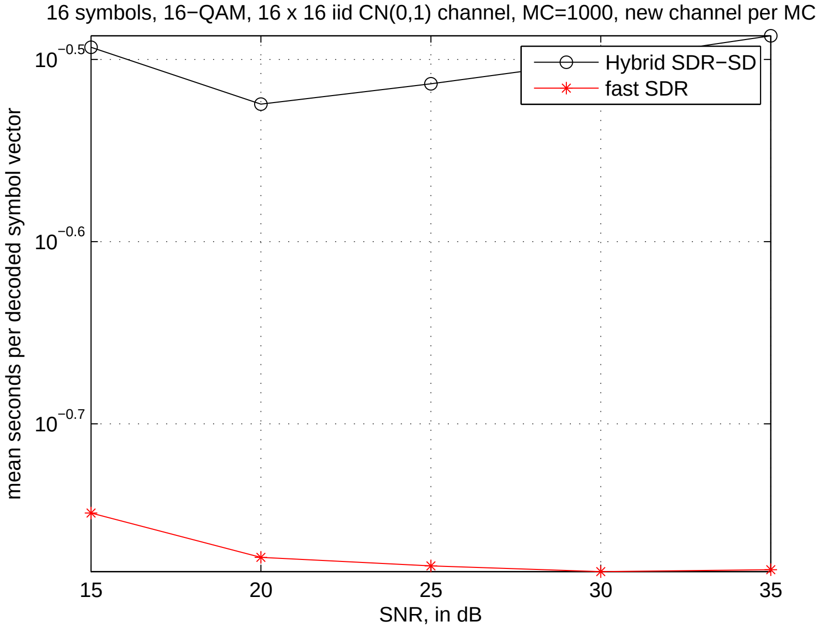

Figure 5.8: Mean execution time vs SNR for 16x16 MIMO system using 16-QAM

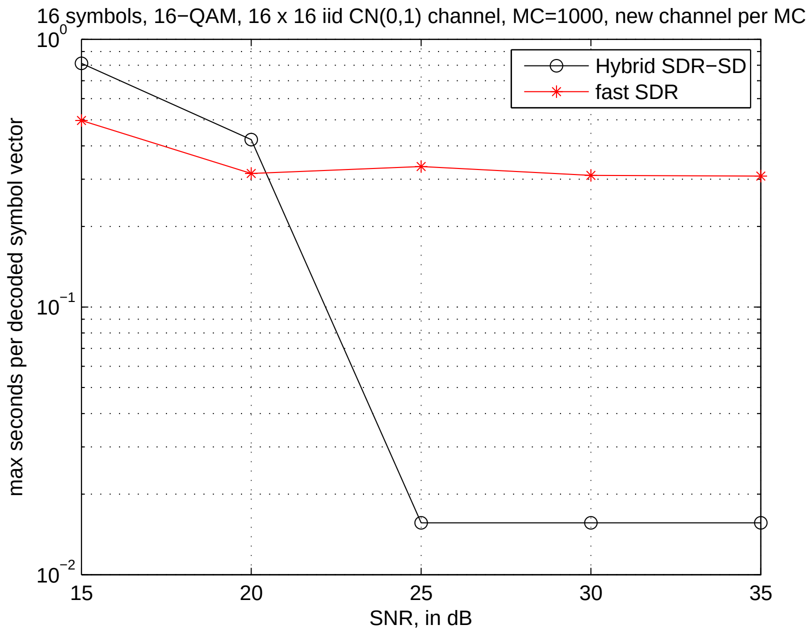

Figure 5.9: Worst case execution time vs SNR for 16x16 MIMO system using 16-QAM

systems that use QAM constellations.

Bibliography

[1] S. Verdu, "Multiuser Detection", *Cambridge University Press*, New York, 1998

[2] S. Boyd, L. Vandenberghe, "Convex Optimization", *Cambridge University Press*, Cambridge, 2004

[3] L.Vandenberghe, S. Boyd, "Semidefinite programming", *SIAM Rev.*, vol. 38, pp. 49-95, 1996

[4] G. B. Giannakis, Z. Liu, X. Ma, and S. Zhou, "Space-Time Coding for Broadband Wireless Communications", *John Wiley & Sons*, Inc., November 2006

[5] J. F. Sturm, "Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones" Optimiz. Methods Softw. vol. 1112, pp. 625653, 1999. [Online]. Available: http://sedumi.mcmaster.ca.

[6] Z. -Q. Luo, W. Yu, "An Introduction to Convex Optimization for Communications and Signal Processing", *IEEE Journal on Selected Areas in Comm.*,

vol. 24, no. 8, August 2006

[7] W.-K. Ma, T.N. Davidson, K.M Wong, Z.-Q. Luo, P.-C. Ching, “Quasi-Maximul-Likelihood Multiuser Detection Using Semi-Definite Relaxation With Application to Synchronous CDMA”, *IEEE Trans. Signal Processing*, vol. 50, no. 4, April 2002

[8] N.D. Sidiropoulos, Z.-Q. Luo, “A Semidefinite Relaxation Approach to MIMO Detection for High-Order QAM Constellations”, *IEEE Sign. Proc. Letters*, vol. 13, no. 9, September 2006

[9] Z. -Q. Luo, X. Luo, and M. Kisialiou, “An efficient quasimaximum likelihood decoder for PSK signals” *in Proc. ICASSP*, 2003

[10] W. K. Ma, P. C. Ching, and Z. Ding, “Semidefinite relaxation based multiuser detection for M-ary PSK multiuser systems”, *IEEE Trans. Signal Process.*, vol. 52, no. 10, pp. 28622872, Oct. 2004

[11] A.Wiesel, Y. Eldar, and S. Shamai, “Semidefinite relaxation for detection of 16-QAM signaling in MIMO channels”, *IEEE Signal Proc. Letters*, vol. 12, no. 9, pp. 653656, Sep. 2005

[12] U. Fincke and M. Pohst, “Improved methods for calculating vectors of short length in a lattice, including a complexity analysis”, *Mathematics of Computation*, vol. 44, pp. 463-471, April 1985

[13] E. Viterbo and J. Boutros, “A universal lattice code decoder for fading channels”, *IEEE Trans. Information Theory*, vol. 45, pp. 1639-1642, July 1999

[14] E. Agrell, T. Eriksson, A. Vardy, and K. Zeger, “Closest point search in lattices”, *IEEE Trans. Inf. Theory*, vol. 48, no. 8, pp. 22012214, Aug. 2002

[15] H. Vikalo, “Sphere Decoding Algorithms for Digital Communications”, *Ph.D Dissertation*, Stanford University, 2003

[16] B. Hassibi and H. Vikalo, “On Sphere Decoding Algorithm. I. Expected Complexity”, *IEEE Transactions on Signal Processing*, vol. 53, no. 8, August 2005, pp. 2806 - 2818

[17] B. Hassibi and H. Vikalo, “On Sphere Decoding Algorithm. II. General- izations, Second-order Statistics, and Applications to Communications”, *IEEE Transactions on Signal Processing*, vol. 53, no. 8, August 2005, pp. 2819 - 2834

[18] A. Chan and I. Lee, “A new reduced-complexity sphere decoder for multiple antenna systems”, *in Proc. ICC*, New York, Apr. 28May 2 2002, vol. 1, pp. 460464

[19] R. Wang and G. B. Giannakis, “Approaching MIMO capacity with reduced-complexity soft sphere-decoding”, *in Proc. WCNC*, Atlanta, GA, Mar. 2125, 2004

[20] H. Vikalo, B. Hassibi, and T. Kailath. “Iterative Decoding for MIMO Channels Via Modified Sphere Decoding”, *IEEE Trans. on Wireless Communications*, vol. 3, no. 6, November 2004

[21] L. Brunel J. Boutros, “Lattice Decoding for Joint Detection in Direct- Sequence CDMA Systems”, *IEEE Trans. Inform. Theory*, vol. 49, no. 4,pp. 1030-1037, April 2003

[22] W. Zhao and G. B. Giannakis, “Sphere decoding algorithms with improved radius search”, *IEEE Trans. on Communications*, vol.53, no.7, July 2005

[23] M. Stojnic, H. Vikalo, and B. Hassibi, “A Branch and Bound Approach to Speed Up the Sphere Decoder”, *in ICASSP*, 2005

[24] M. Stojnic, H. Vikalo, and B. Hassibi, “Further Results on Speeding Up the Sphere Decoder”, *in ICASSP*, 2006

[25] G. Latsoudas, N. D. Sidiropoulos, “A Hybrid Probabilistic Data Association-Sphere Decoding Detector for Multiple-InputMultiple-Output Systems”, *IEEE Sign. Proc. Letters*, vol.12, no.4, April 2005

[26] Joakim Jaldén, Björn Ottersten, “An Exponential Lower Boung on the Expected Complexity of Sphere Decoding”, *in ICASSP*, 2004

[27] M. Kisialiou, X.-D. Luo, Z.-Q. Luo, “Quasi-Maximum-Likelihood Detection Based on Semidefinite Relaxation: Analysis and Implementation”, Sub-

mitted to *IEEE Trans. on Sign. Proc.*, December 2006. Also available at : http://www.ece.umn.edu/users/luozq/software/sw_about.html

[28] M. Grotschel, L. Lovasz, and A. Schriver, "Geometric Algorithms and Combinatorial Optimization", *Springer Verlag*, 2nd ed., 1993

[29] G. J. Foschini, "Layered space-time architecture for wireless communication in a fading environment when using multi-element antennas", *Bell Labs. Tech. J.*, vol. 1, no. 2, pp. 4159, 1996

[30] B. Hassibi, "An efficient square-root algorithm for BLAST", submitted to *IEEE Trans. Sig. Proc., 2000.* Download available at http://mars.bell-labs.com