Comparison and Speedup of Near-Optimal Decoding Algorithms for MIMO Systems

DIPLOMA THESIS

BY

Giorgos Latsoudas

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

Technical University of Crete

Advisor: Professor Sidiropoulos Nikolaos

Co-advisor: Professor Paterakis Michalis

Co-advisor: Associate Professor Liavas Athanasios

June 2004

Contents

Chapter 1

Introduction

This thesis is concerned with the comparison and further development of certain detection algorithms which can be applied for multiuser detection in DS-CDMA systems, and for low complexity quasi-optimal detection in Multiple-Input Multiple-Output communication systems.

1.1 MIMO Observation Model

We consider an M-dimensional data vector $\mathbf{s}$, consisting of real-valued ± 1 elements, transmitted through a linear time-invariant $N \times M$ channel $\mathbf{H}$. The received $N \times 1$ vector is corrupted by a Gaussian noise vector with zero mean and covariance matrix $\sigma^2 I$, where I denotes the $N \times N$ identity matrix. Thus at an instant of time the received signal is given by

$$\begin{gathered}\mathbf{y} = \mathbf{H}\mathbf{s} + \mathbf{n} \\ s_i \in \{1, -1\}\ ,\ i = 1, ..., M\end{gathered} \tag{1.1}$$

where $\mathbf{H}$ is the channel mixing matrix and $\mathbf{n}$ is the white Gaussian noise vector. We assume that the mixing matrix is known to the receiver but not to the transmitter. Model (1.1) appears in the context of various modern communication systems, including muti-user spread spectrum systems like synchronous CDMA [1] and multi-antenna systems like V-BLAST [2]. The case in which the symbols of $\mathbf{s}$ are taken from a complex constellation and transmitted through a complex channel with white Gaussian complex noise vector, can be easily reduced to a real model, as in (1.1). Our purpose at the receiver is to accurately reproduce the transmitted signal.

1.2 ML Detection

The Maximum Likelihood (ML) detector chooses the value of the vector $\mathbf{s}$ which maximizes the conditional probability that $\mathbf{y}$ is received given that $\mathbf{s}$ has been sent

$$\max_{\mathbf{s}} p_{\mathbf{y}|\mathbf{s}}(\mathbf{y}|\mathbf{s}) \tag{1.2}$$

The maximization in (1.2) is performed over the entire M-dimensional symbol space, i.e., over all the combinations of the possible values for every s_i. In our case, where the noise is modeled as additive white Gaussian, the conditional probability of observing $\mathbf{y}$ given that $\mathbf{s}$ has been transmitted, is

$$p_{\mathbf{y}|\mathbf{s}} = \frac{1}{2\pi} e^{-\frac{||\mathbf{y}-\mathbf{Hs}||^2}{2\sigma^2}} \tag{1.3}$$

Therefore the maximization of the conditional probability (1.2) is reduced to the combinatorial optimization problem

$$\min_{\mathbf{s}} ||\mathbf{y} - \mathbf{Hs}||^2 \tag{1.4}$$

Where the components of s are -1 or 1. More formally, problem (1.4) can be stated as

Given the vector $\mathbf{y} \in \mathbb{R}^N$, *and the matrix* $\mathbf{H} \in \mathbb{R}^{N \times M}$, *find the column vector* $\mathbf{s} \in \{1, -1\}^M$ *which minimizes*

$$||\mathbf{y} - \mathbf{Hs}||_2 \tag{1.5}$$

The above problem is the binary case of the well known *Integer Least Squares* (ILS) problem [3]. Problem (1.5) can be rewritten as

$$\begin{aligned} \arg\min_{\mathbf{s}} ||\mathbf{y} - \mathbf{Hs}||_2 &= \arg\min_{\mathbf{s}} (\mathbf{y} - \mathbf{Hs})^T (\mathbf{y} - \mathbf{Hs}) \\ &= \arg\min_{\mathbf{s}} (-\mathbf{s^T H^T y} - (\mathbf{s^T H^T y})^T + \mathbf{s^T H^T H s}) \\ &= \arg\min_{\mathbf{s}} (\mathbf{s^T H^T H s} - 2\mathbf{s^T H^T y}) \end{aligned} \tag{1.6}$$

As we have said our purpose at the receiver is to detect as correctly as possible the components of $\mathbf{s}$. The ML detector minimizes $||\mathbf{y} - \mathbf{Hs}||_2$ and exhibits optimal performance (in terms of error probability) among all detectors, provided that all symbol vectors, $\mathbf{s}$ are equiprobable. Unfortunately, it is known that its computational cost is exponential in M. Specifically, problem (1.5) has been proven to be NP-hard [1]. For this reason a number of alternative algorithms have been developed in order to achieve near-optimal performance with low computational cost. Those algorithms include the *Sphere Decoder (SD), Semidefinite Relaxation (SDR) and Probabilistic Data Association (PDA).* We will first review these three algorithms. Then, we will propose a combination of SD and PDA, and show that it attains a better performance-complexity trade-off than SD, PDA and SDR.

1.3 Measures of Performance

For binary symbols like in (1.1), a meaningfull statistical measure of the effectiveness of a detector is the Bit Error Rate (BER), which expresses the probability that a transmitted bit has been mistaken for its complement.

Moreover, a detector should be computationally efficient in order to be implementable in practical communication systems. For this reason, as a measure of the computational cost of the various detection algorithms, we use the number of Floating Point Operations (FLOPS). This kind of arithmetical operations is the most expensive (regarding the time which is needed to be completed) in the hardware components which are used in modern communication systems.

Finally a detection algorithm must be quite simple in order to be easily implementable in practical communication systems.

1.4 Outline of the Thesis

In this thesis, we implement various detection algorithms for the model in (1.1), and compare their performance in the context of several pertinent communication systems.

In chapter 2 we explain in detail three near optimal decoding algorithms. In the first section we study Semidefinite Relaxation. We start with a description of Semidefinite relaxation for the solution of the Boolean Quadratic Programming (QP) problem. After that we make the link between Boolean QP and ML decoding problems. We close this section with a description of a randomization procedure for converting the relaxed problem solution to a proper solution for the original ML decoding problem. The second section of this chapter is devoted to a survey of the Sphere Decoding (Fincke-Pohst) algorithm. We describe the stages of this algorithm and we study in brief its computational complexity. In the third section of this chapter we give a detailed description of the Probabilistic Data Association (PDA) algorithm, and of certain modifications which improve its computational complexity. We next derive a new algorithm which is a combination of the Sphere decoder and PDA algorithms. This algorithm consists of a PDA first stage which decides the values of particular symbols (bits) which can be reliably decoded, followed by a properly modified sphere decoder which uses the decisions of the PDA stage to give the final decoded vector of symbols. We finally make a coarse estimation of its computational complexity.

In chapter 3 we describe specific multiple input multiple output communication system models and we give the simulation results for the above four near-optimal algorithms on these scenarios. The first section is a study of the synchronous CDMA system model and a presentation of the results of simulation examples in which we compare the BER performance and computational complexity of the first three algorithms. In the second section we describe the multiple antenna model. The simulation examples which are presented in this section compare the first three algorithms with the proposed algorithm and provide a qualitative description of the pertinent algorithm parameters.

In the final chapter we give some conclusions regarding the comparison of the algorithms in the scenarios considered. We also propose some ideas for future work in the field of optimum detection.

Chapter 2

Computationally Efficient Decoding Algorithms for MIMO systems

In this chapter we describe various algorithms which can be used for detection in MIMO communication systems. Semidefinite Relaxation, Sphere Decoding and Probabilistic Data Association are explained in detail. We also propose an algorithm which combines Probabilistic Data Association and Sphere Decoding.

2.1 Semidefinite Relaxation

Semidefinite Relaxation has been proposed for synchronous CDMA Multiuser detection in [4]. The ML detection problem (1.4) is an instance of the general

Boolean Quadratic Programming (QP) problem. In this section, following the algorithm description in [4], we present the application of SD relaxation for the QP problem and see how QP and ML detection problems are connected.

The form of the Boolean QP problem is:

$$\max_{\mathbf{x}\in\{-1,+1\}^n} \mathbf{x}^T\mathbf{Q}\mathbf{x} \tag{2.1}$$

Where $\mathbf{Q}$ is an $n \times n$ symmetric matrix. For the description of SD relaxation, we must reformulate the Boolean QP problem. Since $\mathbf{x}^T\mathbf{Q}\mathbf{x} = Trace(\mathbf{x}\mathbf{x}^T\mathbf{Q})$, problem (2.1) can be *equivalently* written as

$$\begin{aligned} &\max \ Trace(\mathbf{X}\mathbf{Q}) \\ s.t. \quad &\mathbf{X} = \mathbf{x}\mathbf{x}^T, \mathbf{x} \in \mathbb{R}^n \\ &X_{ii} = 1, i = 1, \ldots, n \end{aligned} \tag{2.2}$$

The constraint $\mathbf{X} = \mathbf{x}\mathbf{x}^T$ is equivalent to saying that $\mathbf{X}$ is symmetric, positive semidefinite and of rank 1. Problem (2.2) is a non-convex optimization problem, because of the rank-1 constraint. In order to convert this to a convex optimization problem we drop the rank-1 constraint from (2.2) and this yields the following relaxed problem

$$\begin{aligned} &\max \ Trace(\mathbf{X}\mathbf{Q}) \\ s.t. \quad &\mathbf{X} \succeq 0 \\ &X_{ii} = 1, \quad i = 1, ..., n \end{aligned} \tag{2.3}$$

Where $\mathbf{X} \succeq 0$ signifies that $\mathbf{X}$ is symmetric and Positive Semidefinite. Problem (2.3) is a Semidefinite Programming (SDP) problem [5] and so (2.3) is called a Semidefinite Relaxation of (2.2). It is important to notice that (2.3) does not exhibit the disadvantage of local maxima because it is a convex optimization problem [4]. For the SDP problem (2.3) an efficient optimization algorithm based on interior-point methods has been developed [6], [7]. This algorithm solves the problem (2.3) in at most $\mathcal{O}(n^{3.5})$ operations for a pre-specified accuracy.

Note that by dropping the rank-1 constraint of the original problem, and employing only the symmetric positive semidefinite constraint, Semidefinite Relaxation causes an increase in problem dimensionality (from n to n^2) . Generally, it is possible that $\mathbf{X}^* \neq \mathbf{x}^*\mathbf{x}^{*T}$, where $\mathbf{X}^*$ stands for the solution of the relaxed problem and $\mathbf{x}^*$ represents the solution of the original problem (2.1) ($-\mathbf{x}^*$ is also a solution of (2.1)). Hence, some special technique is required for converting the solution of the relaxed problem to a proper solution of the original problem.

A randomization method has been proposed in [8], [9] and revised in [4], for the conversion of the SD relaxation solution to an approximate Boolean QP solution. Here, in the implementation of the SDR-ML detector, we use the above method.

The Boolean QP problem can be rewritten as:

$$\max_{{x_i}^2=1, i\in\{1,..,n\}} \sum_{i=1}^{n}\sum_{j=1}^{n} x_i x_j Q_{ij} \tag{2.4}$$

.

For the SD relaxation problem (2.3), we consider the Cholesky decomposition $\mathbf{V}^T\mathbf{V} = \mathbf{X}$, $\mathbf{V} = [\mathbf{v}_1, ..., \mathbf{v}_n]$. By replacing $\mathbf{X} = \mathbf{V}^T\mathbf{V}$ problem (2.3) can be equivalently written as

$$\max_{||\mathbf{v}_i||_2=1, i\in\{1,..,n\}} \sum_{i=1}^{n}\sum_{j=1}^{n} \mathbf{v}_i^T \mathbf{v}_j Q_{ij}. \tag{2.5}$$

Let $\mathbf{V}^* = [\ \mathbf{v}_1^*, ..., \mathbf{v}_n^*\ ]$ be the Cholesky decomposition of $\mathbf{X}^*$. Thus $\mathbf{V}^*$ is the solution of problem (2.5). Notice that since the problem's dimensionality has been increased, the scalar product $x_i x_j$ in (2.4) is relaxed to the inner product $\mathbf{v}_i \mathbf{v}_j$ in (2.5). Hence, it is possible to approximate x_i^* using $\mathbf{v}_i^*$. A randomization method is used for this reason. A description of its steps follows.

1. Generate an $n \times 1$ random vector $\mathbf{u}$ with elements uniformly distributed in $[-1, 1]$.

2. For $i = 1, ..., n$, if $\mathbf{v}_i^{*T}\mathbf{u} \geq 0$ set $x_i = 1$, otherwise set $x_i = -1$.

In order to get a better approximation, the randomization is iterated for a number of times M_{rand} [4], and the randomization result which attains the largest objective value is chosen as the preferable approximate solution. In many cases, obtaining a good approximation requires a relatively small number of iterations (In synchronous CDMA multiuser detection $M_{rand} = 10$ to 20 [4]). The computational complexity of the above process is $\mathcal{O}(n^2 M_{rand})$.

The final step for the application of SD relaxation algorithm to the the ML detection problem is to reformulate the ML detection problem in (1.4) in order to be in the same form as the Boolean QP problem (2.1). For this reason, let $c \in \{-1, 1\}$ be a scalar with value -1 or 1. Noticing that $c\mathbf{s} \in \{-1, 1\}^M$ for any

$\mathbf{s} \in \{-1, 1\}^M$, the ML detection problem can be reformulated as:

$$\max_{\mathbf{s}\in\{-1,1\}^M} J(\mathbf{s}) \equiv \max_{\mathbf{s}\in\{-1,1\}^M, c\in\{-1,1\}} J(c\mathbf{s}) \tag{2.6a}$$

$$= \max_{\mathbf{s}\in\{-1,1\}^M, c\in\{-1,1\}} c\mathbf{s}^T\mathbf{H}^T y + (c\mathbf{s}^T\mathbf{H}^T y)^T - \mathbf{s}^T\mathbf{H}^T\mathbf{H}\mathbf{s} \tag{2.6b}$$

$$= \max_{\mathbf{s}\in\{-1,1\}^M, c\in\{-1,1\}} \begin{bmatrix} \mathbf{s}^T & c \end{bmatrix} \begin{bmatrix} -\mathbf{H}^T\mathbf{H} & \mathbf{H}^T\mathbf{y} \\ (\mathbf{H}^T\mathbf{y})^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{s} \\ c \end{bmatrix} \tag{2.6c}$$

It is obvious that (2.6c) is an instance of the Boolean QP problem in (2.1), with $n = M + 1$, $\mathbf{x} = [\mathbf{s}^T \quad c]^T$, and

$$\mathbf{Q} = \begin{bmatrix} -\mathbf{H}^T\mathbf{H} & \mathbf{H}^T\mathbf{y} \\ (\mathbf{H}^T\mathbf{y})^T & 0 \end{bmatrix} \tag{2.7}$$

Denote the solution of (2.6c) as $(\mathbf{s}^*, c^*)$ and the ML estimate as $\hat{\mathbf{s}}_{ml}$. $\hat{\mathbf{s}}_{ml}$ achieves the same maximum objective value with $(\mathbf{s}^*, c^*)$ in (2.6a). Hence we can write

$$\hat{\mathbf{s}}_{ml} = c^*\mathbf{s}^*. \tag{2.8}$$

We see that for solving (2.6c) we can use the SD relaxation algorithm described above, and then we can use (2.8) to find an approximate ML estimate. The computational complexity of the SD relaxation algorithm for the ML detection problem (1.4) is $\mathcal{O}((M+1)^{3.5}) \simeq \mathcal{O}(M^{3.5})$, assuming that the computational cost of the randomization procedure is a negligible part of the overall cost.

Table 2.1: The Semidefinite Relaxation Algorithm

1. Set $n = M + 1$ and
   $\mathbf{Q} = \begin{bmatrix} -\mathbf{H}^T\mathbf{H} & \mathbf{H}^T\mathbf{y} \\ (\mathbf{H}^T\mathbf{y})^T & 0 \end{bmatrix}$
2. Solve the semidefinite programming problem

$$\mathbf{X}^* = \arg \max_{\mathbf{X}\succeq 0,\; X_{ii}=1\;\forall i} Trace(\mathbf{XQ})$$

3. Compute the Cholesky factorization $\mathbf{X}^* = \mathbf{V}^{*T}\mathbf{V}^*$.
4. Randomization procedure:

   Let M_{rand} be the number of randomizations

   for $i = 1, ...M_{rand}$

   - Randomly generate a normally distributed vector $\mathbf{u}_i$. The i-th candidate solution is computed by $\tilde{\mathbf{x}}_\mathbf{i} = sign(\frac{\mathbf{V}^{*T}\mathbf{u}_\mathbf{i}}{||\mathbf{u}_\mathbf{i}||})$

   end

   Compute

$$j = \arg \max_{i=1,\ldots,M_{rand}} \tilde{\mathbf{x}}_\mathbf{i}^\mathbf{T}\mathbf{Q}\tilde{\mathbf{x}}_\mathbf{i}.$$

   $\hat{\mathbf{x}} = \tilde{\mathbf{x}}_\mathbf{j}$ is the approximate solution for $\mathbf{x}^*$

5. Take $\hat{\mathbf{s}}_{SDR} = \hat{x}_{M+1}[\hat{x}_1, ..., \hat{x}_M]$ as the approximate ML detection problem solution

2.2 Sphere Decoder

The Sphere Decoding (SD) Algorithm, first introduced in [10], has been applied to lattice decoding [11], and multiple antenna systems [12], where it is used to compute the quasi-Maximum Likelihood symbol-vector estimate with a moderate computational complexity. It has also been proposed for the Multiuser Detection problem in the Synchronous CDMA system [13]. In this section we give a detailed description of the SD steps and we briefly summarize the results of particular studies on its computational complexity.

2.2.1 ML Decoding Using the Sphere Decoder

The SD algorithm is useful for the detection of transmit vectors consisting of integer symbols. In the case of system (1.1), in order to convert the transmit vector (comprised of symbols with values -1 or +1) to a vector with 1 or 0 elements, we add to both sides the quantity $\mathbf{He}$, where $\mathbf{e}$ is an $M \times 1$ vector with all its components 1, and we divide the result by 2. Hence we obtain a system with 0 or 1 transmit elements and the SD can be straightly applied. The noise variance of the new system is $\frac{\sigma^2}{4}$. For simplicity we keep the same symbols for the vectors, matrices and the noise variance of the new system, as they are in the original system. The maximum likelihood solution for the above problem as we have seen is

$$\hat{\mathbf{s}}_{ml} = \arg \min_{s \in L^M} ||\mathbf{y} - \mathbf{Hs}||_2 \tag{2.9}$$

where L is the set $\{0, 1\}$. The ML solution s_{ml} can be approximated by examining only the possible values of $\mathbf{s}$ for which $||\mathbf{y}-\mathbf{Hs}||_2$ is small, and rejecting the values for which the above metric is large.

We observe that the distance $||\mathbf{y} - \mathbf{Hs}||_2$ can be rewritten as

$$||\mathbf{y} - \mathbf{Hs}||_2 = (\mathbf{s} - \hat{\mathbf{s}})^T\mathbf{H}^T\mathbf{H}(\mathbf{s} - \hat{\mathbf{s}}) + \mathbf{y}^T(\mathbf{I} - \mathbf{H}(\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T)\mathbf{y} \tag{2.10}$$

where $\hat{\mathbf{s}} = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{y}$ is the unconstrained least square solution of $\mathbf{s}$ (the decorrelator's output). Hence (2.9) becomes

$$\hat{\mathbf{s}}_{ml} = \arg\min_{\mathbf{s}\in L^M} ||\mathbf{y} - \mathbf{Hs}||_2 = \arg\min_{\mathbf{s}\in L^M} (\mathbf{s} - \hat{\mathbf{s}})^T\mathbf{H}^T\mathbf{H}(\mathbf{s} - \hat{\mathbf{s}}) \tag{2.11}$$

We denote $\mathbf{w} = \mathbf{Hs}$, $\boldsymbol{\rho} = \mathbf{s} - \hat{\mathbf{s}}$ and $\Omega = \{\mathbf{w} = \mathbf{Hs}, \mathbf{s} \in L^M\}$. Sphere Decoder solves the equivalent problem

$$\min_{\mathbf{w}\in\Omega} ||\mathbf{y} - \mathbf{w}||_2 = \min_{\mathbf{w}\in\Omega} (\mathbf{y} - \mathbf{w})^T(\mathbf{y} - \mathbf{w}) = \min_{\mathbf{s}\in L^M} (\mathbf{s} - \hat{\mathbf{s}})^T\mathbf{H}^T\mathbf{H}(\mathbf{s} - \hat{\mathbf{s}}) \tag{2.12}$$

with the search sphere centered at the received point in the coordinate system defined by $\mathbf{w}$. This sphere is transformed into an ellipsoid centered at the origin of the axes if we consider the coordinate system defined by the translated vector $\boldsymbol{\rho}$. It is clear from (2.12) that the sphere decoder may be used to obtain the ML estimate $\hat{\mathbf{s}}_{\mathbf{ml}}$. We have seen in the first chapter that solving (2.12) requires exhaustive search over all the possible values of $\mathbf{s}$. The possible values of $\mathbf{s}$ can be represented as points in an n-dimensional lattice. The Sphere Decoder avoids the exhaustive search by including in the search only the points which lie inside the search sphere, i.e. satisfy the inequality

$$(\mathbf{s}-\hat{\mathbf{s}})^T\mathbf{H}^T\mathbf{H}(\mathbf{s}-\hat{\mathbf{s}}) \le C \tag{2.13}$$

Consider the Gram matrix $\mathbf{G} = \mathbf{H}^T\mathbf{H}$ and compute its Cholesky factorization $\mathbf{U}^T\mathbf{U} = \mathbf{G}$. $\mathbf{U}$ is an upper triangular matrix with positive diagonal elements. Let, also, u_{ij} denote the component of $\mathbf{U}$ located at its i-th row and j-th column. Then (2.13) can be reformulated as

$$\begin{aligned}&(\mathbf{s}-\hat{\mathbf{s}})^T\mathbf{U}^T\mathbf{U}(\mathbf{s}-\hat{\mathbf{s}})\\ &=\sum_{i=1}^{M} u_{ii}^2\left[s_i-\hat{s}_i+\sum_{j=i+1}^{M}\frac{u_{ij}}{u_{ii}}(s_j-\hat{s}_j)\right]^2 \le C.\end{aligned} \tag{2.14}$$

Every term of the above sum is greater than or equal to zero. The Sphere Decoder computes bounds on each component of $\mathbf{s}$, s_n $n = M, ..., 1$.

The algorithm starts with $i = M$, and excluding from the sum in (2.14) the terms $i = 1, ..., M-1$, we have

$$u_{MM}^2(s_M-\hat{s}_M)^2 \le C$$

and since every component of $\mathbf{s}$ belongs to an integer signal constellation, we obtain

$$\lceil \hat{s}_M - \frac{\sqrt{C}}{u_{MM}} \rceil \le s_M \le \lfloor \hat{s}_M + \frac{\sqrt{C}}{u_{MM}} \rfloor. \tag{2.15}$$

Where the function$\lceil x \rceil$ is the *ceiling* function which finds the smallest integer $c \ge x$, and $\lfloor . \rfloor$ is the *floor* function which finds the largest integer $c \le x$. Then, the SD chooses a candidate value for s_M which falls inside the bounds in (2.15). Continuing, in order for the algorithm to compute the bounds for s_{M-1}, it takes into account only the effect of the chosen value for s_M and excludes from the sum in (2.14) the terms $i = 1, ..., M-2$, leading to the inequality

$$u^2_{M-1,M-1}\left[s_{M-1} - \hat{s}_{M-1} + \frac{u_{M-1,M}}{U_{MM}}(s_M - \hat{s}_M)\right]^2 + u^2_{MM}(s_M - \hat{s}_M)^2 \leq C \tag{2.16}$$

Thus the corresponding range is

$$\left\lceil \hat{s}_{M-1} - \frac{\sqrt{C - u^2_{MM}(s_M - \hat{s}_M)^2}}{u_{M-1,M-1}} - \frac{u_{M-1,M}}{u_{MM}}(s_M - \hat{s}_M)\right\rceil \leq s_{M-1} \leq \left\lfloor \hat{s}_{M-1} + \frac{\sqrt{C - u^2_{MM}(s_M - \hat{s}_M)^2}}{u_{M-1,M-1}} - \frac{u_{M-1,M}}{u_{MM}}(s_M - \hat{s}_M)\right\rfloor \tag{2.17}$$

The SD selects a candidate value for s_{M-1} within the upper and lower bounds in (2.17), and working backwards computes successively the bounds for $s_{M-2}, ..., s_1$. For the ith component of $\mathbf{s}$ the range is

$$\left\lceil -\frac{1}{u_{ii}}\sqrt{C - \sum_{l=i+1}^{M} u^2_{ll}\left(s_l - \hat{s}_l + \sum_{j=l+1}^{M}\frac{u_{lj}}{u_{ll}}(s_j - \hat{s}_j)\right)^2} + \hat{s}_i - \sum_{j=i+1}^{M}\frac{u_{ij}}{u_{ii}}(s_j - \hat{s}_j)\right\rceil \leq s_i \leq \left\lfloor \frac{1}{u_{ii}}\sqrt{C - \sum_{l=i+1}^{M} u^2_{ll}\left(s_l - \hat{s}_l + \sum_{j=l+1}^{M}\frac{u_{lj}}{u_{ll}}(s_j - \hat{s}_j)\right)^2} + \hat{s}_i - \sum_{j=i+1}^{M}\frac{u_{ij}}{u_{ii}}(s_j - \hat{s}_j)\right\rfloor \tag{2.18}$$

The following auxiliary variables can be used for computing the bound recursively

$$
\begin{aligned}
S_i &= S_i(s_{i+1} - \hat{s}_{i+1}, \ldots, s_M - \hat{s}_M) = \hat{s}_i - \sum_{l=i+1}^{M} \frac{u_{il}}{u_{ii}}(s_l - \hat{s}_l) \\
T_{i-1} &= T_{i-1}(s_i - \hat{s}_i, \ldots, s_M - \hat{s}_M) = C - \sum_{l=1}^{M} u_{ll}^2 \left(s_l - \hat{s}_l + \sum_{j=l+1}^{M} \frac{u_{lj}}{u_{ll}}(s_j - \hat{s}_j) \right)^2 \\
&= T_i - u_{ii}^2 (S_i - s_i)^2.
\end{aligned}
\tag{2.19}
$$

During its execution, the SD may find that no possible value for some component s_m lies within its corresponding range. In this case the algorithm goes back to the previous component s_{m+1} and changes its value, choosing another constellation point between the upper and lower bounds for s_{m+1}. If there is no other possible value for s_{m+1}, the SD changes the value of s_{m+2} and the algorithm proceeds in the same way until a new candidate solution is found (or until it is detected that no point lies inside the sphere).

If the algorithm reaches s_1 and selects a candidate value for it within its corresponding range, we have a new candidate solution for the vector $\mathbf{s}$. In the case that the square distance of this candidate solution from the center of the sphere is smaller than the search radius, the algorithm updates its searching radius by this distance which is given by

$$
\hat{d}^2 = C - T_1 + {u_{ii}}^2 (S_1 - s_1)^2 \tag{2.20}
$$

Then the algorithm repeats the search but this time with this smaller radius in order to find any closer candidate point. Therefore, the final solution of the SD is the point inside the sphere having the smallest distance from the center.

In the case, finally, that no candidate value is found inside the whole search sphere an erasure on the detected vector is declared or the algorithm starts over again with larger initial radius. A summary of the sphere decoder steps is given in Table (2.2).

Table 2.2: The Sphere Decoder algorithm

Input $C, \hat{\mathbf{s}}, \mathbf{G} = \mathbf{H}^T\mathbf{H}, \mathbf{U} = chol(\mathbf{G}), q_{i,i} = u_{ii}^2, qi, j = \frac{u_{ij}}{u_{ii}}$

1. Set $\hat{d}^2 = C,\ T_M = C,\ S_k = \hat{s}_k,\ \ k = 1, \dots M$

2. Set $i = M$

3. $L_i = \lfloor \sqrt{T_i/u_{ii}^2} + S_i \rfloor$, $s_i = \lceil S_i - \sqrt{T_i/u_{ii}^2} \rceil - 1$

4. $s_i = s_i + 1$

   if $s_i > L_i$ goto 5

   else goto 6

5. if $i = M$ Output $\hat{\mathbf{x}}, \hat{d}^2$

   else i=i+1 goto 4

6. if $i > 1$

   $\xi_i = s_i - \hat{s}_i$, $T_{i-1} = T_i - u_{ii}^2(S_i - s_i)^2$

   $S_{i-1} = \hat{s}_{i-1} - \sum_{j=i}^{M} q_{i-1,j}\xi_j$

   $i = i + 1$ goto 3

   else set $\hat{d}^2 = T_M - T_1 + q_{1,1}(S_1 - s_1)$ goto 7

7. if $(\hat{d}^2 < d^2)$

   set $\hat{x}_k = s_k, k = 1, \dots M$, $d^2 = \hat{d}^2$, $T_M = \hat{d}^2$ goto 2

   else goto 4

2.2.2 Sphere Decoder's Complexity

The performance of the algorithm depends on the value of the initial search radius $\sqrt{C}$. The initial search sphere should be large enough, to ensure that it contains the solution. However, if the initial radius is too large, the search becomes time-consuming. In [10] it is shown that the number of arithmetic operations is

$$\mathcal{O}\left(M^2 \times \left(1 + \frac{M-1}{4dC}\right)^{4dC}\right) \tag{2.21}$$

where d^{-1} is a lower bound for the eigenvalues of the positive definite Gram matrix $\mathbf{G} = \mathbf{U}^T\mathbf{U}$.

Moreover in [12], it is noted that, adjusting the initial radius to $C = d^{-1}$, the sphere decoder's complexity can be approximated by $\mathcal{O}(M^6)$, .

In [14] Hassibi and Vikalo propose a formula for choosing C

$$C = \alpha M \sigma^2 \tag{2.22}$$

where α is such that the quantity

$$\int_0^{\alpha M/2} \frac{x^{(M/2-1)}}{\Gamma(M/2)} e^{-x} dx \tag{2.23}$$

which is the probability of finding at least one point inside the sphere, is close enough to 1 (say 0.99). The same authors have also given a closed form for the sphere decoder's average computational complexity, which has been used to compute the SD complexity over a wide range of SNR and of M. In [15] the results of this study are summarized to the fact that, for relatively high SNRs, the sphere decoder's complexity (as a function of M) approaches cubic when the transmit signal constellation is such that the data rate is smaller than the capacity of the relevant multi-antenna channel. It is also found that the complexity is a decreasing function of SNR, and it is also roughly cubic when the data rate can be supported by the channel.

2.3 Probabilistic Data Association Algorithm

The Probabilistic Data Association filter was originally proposed as a highly successful approach to target tracking in the case that the measurements are unlabeled and may be spurious. Recently, it has been re-introduced for multiuser decoding in synchronous CDMA [16] and for multiple antenna systems [17]. It is based on a repeated conversion of a multimodal Gaussian mixture probability structure to a single Gaussian with matched mean and covariance. In this section we present the basic steps of the PDA algorithm and describe certain modification which improve its complexity. Our description of the algorithm is based on that in [16].

To apply the PDA algorithm we must construct a system with a positive definite (PD) and symmetric channel matrix. So in system (1.1) if the mixing matrix is not symmetric and PD, but is of full rank, we multiply both sides of (1.1) with the transpose of $\mathbf{H}$ and we obtain

$$\mathbf{z} = \mathbf{G}\mathbf{s} + \mathbf{v} \tag{2.24}$$

with $\mathbf{G} = \mathbf{H}^T\mathbf{H}$, $\mathbf{z} = \mathbf{H^T y}$ and $\mathbf{v} = \mathbf{H^T n}$. Passing $\mathbf{z}$ through a decorrelator (i.e., multiplying it from the left by $\mathbf{G}^{-1}$), system (2.24) becomes

$$\tilde{\mathbf{z}} = \mathbf{G}^{-1}\mathbf{z} = s_i\mathbf{e}_i + \sum_{j \neq i} s_j\mathbf{e}_j + \tilde{\mathbf{v}} \tag{2.25}$$

where $\tilde{\mathbf{v}} = \mathbf{G}^{-1}\mathbf{v}$, and $\sigma^2\mathbf{G}^{-1}$ is the covariance matrix of $\tilde{\mathbf{v}}$. The variable s_i represents the i-th element of $\mathbf{s}$, and $\mathbf{e}_i$ is an $M \times 1$ vector with its i-th component set to 1 and the rest of its entries set to 0.

In (2.25), the transmit components $s_i,\ i = 1, ..., M$ are considered as binary random variables. For each element of $\mathbf{s}$, $P_b(i)$ denotes the probability of s_i to be 1 and $1 - P_b(i)$ the probability for the same symbol to be -1. The PDA algorithm decides for the values of each data component , based on the corresponding probability. This probability is given by $P_b(i) = p(s_i = 1|\tilde{\mathbf{y}}, \{P_b(j)\}_j \neq i)$. The computational cost of computing directly the a posteriori probability is exponential. To circumvent this, the PDA considers as effective noise, for each component of the data vector, the noise plus the interference random variable :

$$\mathbf{N_i} = \sum_{j \neq i} s_j\mathbf{e}_j + \tilde{\mathbf{v}}, \tag{2.26}$$

on which the "Gaussian Forcing" idea is applied: that is, it is assumed that $\mathbf{N_i}$ is an approximately Gaussian random variable (each component of which depends on the values of the probabilities $P_b(j),\ i \neq j$) with mean

$$E[\mathbf{N_i}] = \sum_{j \neq i} \mathbf{e}_j(2P_b(j) - 1), \tag{2.27}$$

and covariance

$$Cov[\mathbf{N_i}] = \sum_{j \neq i} [4P_b(j)(1 - P_b(j))\mathbf{e}_j\mathbf{e}_j^T] + \sigma^2\mathbf{G}^{-1} \tag{2.28}$$

Letting

$$\begin{aligned} \theta_i &= \sum_{j \neq i} \mathbf{e}_j(2P_b(j) - 1) - \tilde{\mathbf{z}} \\ \mathbf{\Omega}_i &= \sum_{j \neq i} [4P_b(j)(1 - P_b(j))\mathbf{e}_j\mathbf{e}_j^T] + \sigma^2\mathbf{G}^{-1} \end{aligned} \tag{2.29}$$

The likelihood ratio for the i-th component of $\mathbf{s}$ is computed by

$$\frac{P_b(i)}{1 - P_b(i)} = exp\left\{-2\theta_i^T\mathbf{\Omega}_i^{-1}\mathbf{e}_i\right\} \tag{2.30}$$

The PDA algorithm consists of the following steps:

1. Sort the system (2.24) according to the user ordering criterion proposed in the decision feedback detector in [18, Theorem 1].

2. $\forall$ i, initialize the probabilities as $P_b(i) = 0.5$. Set the stage counter $k = 1$.

3. initialize the symbol counter $i = 1$

4. For symbol i, based on the current values of $P_b(j)(j \neq i)$ calculate $\theta_i, \mathbf{\Omega}_i^{-1}$ and compute the likelihood ratio via

$$\frac{P_b(i)}{1 - P_b(i)} = exp\left\{-2\theta_i^T\mathbf{\Omega}_i^{-1}\mathbf{e}_i\right\} \tag{2.31}$$

5. If $i < M$, increase i by 1 and go to step 4

6. if $\forall i$, $P_b(i)$ has converged, go to step 7. Otherwise increase k by 1 and return to step 3.

7. $\forall i$, make a decision on the s_i via

$$s_i = \begin{cases} 1, & P_b(i) \geq 0.5 \\ -1, & P_b(i) < 0.5 \end{cases} \tag{2.32}$$

For further improving the complexity of the PDA algorithm, some modifications have been proposed in [16]. Firstly, the computation of the probability $P_b(i)$ requires the calculation of the inverse $\mathbf{\Omega}_i^{-1}$. This leads to a computational complexity of $\mathcal{O}(M^4)$ for each PDA stage. To reduce the computational cost of calculating the inverse the following variables are used :

$$\begin{aligned} \theta = \sum_j \mathbf{e}_j(2P_b(j)-1) - \tilde{\mathbf{z}} = \theta_i + \mathbf{e}_i((2P_b(i)-1) \\ \mathbf{\Omega} = \sum_j [4P_b(j)(1-P_b(j))\mathbf{e}_j\mathbf{e}_j^T] + \sigma^2\mathbf{G}^{-1} \\ = \mathbf{\Omega}_i + 4P_b(i)(1-P_b(i))\mathbf{e}_i\mathbf{e}_i^T \end{aligned} \tag{2.33}$$

Then, by the Shermann-Morrison-Woodbury formula [16],[19] we have

$$\begin{aligned} \theta_i = \theta - \mathbf{e}_i(2P_b(i)-1) \\ \mathbf{\Omega}_i^{-1} = \mathbf{\Omega}^{-1} + \frac{4P_b(i)(1-P_b(i))\mathbf{\Omega}^{-1}\mathbf{e}_i\mathbf{e}_i^T\mathbf{\Omega}^{-1}}{1-4P_b(i)(1-P_b(i))\mathbf{e}_i\mathbf{\Omega}_i^{-1}\mathbf{e}_i} \end{aligned} \tag{2.34}$$

$$\begin{aligned} \theta = \theta_i + \mathbf{e}_i(2P_b(i)-1) \\ \mathbf{\Omega}^{-1} = \mathbf{\Omega}_i^{-1} - \frac{4P_b(i)(1-P_b(i))\mathbf{\Omega}_i^{-1}\mathbf{e}_i\mathbf{e}_i^T\mathbf{\Omega}_i^{-1}}{1+4P_b(i)(1-P_b(i))\mathbf{e}_i^T\mathbf{\Omega}_i^{-1}\mathbf{e}_i} \end{aligned} \tag{2.35}$$

So, step 4 of the PDA algorithm is executed in three substeps as follows:

4.a. Calculate θ_i and Ω_i^{-1} using (2.34).

4.b. Obtain $P_b(i)$ using (2.41).

4.c. update θ and Ω^{-1} according to (2.35) by using the new $P_b(i)$.

As a consequence, the computational cost of step 4 is reduced to $\mathcal{O}(M^2)$ and the overall complexity of each PDA stage to $\mathcal{O}(M^3)$.

In addition, the computational cost of PDA algorithm can be further reduced using successive cancelation [16]. After each stage of the PDA algorithm, a significant number of symbols may have converged and there is no reason to continue the execution of the PDA procedure for them. So at the end of the k-th stage, the following set is defined

$$P_b(i) = \in [0, \epsilon] \cup [1 - \epsilon] \qquad \forall i \in D. \tag{2.36}$$

For every $i \in D$ the algorithm decides for the corresponding transmit components as follows:

$$s_i = sign(P_b(i) - 0.5) \qquad \forall i \in D \tag{2.37}$$

with ϵ to be a small positive number. $\overline{D}$ is the complement of D. Canceling the influence of the converged symbols yields the following decorrelated system for the non-converged symbols

$$\mathbf{G}_{\overline{D}\overline{D}}^{-1}\mathbf{z}_{\overline{D}} - \mathbf{G}_{\overline{D}\overline{D}}^{-1}\mathbf{G}_{\overline{D}D}\mathbf{s}_D = \mathbf{s}_{\overline{D}} + \tilde{\mathbf{v}}_{\overline{D}} \tag{2.38}$$

Where $\mathbf{G}_{\overline{\mathbf{DD}}}$ stands for the matrix which consists of only the columns and rows of $\mathbf{G}$ which correspond to symbols in $\overline{D}$. Similarly, the rest of the vectors and matrices of (2.38) have the respective meanings. So, for the $k + 1$ PDA stage we update the system for which PDA algorithm is performed, by the reduced dimensionality system (2.38).

2.4 Hybrid PDA-SD Algorithm

As we have seen so far, the PDA is an algorithm for obtaining a solution with low computational complexity (cubic for every level of SNR). Moreover, the sphere decoder reduces significantly the computational cost of solving ILS relative to exhaustive search, while achieving a probability of error very close to the optimal solution, with a good choice of the initial radius. Despite the very significant complexity improvement that it affords over exhaustive search, SD complexity remains relatively high for large values of M and at low and moderate SNRs. In this section we propose a hybrid algorithm which runs a one-stage PDA procedure, proceeds with the interference cancelation idea introduced in [16] (to improve the computational complexity of the PDA algorithm), and *then* applies the sphere decoder in the reduced dimensionality system which appears after the cancelation procedure. Computer simulations which are presented in Chapter 3, show that this algorithm has BER performance close to that of the sphere decoder for a wide range of SNRs, and significantly reduced computational cost for relatively high values of M.

2.4.1 The Algorithm

The basic idea of this algorithm is to make decisions for some of the symbols of $\mathbf{s}$ using a stage of the PDA detector, which is a low-complexity method, and after this to apply the Sphere Decoder only for the symbols for which the first stage has not already decided.Assuming the case that the mixing matrix in (1.1) is of full rank , we multiply (1.1) from the left with $\mathbf{H}^T$ to obtain a system model for the PDA detector

$$\mathbf{z} = \mathbf{G}\mathbf{s} + \mathbf{v}. \tag{2.39}$$

Matrix $\mathbf{G}$ is then symmetric positive definite by construction. We then apply one stage of the PDA detector for the system (2.39) and we obtain a vector of probabilities $\mathbf{P_b}$ for all symbols. We denote D to be the symbols that satisfy

$$P_b(i) \in [0, \epsilon] \cup [1 - \epsilon] \qquad \forall i \in D \tag{2.40}$$

with ϵ to be a small non-negative number properly selected. We also denote $\overline{D}$ to be the complementary set of D. We make decisions for the symbols of D

$$s_i = sign(P_b(i) - 0.5) \qquad \forall i \in D \tag{2.41}$$

We define $\mathbf{G_{\overline{D}\overline{D}}}$ to be the matrix that only contains the rows and the columns of $\mathbf{G}$ corresponding to the symbols of $\overline{D}$ and $\mathbf{y_{\overline{D}}}$, $\mathbf{v_{\overline{D}}}$ to be the received and noise vectors respectively which contain only the entries corresponding to the above symbols. We then compute the effect of the symbols in D to the elements of $\mathbf{y_{\overline{D}}}$.

$$\mathbf{c} = \mathbf{G}_{\overline{D}D}\mathbf{s}_D \tag{2.42}$$

We define $\mathbf{y_c} = \mathbf{y}_{\overline{\mathbf{D}}} - \mathbf{c}$ as the received vector which corresponds to the symbols of $\overline{D}$ without the effect of the rest of the symbols. The resulting subsystem can be written as

$$\mathbf{y_c} = \mathbf{G}_{\overline{DD}}\mathbf{s}_{\overline{D}} + \mathbf{v}_{\overline{D}} \tag{2.43}$$

Assuming perfect decisions (and subsequent cancelation of the effects of the symbols in D) the noise vector in (2.43) is colored Gaussian with zero mean and covariance matrix $\sigma^2\mathbf{G}_{\overline{DD}}$. We compute the Cholesky factor of $\mathbf{G}_{\overline{DD}}$. Let

$$\mathbf{G}_{\overline{DD}} = \mathbf{L}_{\overline{DD}}^T\mathbf{L}_{\overline{DD}}. \tag{2.44}$$

We define

$$\mathbf{\Psi}_{\overline{DD}} = (\mathbf{L}_{\overline{DD}}^T)^{-1}. \tag{2.45}$$

We multiply system (1.53) from the left with $\mathbf{\Psi}_{\overline{DD}}$ and we obtain the system

$$\mathbf{r} = \mathbf{\Psi}_{\overline{DD}}\mathbf{y_c} = \mathbf{\Lambda}\mathbf{s}_{\overline{D}} + \mathbf{w} \tag{2.46}$$

The noise vector $\mathbf{w}$ is white Gaussian with covariance matrix $\sigma^2\mathbf{I}$. Now, we apply the Sphere Decoder to the subsystem (2.46). Define K as the number of the elements for which the PDA stage has not decided. The initial radius for the Sphere Decoder Stage of the Algorithm is set to

$$C = aK\sigma^2 \tag{2.47}$$

with a such that

$$\int_0^{aK/2} \frac{x^{(K/2-1)}}{\Gamma(K/2)} e^{-x} dx = 0.99 \tag{2.48}$$

The steps of the proposed combination algorithm are summarized in Table (2.3)

2.4.2 Threshold Parameter-Computational Complexity

The Threshold ϵ is an important parameter for the hybrid algorithm. If ϵ is large, the algorithm may take a wrong decision for an element in the PDA stage. This has destructive results, because the Sphere Decoder cannot then estimate correctly the elements of $\overline{D}$ (successive cancelation is based on wrongly detected elements of D). On the other hand ϵ should not be very small, so that only a moderate number of elements will be passed to the computationally expensive Sphere Decoder. As we will see in the next chapter the parameter ϵ should be increased with σ^2 (decreased with the SNR). We will also provide means of a suitable ϵ in the case of a Multiple Antenna System.

As indicated by simulations for the V-BLAST architecture, the computational cost of the PDA stage of the above algorithm dominates the computational complexity of the overall hybrid algorithm for M up to 50. In fact the cost of the Sphere Decoder part is very small compared to the cost of the PDA part. Hence, the overall complexity of the algorithm for these values of M is roughly cubic.

Table 2.3: Sphere decoder with PDA first stage

Input $\mathbf{H}, \mathbf{y}, \sigma^2, \epsilon$

1. Multiply $\mathbf{H}, \mathbf{y}$ from the left with $\mathbf{H^T}$, denote $\mathbf{G} = \mathbf{H^T H}$

2. Order system according to [18, theorem 1]. Run one stage of the PDA procedure (steps 2-5 of the PDA procedure described in [16])

3. Denote by D the set of elements for which we have made a decision via

$$s_i = \begin{cases} 1 \quad if \quad P_b(i) \geq 1 - \epsilon \\ -1 \quad if \quad P_b(i) \leq \epsilon \end{cases}$$

   and $\bar{D}$ as its complement

4. Create vectors $\mathbf{y}_{\bar{D}}, \mathbf{s}_D, s_{\bar{D}}$ and matrices $\mathbf{G}_{\overline{DD}}$, $\mathbf{G}_{\bar{D}D}$, as in (2.42),(2.43)

5. Compute $\mathbf{y_c} = \mathbf{y}_{\bar{D}} - \mathbf{G}_{\bar{D}D}\mathbf{s}_D$

6. Compute $\mathbf{L} = chol(\mathbf{G}_{\overline{DD}})$

7. Compute $\mathbf{r} = (\mathbf{L}^T)^{-1}\mathbf{y_c}, \ \mathbf{\Lambda} = (\mathbf{L}^T)^{-1}\mathbf{G}_{\overline{DD}}$

8. Run the Sphere decoder with inputs $\mathbf{r}, \mathbf{\Lambda}$ and initial radius chosen so that the probability of finding at least one point inside the sphere is 0.99 (cf (2.48))

Chapter 3

MIMO models - Simulation results

In the present chapter we describe certain MIMO communication models and study the performance of SDR, SD, PDA and hybrid PDA-SD applied to them, using computer simulation examples. The models which we study are the Synchronous CDMA system model and the multiple Antenna V-BLAST system model .

3.1 Synchronous CDMA

3.1.1 System Model

In the synchronous Code Division Multiple Access (CDMA) system, symbols directed to several users are transmitted simultaneously from a base station to the users using proper signature sequences. Consider the synchronous CDMA system depicted in figure (3.1). The amplitude (the square-root of the received signal energy) for the $k-$th user is denoted by a_{kk}, $s_k(t)$ stands for the $k-$th user's signature waveform and $w(t)$ is a white Gaussian stochastic process. T denotes the symbol duration.

In the receiver, the baseband signal $y(t)$ of the $i-$th symbol period is given by

$$y(t) = \sum_{k=1}^{K} b_k(i) a_{kk} s_k(t - iT) + w(t) \quad t \in [iT, iT + T] \tag{3.1}$$

where $b_k(i)$ is the bit which corresponds to the k-th user. We also assume that each signature waveform has unit (normalized) energy.

$$\int_0^T s_k^2(\tau) d\tau = 1 \tag{3.2}$$

The output of the $k-$th user matched filter can be represented by

$$y_k = \int_0^T y(\tau) s_k(\tau) d\tau \tag{3.3}$$

So, using vector notation, the output of the symbol matched filter bank, for the $i-$th symbol period can be written as

$$\mathbf{y} = \mathbf{R}\mathbf{A}\mathbf{b} + \mathbf{n}. \tag{3.4}$$

where $\mathbf{A}$ is the diagonal matrix whose k-th diagonal element is equal to a_{kk}, R is the $K \times K$ cross-correlation matrix each entry of which is given by

$$r_{kj} = \int_0^T s_k(\tau)s_j(\tau)d\tau, \quad k = 1, ..., K, ; \ j = 1, ..., K, \tag{3.5}$$

and n denotes a noise Gaussian vector with zero mean and covariance $\sigma^2\mathbf{R}$. The vector of bits transmitted by the K users is denoted by $\mathbf{b}$.

In Direct Sequence (DS) CDMA systems, the user signature waveforms are constructed by delayed versions of particular signature waveform, called chip, and a corresponding binary signature sequence. So the signature waveform is given by

$$s_k(t) = \sum_{k=1}^{N} c_{ki}\psi(t - (i-1)T_c) \tag{3.6}$$

where N is called the spreading factor [1] of the system, $\psi(t)$ represents the chip waveform, and c_{ki} denotes the $i-$th element of the signature sequence. We also assume that the binary signature sequence and the chip waveform are both normalized, so

$$\begin{gathered} \sum_{i=1}^{N} c_{ki}^2 = 1 \\ \int_0^{T_c} \psi(t)^2 dt = 1 \end{gathered} \tag{3.7}$$

In fig. (3.2) we illustrate the DS-CDMA receiver with the chip matched filter. The $n-$th sample of the chip matched filter output for the $i-$th symbol period is given by

$$x(iT + nT_c) = \sum_{k=1}^{K} c_{kn}a_{kk}b_k(i) + w_n(i) \tag{3.8}$$

where $w_n(i)$ is a white Gaussian noise variable. Letting $\mathbf{c}_k = [c_{k1}c_{k2}....c_{kN}]$ and

$\mathbf{x} = [x_1 x_2 ... x_N]$, the output of the $k-$th user matched filter can be written as

$$y_k = \mathbf{c}_k \mathbf{x}^T, \tag{3.9}$$

and using vector notation the outputs of the K matched filters can be represented by

$$\mathbf{y}(i) = \mathbf{RAb} + n \tag{3.10}$$

where $\mathbf{n}$ is Gaussian noise vector with zero mean and covariance matrix $\sigma^2\mathbf{R}$. Now, multiplying equation (3.10) from the left by the inverse of the cholesky factor $\mathbf{L}^T\mathbf{L} = \mathbf{R}$, we obtain the system

$$\tilde{\mathbf{y}} = (\mathbf{L}^T)^{-1}\mathbf{y} = (\mathbf{L}^T)^{-1}\mathbf{RAb} + (\mathbf{L}^T)^{-1}\mathbf{n} = (\mathbf{L}^T)^{-1}\mathbf{RAb} + \mathbf{w} \tag{3.11}$$

where $\mathbf{w}$ is a white Gaussian noise vector with zero mean. So, we can easily observe that system (3.11) is a case of the model (1.1) with $M = N = K$.

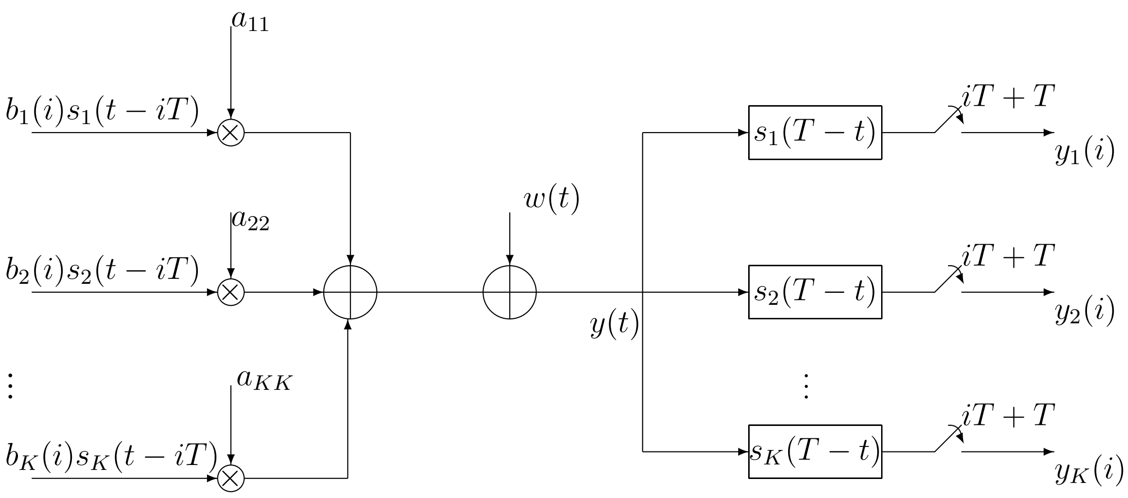

Figure 3.1: Synchronous CDMA system

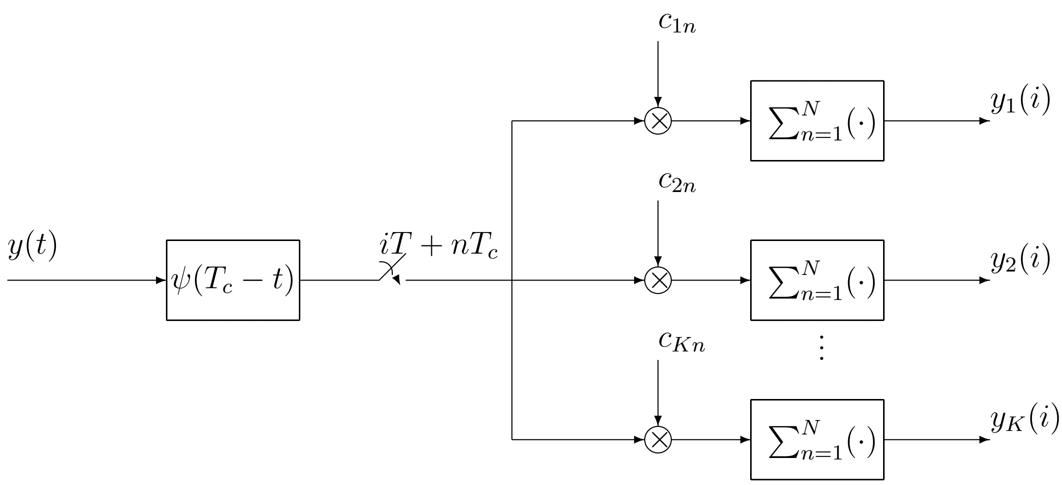

Figure 3.2: DS-CDMA receiver with chip-matched filter.

3.1.2 DS-CDMA Simulation Results

In this subsection we run computer simulations in order to compare the performance of Semidefinite Relaxation, Sphere Decoder and Probabilistic Data Association Algorithms. For solving the semidefinite program in the SDR algorithm we use the package SDPPACK [20]. Our implementation of PDA does not incorporate the bit-flip stage [16].

Example 1: In the first example, we show the probability of error performance versus SNR, for algorithms SDR, PDA and SD. We use 31-length Gold codes as signature sequences. The matrix $\mathbf{A}$ changes for every group of 100 transmissions. The user signal amplitudes are randomly and independently generated by $a_{ii} \sim N(4.5, 4), \forall i$ and are limited within the range $[2, 7]$. The number of randomizations for SDR is set to 20. The initial radius of SD is chosen so that the probability of finding at least one point inside the sphere is 0.99. For the PDA algorithm, $\epsilon = 10^{-2}/4SNR$. Figure (3.3) shows the results of this example in which we use dynamic Monte-Carlo simulation, that is the simulation is running for each value of SNR until the number of errors which occur reaches 100, and a minimum number of 15000 runs.

Example 2: In the second example we compare the computational efficiency of Sphere Decoder and PDA algorithm. We use random signature sequences with spreading factor $Q = 1.2K$. We also set the SNR to 12 dB. The number of users varies from 2 to 50. Figure (3.4) shows the results of the simulation. We observe that the PDA algorithm is significantly faster than the sphere decoder for

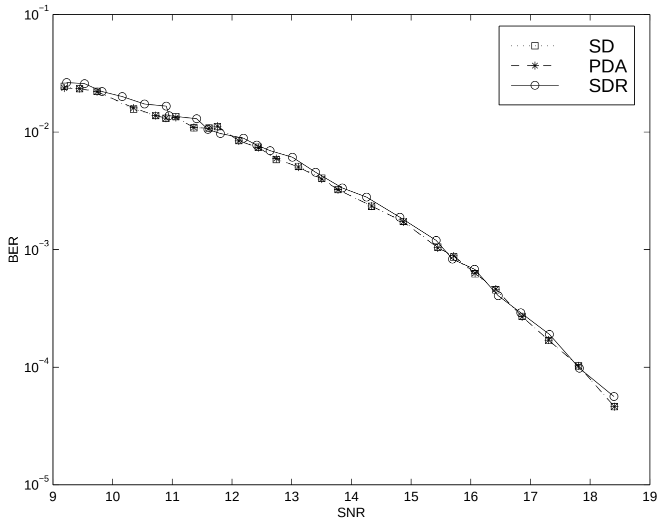

Figure 3.3: Example 1. Probability of error versus SNR for DS-CDMA with 29 users, 31- length Gold signatures sequences

relatively large number of users. The number of Monte-Carlo runs is 1000. (**Remark: We have not measured the computational cost of SDR becaused we used SDPPACK for its implementation**).

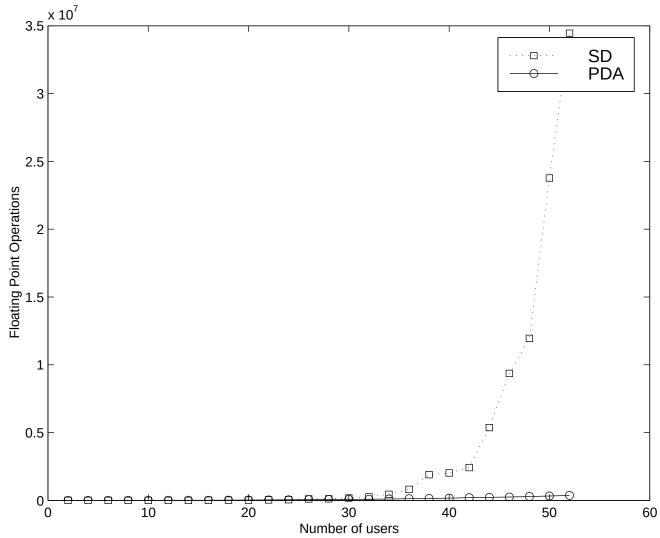

Figure 3.4: Example 2. Average computational cost versus K, random signature sequences, spreading factor=1.2K, $SNR = 12db$

3.2 V-Blast Multiple Antenna Architecture

V-BLAST [2] is a multiple antenna architecture which does not use spatial coding (e.g., as in space-time codes) but relies instead on temporal coding of the individual symbol streams. This way, high spectral efficiencies are attainable via spatial multiplexing; hence the interest in V-BLAST .

3.2.1 System Model

The V-BLAST architecture is a symbol synchronized multiple antenna system with n_T transmit and n_R receive antennas, where $n_T \leq n_R$. The input stream of bits is mapped to a particular constellation and the resulting symbol stream is demultiplexed into n_T substreams. The transmissions are organized into bursts of L symbols. It is assumed that the channel is quasi-stationary, which means that its variation is negligible over the L symbol periods comprising a burst, but it varies randomly from one burst to the next. The channel is accurately estimated for every burst at the receiver, through the use of training bits but it is uknown to the transmitter. From the discrete-time baseband-equivalent viewpoint the system can be represented as:

$$\mathbf{r} = \sqrt{\frac{\rho}{n_T}}\mathbf{H}\mathbf{s} + \mathbf{v} = \mathbf{A}\mathbf{s} + \mathbf{v} \tag{3.12}$$

where $\mathbf{r} = [r_1, r_2, \ldots, r_{n_R}]^T, \mathbf{s} = [s_1, s_2, \ldots, s_{n_T}]^T$ are the receive and the transmit vector, respectively, $\mathbf{H}$ is a generally complex $n_R \times n_T$ channel matrix with entries h_{ij} and $\mathbf{v}$ is a Gaussian $n_R \times 1$ noise vector with zero mean and covariance matrix

$2\sigma^2\mathbf{I}$. The normalized amplitude $\sqrt{\frac{\rho}{n_T}}$ ensures that the SNR is constant for a given noise variance, independently of n_T. Assuming rich scattering, the elements of $\mathbf{H}$ are considered i.i.d. complex Gaussian variables with zero mean and with unit variance of the real and imaginary parts. We assume that the transmit symbols are taken from a complex constellation with binary real and imaginary parts (4-QAM). In order to transform the above model to the analogous real model we denote

$$\tilde{\mathbf{s}} = [\Re(\mathbf{s}^T) \quad \Im(\mathbf{s}^T)]^T \tag{3.13}$$

$$\tilde{\mathbf{r}} = [\Re\{\mathbf{r}^T\} \quad \Im\{\mathbf{r}^T\}]^T \tag{3.14}$$

$$\tilde{\mathbf{H}} = \begin{bmatrix} \Re\{\mathbf{H}\} & -\Im\{\mathbf{H}\} \\ \Im\{\mathbf{H}\} & \Re\{\mathbf{H}\} \end{bmatrix} \tag{3.15}$$

$$\tilde{\mathbf{v}} = [\Re\{\mathbf{v}^T\} \quad \Im\{\mathbf{v}^T\}]^T \tag{3.16}$$

Using the above vectors and matrices we obtain the real-valued vector equation

$$\tilde{\mathbf{r}} = \sqrt{\frac{\rho}{n_T}}\tilde{\mathbf{H}}\tilde{\mathbf{s}} + \tilde{\mathbf{v}} \tag{3.17}$$

We can easily observe that system (3.17) is a case of (1.1) with $N = 2n_R$ and $M = 2n_T$.

3.2.2 Simulation Results

In the present section we run several simulation examples to compare the decoding algorithms presented in the previous chapter over the described multiple antenna model. For the computer simulations presented below, the burst length is set to 100 symbol durations, the power ρ is set to 8 (the transmitted symbols have unit average energy). The initial sphere radius of the Sphere Decoder is adjusted so that the probability of finding at least one point inside the sphere is 0.99. If SD fails to find a point inside the sphere the initial radius is increased by 1. This is repeated 5 times at most. The transmitted symbols are taken from the 4-QAM constellation. The SNR is calculated by $SNR = 10log_{10}\frac{\rho}{\sigma^2}$. For the PDA algorithm we set the threshold number $\epsilon = 10^{-2}/4SNR$ as in [16]. For the hybrid PDA-SD algorithm we set $\epsilon = 10^{-p}, p = 3.5(\sigma^2)^{-1.55}$, limited by $0.45 \geq \epsilon \geq 0$. The implementation of PDA does not incorporate the bit-flip stage [16]. For the semidefinite programming part of the SDR algorithm we use the package SDPPACK [20], and the number of randomizations is $M_{rand} = 2 \cdot 10n_T$

Example 1: In the first example we compare the probability of error (BER) which is exhibited by the four algorithms under study, versus SNR. The number of the receive and transmit antennas is considered to be 16 ($n_T = n_R = 16$). We keep the value of ρ constant, changing the value of the noise variance σ^2 from 3.6 to 0.25. For the present example we use dynamic Monte-Carlo simulation to avoid useless Monte-Carlo iterations. For every value of SNR that it is used, the simulation stops when both the number of errors has reached 150 and the number of the simulated bursts has reached 5. The results of example 1 are presented in figure (3.5).

We observe that the BER performance of SD and the hybrid PDA-SD algorithm are very close for values of SNR up to 13.5 dB. Also observe that the

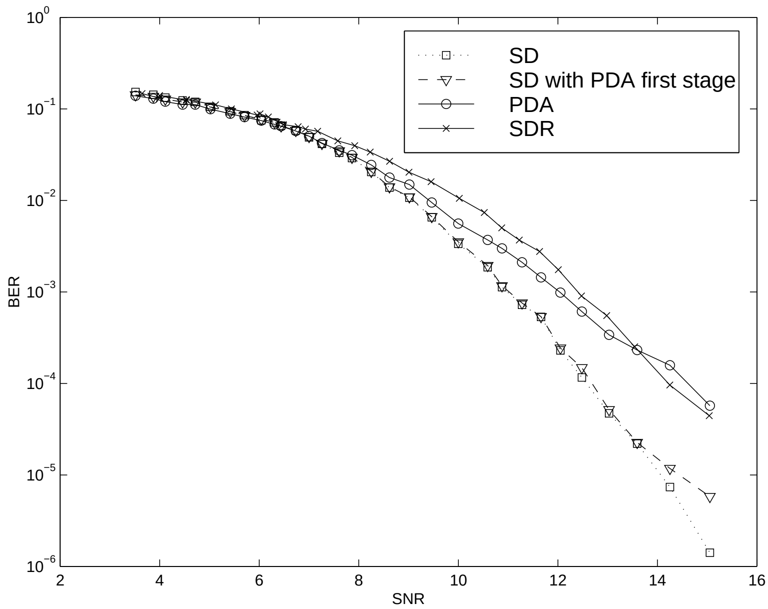

Figure 3.5: Example 1. Probability of error comparison for 4-QAM with $n_T = n_R = 16$.

performance of PDA is relatively poor for SNRs larger than 8.5 db.

Figure (3.6) presents the average number of elements of $\tilde{\mathbf{s}}$ whose value is decided from the PDA part of the combination algorithm, versus SNR. We see that this number decreases with SNR in order to preserve the BER performance of the algorithm close to the performance of SD algorithm. For SNRs larger than 13.5 db this number starts to increase but this has negative results for the BER performance of the combination algorithm (the algorithm yields BER larger than that of the SD).

Example 2: In the second example we present the average computational cost (number of floating point operations) of the SD, PDA and the proposed algorithm for decoding one transmit vector, versus the SNR. The system param-

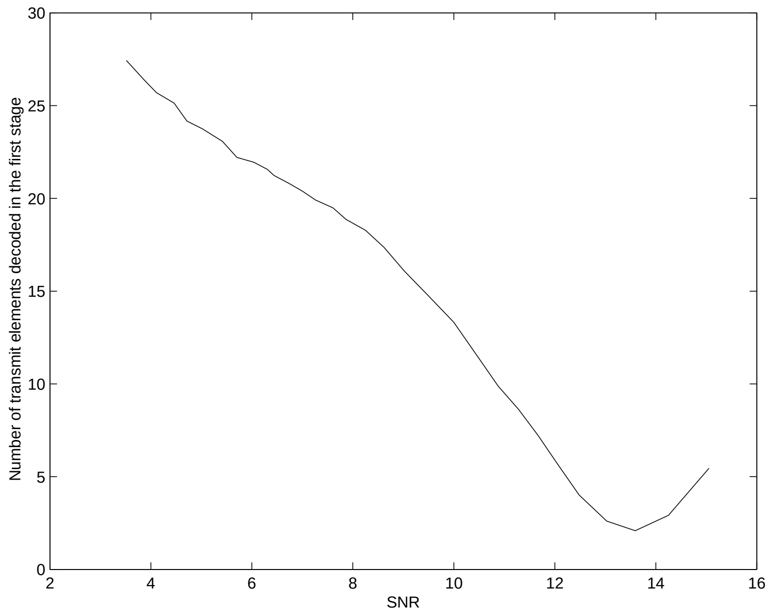

Figure 3.6: Example 1. Average number of transmit vector elements the value of which is decided from the PDA stage of the proposed combination algorithm, $n_T = 16$

eters are the same as in the previous example. The results are shown in figure (3.7). The number of monte-carlo iterations is 10^3. We observe that, for relatively low SNRs, SD requires a much larger number of flops than the PDA, and the proposed hybrid PDA-SD algorithm.

Example 3: In the third example we simulate the multiple-antenna model using $n_T = n_R = 8$. The noise variance σ^2 varies from 3.6 to 0.2. The rest of the parameters of the model are set as in the first example. The results of example 3 are shown in figures (3.8) and (3.9). Again we have used dynamic Monte-Carlo simulation.

Example 4: Here the average computational cost is presented for the SD,PDA

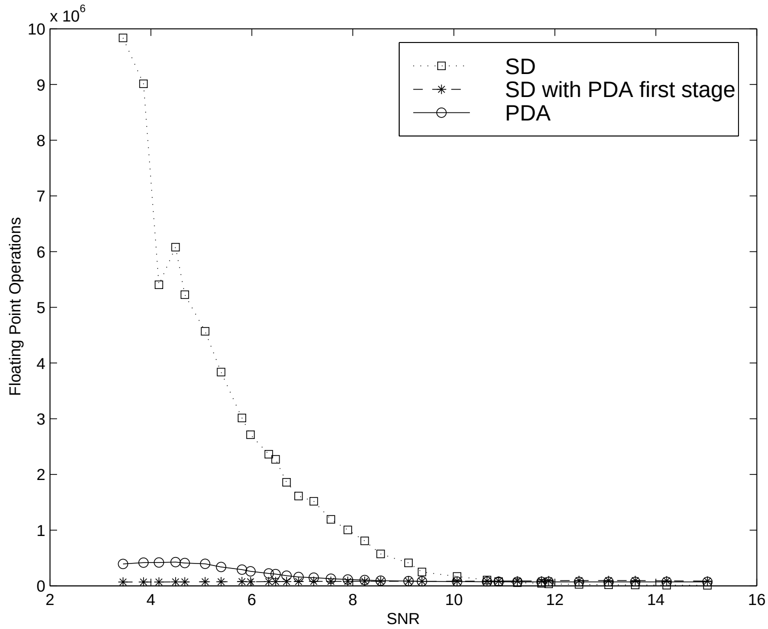

Figure 3.7: Example 2. Average computational cost versus SNR, $n_T = n_R = 16$, 4-QAM

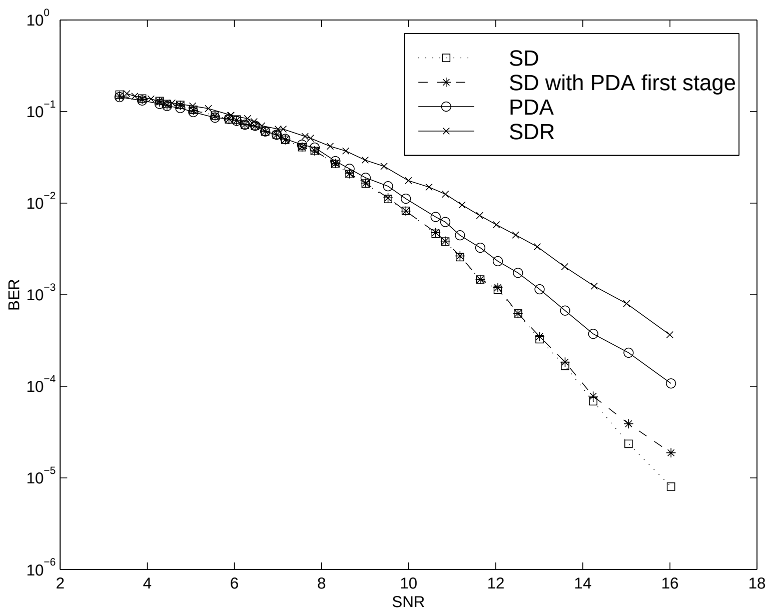

Figure 3.8: Example 3. Probability of error comparison for 4-QAM with $n_T = n_R = 8$.

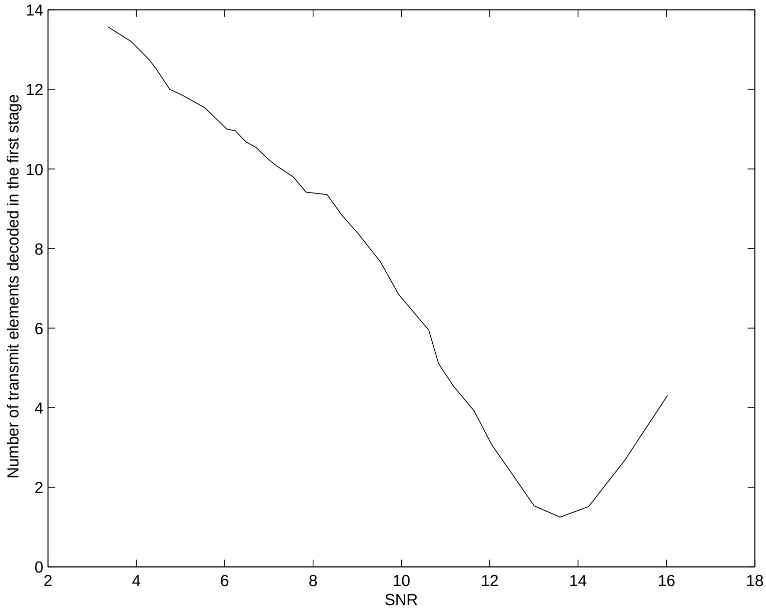

Figure 3.9: Example 3. Average number of transmit vector elements the value of which is decided from the PDA stage of the proposed combination algorithm, $n_T = 8$

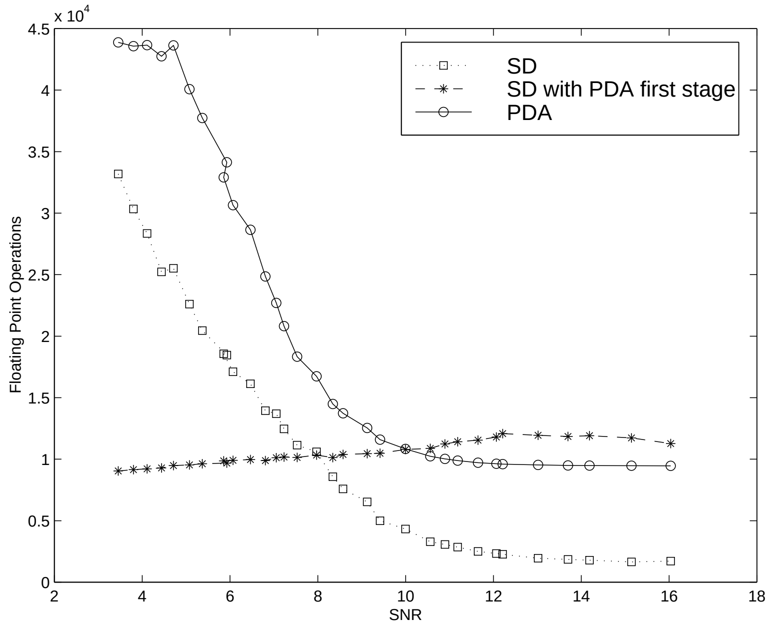

Figure 3.10: Example 4. Average computational cost versus SNR, $n_T = n_R = 8$, 4-QAM

and the proposed hybrid PDA-SD algorithm versus SNR, for the model simulated in the previous example. Figure (3.10) shows the results of this computational cost comparison. The number of Monte-Carlo runs is 10^3

Example 5: In this example the average computational costs of SD, PDA and the hybrid PDA-SD are assessed versus the number of transmit antennas n_T. We adjust the SNR to 10 dB and $n_T = n_R$. As we see for $n_T = 25$ the average computational cost of the SD is about 30 times greater than that of the proposed hybrid algorithm. The computational cost of the proposed algorithm is close to that of the PDA algorithm which, is $\mathcal{O}(n_T^3)$. The number of Monte-Carlo runs is 10^3. The results are shown in figure (3.11).

Example 6: We repeat the above example setting the SNR equal to 11 dB.

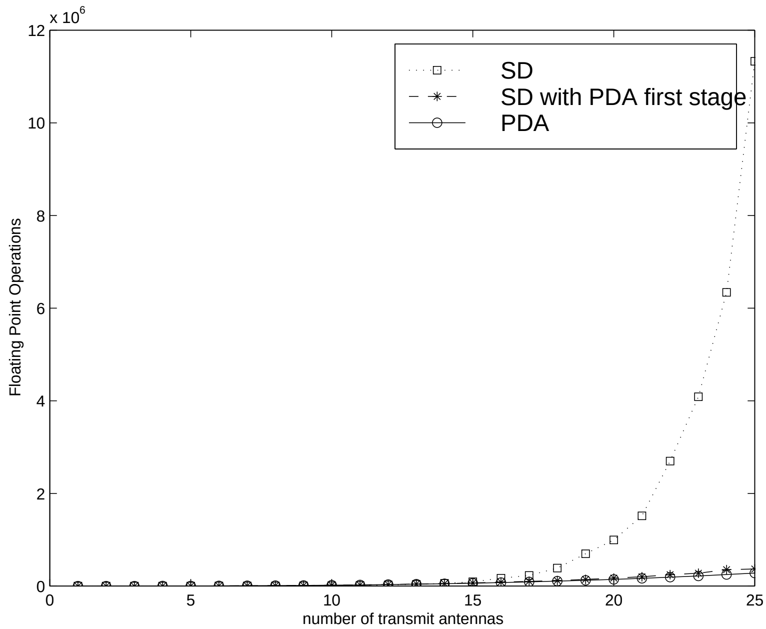

Figure 3.11: Example 5. Average computational cost versus n_T, $n_T = n_R$, 4-QAM, $SNR = 10db$

We also set $n_T = n_R$. The results are shown in figure (3.12). The combination algorithm is, in this case, about 6.5 times faster than the Sphere Decoder for $n_T = 25$. Here the average computational cost of the SD is significantly reduced in relation with its cost in the previous example. This occurs because the initial starting radius is smaller for $SNR = 11db$ than this for $SNR = 10db$.

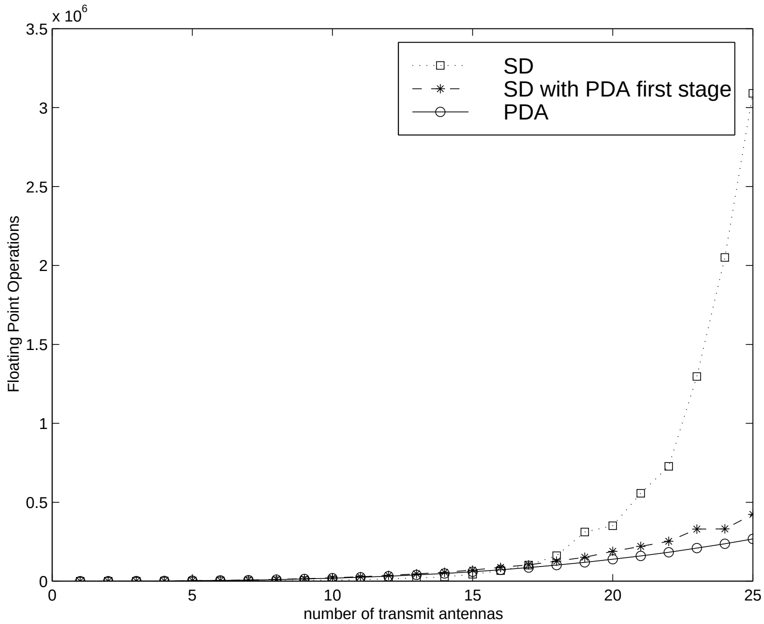

Figure 3.12: Example 6. Average computational cost versus n_T, $n_T = n_R$, 4-QAM, $SNR = 11db$

Chapter 4

Conclusions

In this Thesis we compared four near-optimal decoding algorithms applied in systems which employ the MIMO model (1.1). Using computer simulations, we showed that in the synchronous CDMA case the Probabilistic Data Association Algorithm is the most efficient algorithm between these studied, from the view of computational efficiency. For multiple Antenna system, the Sphere Decoder is the algorithm which exhibits the smallest bit error rate for most values of SNR. The proposed hybrid PDA-SD algorithm has performance very close to that of SD for a wide range of SNRs. For the other two algorithms (SDR, PDA), there are scenarios wherein they exhibit relatively poor probability of error performance. In addition, the proposed algorithm is significantly faster than SD at low and moderate SNRs in the multiple antenna system we studied. For these values of SNR, SD requires a large number of operations because the initial radius of the search sphere is increased with the noise variance σ^2. In fact the proposed algorithm has computational cost close to that of the PDA detector and exhibits better BER performance than the PDA.

The comparison of the described algorithms could extend to more cases where the MIMO model appears. These may be channels with memory, Linear Dispersion code systems [21], as well as cases that the transmit signals are taken from a larger complex constellation like 16 or 64-QAM.

Moreover the idea of combining Probabilistic Data Association and the Sphere Decoder can be extended to various size q-QAM constellations using one stage of the PDA procedure [17] before running the real or , alternatively, the complex version of SD [22].

Bibliography

[1] S. Verdu, “Multiuser Detection,” Cambridge University Press, New York, 1998.

[2] G.D. Golden, C.J. Foschini, R.A. Valenzuela and P.W. Wolniansky, “Detection Algorithm and Initial Laboratory Results Using V-BLAST Space-Time Communication Architecture,” *Electronic Letters*, vol. 35, pp. 14-15, Jan. 1999.

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

[4] W. Ma, T.N. Davidson, K.M. Wong,Z. Luo and P. Ching, “Quasi-maximum-likelihood multiuser detection using semi-definite relaxation with application to synchronous CDMA,” *IEEE Trans. Signal processing*, vol.50, no.4, pp.912-922, April, 2002.

[5] L.Vandenberghe and S. Boyd, “Semidefinite programming,” *SIAM Rev.*, vol. 38, pp. 49-95, 1996.

[6] F. Rendl, R. J. Vanderbei, and H. Wolkowicz, "Max-min eigenvalue problems, primal-dual interior point algorithms, and trust region subproblems," *Optim. Meth. Softw.*, vol. 5, pp. 1-16, 1995.

[7] C. Helmberg, F. Rendl, R. J. Vanderbei, and H.Wolkowicz, "An interior-point method for semidefinite programming," *SIAM J. Optim.*, vol. 6, no. 2, pp. 342-361, 1996.

[8] M. X. Goemans and D. P. Williamson, "Improved Approximation algorithms for maximum cut and satifiability problem using semi-definite programming," *J. ACM*, vol. 42, pp. 1115-1145, 1995.

[9] Y. E. Nesterov, "Quality of Semidefinite Relaxation for Nonconvex Quadratic Optimization," , *CORE Discussion* Paper #9719, Mar. 1997.

[10] 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.

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

[12] M. O. Damen, A. Chkeif, and J.-C. Belfiore, "Lattice code decoder for space-time codes," *IEEE Communication Letters*, pp. 161-163, May 2000.

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

[14] B. Hassibi and H. Vikalo, "On Sphere Decoding Algorithm. I. Expected Complexity," to appear in *IEEE Transactions on Signal Processing.*

[15] B. Hassibi and H. Vikalo, "On Sphere Decoding Algorithm. II. Generalizations, Second-order Statistics, and Applications to Communications," to appear in *IEEE Transactions on Signal Processing.*

[16] J. Luo, K. Pattipati, P. Willett and F. Hasegawa, "Near-Optimal Multiuser Detection in Synchronous CDMA using Probabilistic Data Association," *IEEE Comm. Lett.*, vol. 5, pp. 361-363, Sept. 2001.

[17] D. Pham, K.R. Pattipati, P.K. Willett, J. Luo, "A Generalized Probabilistic Data Association Detector for Multiple Antenna Systems," to appear in *IEEE Communication Letters.*

[18] M.K. Varanasi, "Decision Feedback Multiuser Detection: a Systematic Approach," *IEEE Trans. Inform. Theory*, vol. 45, pp. 219-240, Jan. 1999.

[19] W. W. Hager, "Updating the inverse of a matrix," *SIAM Review*, vol. 31, no. 2, pp. 221-239, 1989.

[20] SDPPACK Version 0.9 for Matlab 5.0, NYU Computer Science Dept Technical Report June 1997. web: http://www.cs.nyu.edu /faculty /overton /sdppack /sdppack.htm.

[21] B. Hassibi and B. Hochwald, "High-rate codes that are linear in space and time," *IEEE Trans. Info. Theory*, vol. 48, pp. 1804-1824, July 2002.

[22] B. Hochwald, S. ten Brink, "Achieving Near-Capacity on a Multiple Antenna Channel," *IEEE Trans. On Comm.*, vol. 51, pp. 389-399, March 2003.