Master Thesis Project

The Sphere Decoding Algorithm applied to Space-Time Block Codes

by
Shi Chen
shi@kth.se

at
Department of Signals, Sensors and Systems (S3)
Royal Institute of Technology (KTH)
Stockholm, Sweden

Advisor
George Jörgren
June 15, 2003
Abstract

The use of digital wireless communication systems has become more and more common during recent years. A multiple-input-multiple-output (MIMO) system using Space-Time Coding techniques can be implemented to enhance the capacity of a wireless link. The optimal decoder is based on the maximum likelihood principle. But as the number of the antennas in the system and the data rates increase, the maximum likelihood decoder becomes too complex to use. Examples of less complex decoding techniques used are zero-forcing and MMSE, as well as V-BLAST have been implemented at the price of reduced performance at the receiver. In this work, we investigate a new type of decoding algorithm called sphere decoding. As will be apparent from the development to follow, this algorithm delivers near optimal performance with reasonably low complexity.

We have investigated the performance of the sphere decoding algorithm. As it has shown in the computer simulations, the decoder based on the sphere decoding algorithm has almost the same performance of a maximum likelihood decoder with much lower complexity. Further simulations of the sphere decoding algorithms has shown, with the channel estimation error at the receiver, the decoder with the sphere decoding algorithm still has the same performance as in a ML decoder without increase the decoding complexity.
Acknowledgments

This master’s thesis is supported by and carried out with the Signal Processing Group at the department of Signal, Sensors and System (S3), Royal Institute of Technology (KTH) in Stockholm, Sweden.

First of all, I would like to thank Ph.D. Student George Jöngren for being my advisor, and for the valuable inputs and feedbacks in my thesis work, even though he has to prepare his own Ph.D. defence at the time. Secondly, I would like to thank Joakim Jaldén for discussions on the decoding algorithms. Finally, I would like to thank all people for creating a positive atmosphere in the lab.
# Contents

Abstract i

Acknowledgements ii

Notations vi

1 Introduction 1
   1.1 Background ........................................ 1
   1.2 Objective ........................................... 2
   1.3 Previous Works .................................... 2
   1.4 Thesis Outline .................................. 2
   1.5 Acronyms .......................................... 3

2 System Model 5
   2.1 Channel Model .................................... 5
   2.2 Data Model ....................................... 6

3 Space-Time Coding 9
   3.1 Overview .......................................... 9
   3.2 V-BLAST ........................................... 10
   3.3 The Alamouti Code ................................ 10
   3.4 Orthogonal Space-Time Block Codes ............. 12
   3.5 Linear Dispersive Codes ....................... 12

4 Conventional Decoder for Space-Time Coded System 15
   4.1 The Zero Forcing Decoder ....................... 15
   4.2 The Minimum Mean-Square Error Decoder ....... 15
   4.3 Decoding in V-BLAST ............................ 16
   4.4 The Maximum Likelihood Decoder .............. 16

5 Sphere Decoding 19
   5.1 Sphere Decoding Algorithm .................... 20
   5.2 Implementation .................................. 22
# 6 Simulation Results

6.1 Fixed Radius versus Adaptive Radius ............................ 26  
6.2 V-BLAST .......................................................... 28  
6.3 Alamouti Code ...................................................... 30  
6.4 Linear Dispersive Codes .......................................... 31  
6.5 Random Generated Weighting Matrices .......................... 31

# 7 Channel Estimation Errors

7.1 Error Model ......................................................... 33  
7.2 Simulation Results .................................................. 33

# 8 Conclusions and Future Work

8.1 Conclusions .......................................................... 37  
8.2 Future Work .......................................................... 38

# A The Kronecker Product

A.1 The vec Operator .................................................... 39  
A.2 The Kronecker Product, Definition ............................... 39  
A.3 A Useful Property ................................................... 40

Bibliography .............................................................. 42
Notation

$\mathbb{C}$ The set of complex numbers.
$\mathbb{R}$ The set of real numbers.
$\mathbb{Z}$ The set of integers.
$X, x$ $X$ is a matrix and $x$ is a column vector.
$X^T, X^*$ The transpose and the conjugate transpose of $X$.
$(X)_{ij}$ The $(i, j)$th element of the matrix $X$.
$I_N$ $I_N$ is the $(N \times N)$ identity matrix.
$X \otimes Y$ The Kronecker product of $X$ and $Y$.
$X^\dagger$ The Moore-Penrose pseudo inverse of $X$.
vec$(X)$ The columns of $X$ stacked in a vector.
$E\{X\}$ The expected value of $X$.
$\text{Tr}\{X\}$ Trace of $X$.
$\Re\{x\}, \Im\{x\}$ Real and imaginary part of $x$.
$\mathcal{N}(a, b)$ Normal distribution with mean $a$ and covariance $b$.
$\mathcal{CN}(a, b)$ Circularly symmetric complex normal distribution.
Chapter 1

Introduction

1.1 Background

In order to meet the demands on higher data rates, better quality and availability from an ever increasing number of wireless subscribers, new techniques in signal processing and coding need to be developed and implemented. To achieve high data rate communication, the system has to overcome problems such as additive noise and channel fading. One way is to make several replicas of the signal available to the receiver with the hope that at least some of them are not severely attenuated. This technique is called diversity [1]. Examples of diversity techniques include time diversity, frequency diversity and space diversity. As the available bandwidth is finite, space diversity schemes seem promising, since they do not involve any loss of bandwidth. The use of multiple antennas at both ends of a wireless link is illustrated in Figure 1.1. Such a multiple-input-multiple-output (MIMO) system promises significant improvements in terms of spectral efficiency, link reliability and also improves the system capacity [2] compared to conventional systems.

![Figure 1.1: A multiple input-multiple output system.](image-url)

The so-called Vertical Bell Laboratories Layered Space-Time (V-
BLAST) architecture is an example of the space diversity scheme, which increases the capacity of the system. Other coding schemes for MIMO systems which combine space diversity and time diversity are called Space-Time Codes (STC). Space-Time Block Codes (STBC) and Space-Time Trellis Codes (STTC) are examples of space-time coding techniques. In this work, we focus our efforts on V-BLAST and Space-Time Block Codes, since both schemes can be easily written in linear form, as will be apparent from the development to follow.

Decoding techniques for Space-Time Codes include zero-forcing (ZF), minimum mean-square error (MMSE), maximum likelihood (ML) decoder and V-BLAST. Among these techniques, maximum likelihood decoding yields the best performance over the other detectors [4]. But the maximum likelihood decoder is often considered to be practically infeasible due to high computational complexity in MIMO systems with large number of antennas and high-order constellations. To lower the complexity, a new type of detection method called sphere decoding [5] can be used. The sphere decoding algorithm has near ML performance with reasonably low complexity.

1.2 Objective

This master’s project aims at evaluating the sphere decoding algorithm for a multiple antenna communication system in a flat fading channel. The coding schemes to be investigated include V-BLAST and Space-Time Block Codes. The performance of the sphere decoding algorithm is compared with three different decoders including zero-forcing, V-BLAST and the optimal maximum likelihood decoder.

1.3 Previous Works

Many of the practical space-time schemes are designed to achieve high data rates, such as V-BLAST [3], orthogonal designs [6], and linear dispersion codes [7]. All these techniques are designed to have simple symbol detection at the receiver. The decoding algorithms used in these papers are maximum likelihood or nulling and cancellation. The maximum likelihood decoder has high decoding complexity and nulling and cancellation may lead to low performance.

1.4 Thesis Outline

In Chapter 2, the system model is presented. The space-time coding methods used in this work are introduced in Chapter 3. In Chapter 4, we give an overview of various conventional decoders we used for simulations in this work. Chapter 5 introduces the sphere decoding algorithm, which is the
main focus of this work. The performance of the sphere decoder is compared with various decoders in Chapter 6. We follow up with investigating the impact of the channel estimation errors introduced in the system in Chapter 7. Finally conclusions and future works are found Chapter 8.

1.5 Acronyms

AWGN Additive White Gaussian Noise
BER Bit Error Rate
CSI Channel State Information
IID Independent and Identically Distributed
LD Linear Dispersion
MIMO Multiple Input-Multiple Output
ML Maximum Likelihood
MMSE Minimum Mean Square Error
PEP Pairwise Error Performance
PSK Phase Shift Keying
QAM Quadrature Amplitude Modulation
SD Sphere Decoder
SNR Signal to Noise Ratio
STBC Space-Time Block Codes
STC Space-Time Codes
STTC Space-Time Trellis Codes
V-BLAST Vertical Bell Labs Layered Space-Time
ZF Zero-Forcing
Chapter 2
System Model

In this work, we focus on a single-user communication link where the transmitter and the receiver is equipped with $M$ and $N$ antennas, as illustrated in Figure 2.1. In this chapter, the data models and preliminaries to be later used are presented.

Figure 2.1: Block diagram of a wireless communication system.

2.1 Channel Model

Channel modeling has always been an important area in wireless communication. More about the modeling of MIMO Radio Propagation Channels can be found in [8]. For simplicity, the system modeled in this project is a narrow-band, flat fading, multiple antenna communication system. Slow channel fading is assumed, meaning that the fading coefficients are constant during the transmission of a block of $L$ symbols, but may vary from one block to another in a random manner. The bandwidth of the transmitted signal is much less than the coherence bandwidth of the channel. Consequently the different frequency components of the transmitted signal undergo the same attenuation and phase shift when propagating through the channel.
In the time domain, flat fading corresponds to a channel delay spread which is much less than the symbol time, therefore the channel assumed in this work does not suffer from inter-symbol interference (ISI).

### 2.2 Data Model

The input data is modulated into information symbols using an arbitrary constellation, as for instance the one in Figure 2.2. However, in this work we limit our attention to symbols that are output from a PAM or QAM modulator.

![16-QAM scheme using 16 different symbols.](image)

Figure 2.2: 16-QAM scheme using 16 different symbols.

The symbols are thereafter divided into blocks \( s(k) = [s_1 \ s_2 \ \cdots \ s_m]^T \), where \( m \geq 1 \). As illustrated in Figure 2.3 the symbol block \( s(k) \) is encoded using a space-time encoder into \( M \) substreams, where each substream is transmitted on a dedicated transmitter antenna. The transmitted signals at the transmit antennas are represented by a vector \( c(n) \), where

\[
c(n) = [c_1(n) \ c_2(n) \ \cdots \ c_M(n)]^T. \tag{2.1}
\]

![Diagram for Space-Time Encoder](image)

Figure 2.3: Diagram for Space-Time Encoder
Each receiver antenna responds to each transmitter antenna through a fading channel coefficient. All the channel coefficients are assumed to be independent and identically distributed (IID). The received signals are corrupted by additive noise that is statistically independent among the $N$ receiver antennas and the symbol periods. The received signal $r_i(n)$ at the $i$th antenna illustrated in Figure 2.4 can be written as

$$r_i(n) = \sum_{j=1}^{M} h_{i,j} c_j(n) + v_i(n), \quad (2.2)$$

where $c_j(n)$ is the signal transmitted from the $j$th antenna, $h_{i,j}$ denotes the fading coefficient between transmitter $j$ and receiver $i$ and $v_i(n)$ is the noise.

For all elements of the receiver antennas, the input-output relation can be written as

$$\begin{pmatrix} r_1(n) \\ r_2(n) \\ \vdots \\ r_N(n) \end{pmatrix} = \begin{pmatrix} h_{1,1} & h_{1,2} & \ldots & h_{1,M} \\ h_{2,1} & h_{2,2} & \ldots & h_{2,M} \\ \vdots & \vdots & \ddots & \vdots \\ h_{N,1} & h_{N,2} & \ldots & h_{N,M} \end{pmatrix} \begin{pmatrix} c_1(n) \\ c_2(n) \\ \vdots \\ c_M(n) \end{pmatrix} + \begin{pmatrix} v_1(n) \\ v_2(n) \\ \vdots \\ v_N(n) \end{pmatrix}, \quad (2.3)$$

or more compactly as

$$r(n) = Hc(n) + v(n), \quad (2.4)$$

where $r(n) \in \mathbb{C}^N$ denotes the complex-valued received signal vector and $v(n) \in \mathbb{C}^N$ is the noise term modeled as spatially and temporally white, zero-mean complex Gaussian with variance $2\sigma_v^2$.

$H$ is an $N \times M$ channel matrix. Each coefficient of $H$ is assumed to be $\mathcal{CN}(0,1)$ (zero mean, unit-variance and complex Gaussian). The channel matrix is also known as the channel state information (CSI). At the receiver, the CSI can be obtained by channel estimation based on the transmission of a training sequence. In this work, a blind transmitter is assumed, meaning
that there is no feedback information transmitted from the receiver to the transmitter. We will also assume that the receiver has perfect knowledge of the CSI at all time.
Chapter 3

Space-Time Coding

The transmission is done in bursts of length $L$ symbols over the fading channel, which changes in a random fashion every $L$th sample. Consider without loss of generality one such burst the time interval $n = 0, 1, \cdots, L-1$. From (2.4) it follows that the corresponding received burst can be written as

$$ R = HC + V, $$

where

$$ R = [r(0) \ r(1) \ \cdots \ r(L - 1)] $$
$$ C = [c(0) \ c(1) \ \cdots \ c(L - 1)] $$
$$ V = [v(0) \ v(1) \ \cdots \ v(L - 1)]. $$

The matrix $C$ is sometimes called a (space-time) codeword, and by proper design, diversity gain, multiplexing gain and/or coding gain can be achieved. In the sections to follow, we will examine Space-Time Coding (STC) more closely.

3.1 Overview

Part of the reason Space-Time Codes give good performance is that they provide the system with transmit diversity. Diversity is obtained by transmitting a signal on several antennas simultaneously (space) and several symbol periods (time), which leads to an increase in diversity without loss of bandwidth. In addition to the spatial diversity, some of the schemes also provide additional coding gain. Many of the practical Space-Time Code schemes that achieve these high capacities, such as spatial multiplexing [3] and Space-Time Block Coded (STBC) systems are designed so as to allow simple symbol detection at the receiver because they map the symbols linearly to the transmitter antennas.
3.2 V-BLAST

Illustrated in Figure 3.1, spatial multiplexing or V-BLAST (Vertical Bell Labs Layered Space-Time) [3] is proposed as a multiple antenna system. The V-BLAST architecture breaks the original data stream into \( M \) substreams that are transmitted on the individual antennas without intercoding among the transmitter antennas. If we for simplicity assume that the substreams are uncoded, the transmission can be seen as using a block code of length one. The corresponding codeword can be written as

\[
C = \begin{bmatrix}
s_1 \\
s_2 \\
\vdots \\
s_M
\end{bmatrix}.
\]

(3.2)

![Figure 3.1: Block diagram of V-BLAST structure with \( M = 3, N = 3 \).](image)

The V-BLAST detector decodes the substreams using a sequence of nulling and cancellation steps. An estimate of the strongest transmitted signal is obtained by nulling out all the weaker transmit signals using the zero forcing criterion, then subtract this strongest signal from the received signal, proceed to decode the strongest signal of the remaining transmitted signals, and so on.

3.3 The Alamouti Code

A simple Space-Time Block Code scheme is the famous Alamouti code [9]. As shown in Figure 3.2, the Alamouti scheme is designed for two transmit antennas. At time instant \( n \), information symbol \( s_1 \) is transmitted using the first antenna and simultaneously \( s_2 \) is transmitted through the second antenna. At time instant \( n + 1 \), \( -s_2^* \) is transmitted at the first antenna and \( s_1^* \) though the second antenna, where \((\cdot)^*\) denotes the complex conjugate.
transpose. The corresponding codeword can be written as

\[
C = \begin{bmatrix} c(n) & c(n+1) \end{bmatrix}
= \begin{bmatrix} s_1 & -s_2^* \\ s_2 & s_1^* \end{bmatrix}.
\] (3.3)

The design of the Alamouti code has given the codeword a special structure, where independent from the information symbols, the codeword is orthogonal

\[
CC^* = (s_1^2 + s_2^2)I_2.
\] (3.4)

The received signal \( R \) for a \( 2 \times 2 \) MIMO system becomes

\[
\begin{align*}
r_1(0) &= h_{1,1}s_1 + h_{1,2}s_2 + v_1, \\
r_2(0) &= h_{2,1}s_1 + h_{2,2}s_2 + v_2, \\
r_1(1) &= -h_{1,1}s_2^* + h_{1,2}s_1^* + v_3, \\
r_2(1) &= -h_{2,1}s_2^* + h_{2,2}s_1^* + v_4,
\end{align*}
\]

where \( r_i(n) \) represents the output from the \( i \)th antenna at time instance \( n \) and \( v_k \) is zero-mean complex Gaussian noise. The Alamouti decoder builds two signals:

\[
\begin{align*}
\tilde{s}_1 &= h_{1,1}^* r_1(0) + h_{1,2}^* r_1^*(1) + h_{1,2}^* r_2(0) + h_{2,2}^* r_2^*(1) \\
\tilde{s}_2 &= h_{1,2}^* r_1(0) - h_{1,1}^* r_1^*(1) + h_{2,2}^* r_2(0) - h_{1,2}^* r_2^*(1)
\end{align*}
\]

Substituting the equations we have

\[
\begin{align*}
\tilde{s}_1 &= (|h_{1,1}|^2 + |h_{1,2}|^2 + |h_{2,1}|^2 + |h_{2,2}|^2)s_1 + e_1, \\
\tilde{s}_2 &= (|h_{1,1}|^2 + |h_{1,2}|^2 + |h_{2,1}|^2 + |h_{2,2}|^2)s_2 + e_2,
\end{align*}
\]

where

\[
\begin{align*}
e_1 &= h_{1,1}^* v_1 + h_{1,2}^* v_2 + h_{2,1}^* v_3 + h_{2,2} v_4, \\
e_2 &= -h_{1,1} v_1^* + h_{1,2} v_2^* - h_{2,1} v_3^* + h_{2,2} v_4^*.
\end{align*}
\]

These combined signals \( \tilde{s}_1 \) and \( \tilde{s}_2 \) are then sent to the maximum likelihood decoder.

---

Figure 3.2: Block diagram of an Alamouti system.
3.4 Orthogonal Space-Time Block Codes

The Alamouti code is one example of Space-Time Block Codes with orthogonal design. The designed codeword has either row or column orthogonality independent from the encoded symbols. For instance, codes designed for three and four transmit antennas are given by

\[
C = \begin{bmatrix}
  s_1 & -s_2 & -s_3 & -s_4 & s_1^* & -s_2^* & -s_3^* & -s_4^*
s_2 & s_1 & s_4 & -s_3 & s_2^* & s_1^* & s_4^* & -s_3^*
s_3 & -s_4 & s_1 & s_2 & -s_4^* & -s_3^* & s_1^* & s_2^*
s_4 & s_3 & -s_2 & s_1 & s_4^* & -s_3^* & -s_2^* & s_1^* 
\end{bmatrix}
\] (3.5)

and

\[
C = \begin{bmatrix}
  s_1 & -s_2 & -s_3 & -s_4 & s_1^* & -s_2^* & -s_3^* & -s_4^*
s_2 & s_1 & s_4 & -s_3 & s_2^* & s_1^* & s_4^* & -s_3^*
s_3 & -s_4 & s_1 & s_2 & -s_4^* & -s_3^* & s_1^* & s_2^*
s_4 & s_3 & -s_2 & s_1 & s_4^* & -s_3^* & -s_2^* & s_1^* 
\end{bmatrix}
\] (3.6)

More generalized orthogonal designs of Space-Time Block Codes for any number of transmit antennas can be found in [6], where the codes were shown to have very simple detectors. These codes are generally designed to optimize a raw block or bit pairwise error performance (PEP) criteria. PEP is a common measure for error probability, since to determine the codeword error probability in general is difficult. PEP is the probability that a transmitted codeword \( C_1 \) is decoded as \( C_2 \) in the receiver. As the number of the antennas at both ends increase, it is hard to design the encoder with complete orthogonality.

3.5 Linear Dispersive Codes

Another category of the space-time codes is the class of the linear dispersive codes [7]. The information symbols \( s = [s_1, \cdots, s_m]^T, m \geq 1 \), where the symbol \( s_i(i = 1, \cdots, m) \) is chosen from a constellation such as QAM or PSK. We can represent each symbol as

\[
s_i = \alpha_i + j \beta_i,
\]

where \( j \) denotes the imaginary unit. The codewords of the linear dispersive codes are formed as

\[
C = \sum_{q=1}^{Q} A_q x_q,
\] (3.7)

where \( \{A_q\} \) represents the weight matrices that define the code, while \( x_q \in \mathbb{R} \) is either the real (\( \alpha_i \)) or the imaginary (\( \beta_i \)) part of the information symbols \( s_i \). Note that the weight matrices are known to the receiver. The V-BLAST
technique is the simplest example of such linear codes. Each incoming symbol is transmitted once on one antenna at one symbol time. For example, for $M = 2$, the codeword

$$ C = \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} $$

(3.8)
can be represented as follows

$$ A_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, $$

$$ A_3 = \begin{bmatrix} j \\ 0 \end{bmatrix}, \quad A_4 = \begin{bmatrix} 0 \\ j \end{bmatrix}, $$

and

$$ x_1 = \Re\{s_1\}, \quad x_2 = \Re\{s_2\}, $$

$$ x_3 = \Im\{s_1\}, \quad x_3 = \Im\{s_2\}, $$

where $\Re\{\cdot\}$ and $\Im\{\cdot\}$ extract the real and imaginary part of their arguments, respectively.

Orthogonal Space-Time Block Codes can also be written in linear form in a similar fashion. For example from (3.3) it is seen that the weight matrices in the Alamouti code can be written as

$$ A_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}, $$

$$ A_3 = \begin{bmatrix} j & 0 \\ 0 & -j \end{bmatrix}, \quad A_4 = \begin{bmatrix} 0 & j \\ j & 0 \end{bmatrix}, $$

with the real-valued symbols defined as

$$ x_1 = \Re\{s_1\}, \quad x_2 = \Re\{s_2\}, $$

$$ x_3 = \Im\{s_1\}, \quad x_4 = \Im\{s_2\}. $$

From (3.7) it follows that the codeword $C$ can be rewritten as

$$ \text{vec}(C) = W x, $$

(3.9)

where

$$ W = \begin{bmatrix} \vdots & \vdots & \vdots & \vdots \\ \text{vec}(A_1) & \text{vec}(A_2) & \cdots & \text{vec}(A_Q) \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} $$

(3.10)

and

$$ x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_Q \end{bmatrix}^T. $$

(3.11)
Then we can rewrite (3.1) in the form of a linear model using properties of the Kronecker product and vec operator

\[
y = \text{vec}(Y) \\
= \text{vec}(HC) + \text{vec}(V) \\
= \text{vec}(H C I_L) + \text{vec}(V) \\
= (I_L \otimes H) \text{vec}(C) + \text{vec}(V) \\
= (I_L \otimes H)W x + \text{vec}(V) .
\] (3.12)

\( \otimes \) denotes the Kronecker product. An introduction of the Kronecker product and vec operator can be found in Appendix A. By defining the matrix \( A \) as

\[
A = (I_L \otimes H)W
\] (3.13)

and

\[
b = \text{vec}(V) ,
\] (3.14)

we can simplify (3.12) as

\[
y = Ax + b .
\] (3.15)

In [7], it is pointed out that linear dispersive codes are simple to encode and decode, and apply to any combination of transmit and receive antennas. In the paper, the codes are decoded by either maximum likelihood or by nulling and cancellation. Since the V-BLAST and some orthogonal designed codes can be written as special case of linear dispersive codes, we are using the linear dispersive codes in the simulations.
Chapter 4

Conventional Decoder for Space-Time Coded System

Various conventional decoders can be applied to decode the signals in the MIMO system. Examples of the decoders include zero-forcing (ZF), V-BLAST, minimum mean-square error (MMSE) and maximum likelihood (ML).

4.1 The Zero Forcing Decoder

A simple linear decoder is the zero-forcing (ZF) decoder, which assuming that $A$ is invertible, obtains an estimate of the vector $x$ is obtained as

$$\hat{x} = A^\dagger y.$$ (4.1)

where $A^\dagger$ denotes the Moore-Penrose pseudo-inverse of the matrix $A$, and is defined as $A^\dagger = (A^TA)^{-1}A^T$. For illconditioned $A$, the ZF decoder performs well only in the high SNR region, where in the low SNR region it will be significant noise enhancement [10].

4.2 The Minimum Mean-Square Error Decoder

The minimum mean-square error (MMSE) decoder minimizes the overall error due to the noise and mutual interference between the signals from the different transmit antennas. Compare with ZF, the MMSE decoder is less sensitive to noise at the cost of reduced signal separation quality. The codeword $C$ is decoded using [4]

$$\hat{x} = \frac{1}{M} A^T (\sigma_n^2 I_N + \frac{1}{M} A A^T)^{-1} y.$$ (4.2)
4.3 Decoding in V-BLAST

The V-BLAST decoder was proposed in [11] as a simple decoder for the BLAST structure. The detection is done by using conventional adaptive antenna array techniques, i.e. linear combination and nulling.

We can use the matrix $A^\dagger$ to estimate the "strongest" signal $x_i$ in the received vector. A nulling vector $w_i$ is formed by taking the $i$th row of $A^\dagger$. The choice of the row $i$ is determined by computing norm of each column in $A$. The "strongest" signal $\hat{x}_i$ is then estimated as

$$\hat{x}_i = w_i^\top y.$$ (4.3)

Thereafter, the "strongest" signal $\hat{x}_i$ is quantized to find the closest point in the constellation. Assuming that $\hat{x}_i = x_i$, the symbol $x_i$ is cancelled from the received vector $y$, resulting in a modified received vector $y_2$:

$$y_2 = y - \hat{x}_i(A)_i$$ (4.4)

where $(A)_i$ denotes the $i$th column of $A$. These steps are repeated until all components are detected.

4.4 The Maximum Likelihood Decoder

Among the mentioned techniques, the maximum likelihood decoder yields the best performance. The performance of Space-Time Codes for a large number of antennas is studied using three different decoders: ML, MMSE and ZF in [12]. We assume that the transmitter chooses between the codewords $C_1, C_2, \ldots, C_Q$, where each codeword $C_i$ is uniquely defined by its information symbols $s$ and in linear dispersive code representation from (3.9), (3.13) and (3.15), the derivation of the ML detector begins by first forming a hypothesis test as follows

$$C_1 : y = Ax_1 + b$$
$$C_2 : y = Ax_2 + b$$
$$C_3 : y = Ax_3 + b$$
$$\vdots$$
$$C_Q : y = Ax_Q + b$$

where the signal $x_i$ are assumed to have equal a priori probabilities. We know that

$$\|y - Ax\|^2 = (y - Ax)^\ast (y - Ax)$$
$$= y^\ast y + x^\ast A^\ast Ax - y^\ast Ax - x^\ast A^\ast y$$
$$= y^\ast y + x^\ast A^\ast Ax - 2 \Re\{y^\ast Ax\}.$$ (4.5)
The maximum likelihood decoder computes the estimate $\hat{x}$ according to

$$\hat{x} = \arg \min_{x} \|y - Ax\|^2 = x^* A^* Ax - 2 \Re \{y^* Ax\}.$$  \hspace{1cm} (4.6)

where the minimization is performed over all possible $x$.

If the size of the scalar constellation used in each component of $x$ is $D$ (e.g. $D = 2$ for BPSK) and $Q$ transmitter antennas, the decoder has to perform an enumeration over a set of size $D^Q$. For higher-order modulation such as 64-QAM this complexity can become prohibitive even for a small number of transmit antennas.
Chapter 5

Sphere Decoding

Maximum-likelihood decoding of a random code over an additive white Gaussian noise channel requires an exhaustive search over all the possible codeword, and so the computational complexity of the optimal decoding scheme is exponential in the length of the codeword. A new type of the detection technique called the sphere decoding algorithm [13] is proposed to lower the computational complexity. The principle of the sphere decoding algorithm is to search the closest lattice point to the received signal within a sphere radius, where each codeword is represented by a lattice point in a lattice field [14]. In a two-dimension problem illustrated in Figure 5.1, one can easily restrict the search by drawing a circle around the received signal just enough to enclose one lattice point and eliminate the search of all the points outside the circle.

Figure 5.1: Geometrical representation of the sphere decoding algorithm.
5.1 Sphere Decoding Algorithm

The sphere decoding algorithm is a simple way to find the solution for $x$ as

$$y = Ax + b.$$  \hfill (5.1)

A lattice can be defined by having each entry of the vector $x$ defined in (3.11). The dimension of the lattice can be defined as

$$S \in D^Q$$ \hfill (5.2)

where $D$ defined by the modulation method used to encoded the information symbols

$$4 - \text{QAM} : D \in \{-1, +1\}$$

$$16 - \text{QAM} : D \in \{-3, -1, +1, +3\}.$$  \hfill (5.3)

The maximum likelihood decoding algorithm is given by

$$\|b\|^2 = \|y - Ax\|^2 = (x - \hat{x})^* A^* A (x - \hat{x}) + \|y\|^2 - \|A \hat{x}\|^2,$$ \hfill (5.4)

where $\hat{x}$ is the unconstrained maximum likelihood estimate of $x$, and defined as $\hat{x} = A^T y = (A^T A)^{-1} A^T y$. We can rewrite the maximum likelihood metric in (4.5) as follows

$$\hat{x}_{ML} = \arg \min_{x \in S} \|y - Ax\|^2$$

$$= \arg \min_{x \in S} (x - \hat{x})^* A^* A (x - \hat{x}).$$ \hfill (5.5)

Based on Fincke Pohst method in [5], a lattice point which lies inside the sphere with radius $d$ has to fulfill the condition

$$d^2 \geq \|y - Ax\|^2 = (x - \hat{x})^* A^* A (x - \hat{x}) + \|y\|^2 - \|A \hat{x}\|^2.$$ \hfill (5.6)

A new constant can be defined as $d'^2 = d^2 - \|y\|^2 - \|A \hat{x}\|^2$, then (5.6) can be rewritten as

$$d'^2 \geq (x - \hat{x})^* A^* A (x - \hat{x})$$ \hfill (5.7)

The matrix $A^* A$ can be decomposed to triangular matrices with Cholesky decomposition ($A^* A = U^* U$), where $U$ is an upper triangular matrix

$$U = \begin{bmatrix}
  u_{1,1} & u_{1,2} & u_{1,3} & \cdots & u_{1,m} \\
  0 & u_{2,2} & u_{2,3} & \cdots & u_{2,m} \\
  0 & 0 & u_{3,3} & \cdots & u_{3,m} \\
  0 & 0 & \ddots & \ddots & \ddots \\
  0 & 0 & 0 & \ddots & u_{n,m}
\end{bmatrix},$$ \hfill (5.8)
where \( u_{i,j} \) denotes element \((i,j)\) of the matrix \( U \). Further simplification of (5.7) gives
\[
d'2 \geq (x - \hat{x})^* A^* A (x - \hat{x})
= (x - \hat{x})^* U^* U (x - \hat{x})
= \sum_{i=1}^{Q} u_{i,i}^2 ((x_i - \hat{x}_i) + \sum_{j=i+1}^{Q} \frac{u_{i,j}}{u_{i,i}} (x_j - \hat{x}_j))^2
\]
(5.9)

Because of the upper triangular nature of \( U \), one can begin evaluation of the last element in \( x \) as
\[
\begin{align*}
    u_{Q,Q}^2 (x_Q - \hat{x}_Q)^2 \leq d'^2
\end{align*}
\]
(5.10)

which leads to
\[
\begin{align*}
    \left[ \hat{x}_Q - \frac{d'}{u_{Q,Q}} \right] \leq x_Q \leq \left[ \hat{x}_Q + \frac{d'}{u_{Q,Q}} \right].
\end{align*}
\]
(5.11)

where \([a]\) rounds to the nearest symbol value greater than or equal to a. Similarly \([a]\) rounds to the nearest symbol value less than or equal to a.

The method employs an iterative search, for every \( x_Q \) satisfying (5.11), \( d'^2_{Q-1} = d'^2 - u_{Q,Q}^2 (x_Q - \hat{x}_Q)^2 \) can be defined, and a new condition can be written as
\[
\begin{align*}
    u_{Q-1,Q-1}^2 \left( x_{Q-1} - \hat{x}_{Q-1} + \frac{u_{Q-1,Q}}{u_{Q-1,Q-1}} (x_Q - \hat{x}_Q) \right)^2 \leq d'^2_{Q-1},
\end{align*}
\]
(5.12)

which is equivalent to
\[
\begin{align*}
    \left[ \hat{x}_{Q-1} - \frac{d'_{Q-1}}{u_{Q-1,Q-1}} \right] \leq x_{Q-1} \leq \left[ \hat{x}_{Q-1} + \frac{d'_{Q-1}}{u_{Q-1,Q-1}} \right].
\end{align*}
\]
(5.13)

In a similar fashion, one proceeds for \( x_{Q-2} \), and so on, stating nested necessary conditions for all elements of \( x \).

To ensure that the lattice point is inside the sphere, the initial radius must be big enough to enclose at least one lattice point. One method is to use statistical properties of the signal. Note that \( \|b\|^2/\sigma_v^2 \) has a Chi-Square distribution [15] with \( 2NL \) degrees of freedom, since each entry of the noise is an independent \( \mathcal{N}(0, \sigma_v^2) \) random variable. The probability of the lattice point inside the sphere can be written as
\[
\begin{align*}
    \Pr[\|b\|^2 \leq d'^2] = \Pr\left[ \frac{\|b\|^2}{\sigma_v^2} \leq \frac{d'^2}{\sigma_v^2} \right]
    = \int_{0}^{d'^2/\sigma_v^2} \frac{\lambda^{NL-1}}{2^{NL} \Gamma(NL)} e^{-\lambda/2} d\lambda,
\end{align*}
\]
(5.14)
where
\[ \Gamma(NL) = \int_0^\infty t^{NL-1} e^{-t} dt. \]

Therefore, one possible choice of the radius \(d\) is as a linear function of the variance of \(\|\mathbf{b}\|^2\)
\[ d^2 = 2\alpha NL \sigma_\nu^2. \quad (5.15) \]
and \(\alpha \geq 1\) chosen so that we can be sure that a solution is inside the sphere.

The complexity of the sphere decoding algorithm has been shown in [5] to be of order
\[ O\left(n^2 \times \left(1 + \frac{n - 1}{4\lambda^{-1}d^2}\right)^{4\lambda^{-1}d^2}\right), \quad (5.16) \]
where \(n\) is the lattice dimension, and \(d\) is the radius of the sphere, while \(\lambda\) is a lower bound for the eigenvalues of the matrix \(\mathbf{UU}^*\).

During the implementation of the sphere decoding algorithm, we did one modification of the original algorithm which further decreases the complexity. When a lattice point is found inside the sphere, the radius \(d\) to the point is updated to the new value. Also the new upper bounds are updated, which leads to less lattice points inside the sphere. If the solution is the closest lattice point, there cannot be any other lattice point inside the sphere.

### 5.2 Implementation

The following steps shows the implementation of the sphere decoding algorithm

**Input:** \(\mathbf{R}, \mathbf{y}, \hat{\mathbf{x}}, d, \hat{\mathbf{s}}\),

1. Set \(k = Q\),
   \[ d_Q^2 = d^2 - \|\mathbf{y}\|^2 + \|\mathbf{A}\hat{\mathbf{x}}\|^2, \]
   \[ \hat{\mathbf{x}}_{Q+1} = \hat{\mathbf{x}}_Q. \]
2. (Set bound for \(x_k\))
   \[ z = \frac{d_k}{\mathbf{u}_{kk}}; \]
   \[ UB(x_k) = [z + \hat{x}_{k+1}], \]
   \[ x_k = [-z + \hat{x}_{k+1}] - s. \]
3. (Increase \(x_k\))
   \[ x_k = x_k + s. \]
   If \(x_k \leq UB(x_k)\) go to 5.
4. \(k = k + 1;\)
   if \(k = Q + 1\), terminate algorithm, else go to 3.
5. (Increase k) \(k = 1\) go to 6.
   Else \(k = k - 1,\)
   \[ \hat{x}_{k+1} = \hat{x}_k + \sum_{j=k+1}^{Q} \frac{u_{kj}}{u_{kk}} (x_j - \hat{x}_j), \]
   \[ d_k^2 = d_{k+1}^2 - u_{k+1,k+1}^2 (x_{k+1} - x_{k+1,k+2})^2, \]
   and go to 2.

\(^1\)distance between two closest lattice points
6. Solution found. Save $\mathbf{x}$ and go to 3.

Illustrated in Figure 5.2, we modified the original sphere decoding algorithm. When a lattice point is found inside the sphere, the search radius is updated. Also the upper bounds are recalculated to limit the number of lattice points that need to be searched.

$$
\begin{align*}
k &= Q \\
\hat{d}_Q^2 &= d^2 - \|\mathbf{y}\|^2 + \|A\hat{\mathbf{x}}\|^2 \\
\hat{x}_{Q,Q+1} &= \hat{x}_Q \\
\end{align*}
$$

$$
\begin{align*}
z &= \frac{d_k}{u_k,k} \\
UB(x_k) &= [z + \hat{x}_{k|k+1}] \\
x_k &= [-z + \hat{x}_{k|k+1}] - s \\
\end{align*}
$$

$$
\begin{align*}
k &= k + 1 \\
x_k &= x_k + s \\
\end{align*}
$$

$$
\begin{align*}
k &= Q \\
x_k &\leq UB(x_k) \\
\end{align*}
$$

Terminate algorithm

$$
\begin{align*}
k &= 1 \\
k &= k - 1 \\
\hat{x}_{k'|k+1} &= \hat{x}_k + \sum_{j=k+1}^{Q} u_{k,j} (x_j - \hat{x}_j) \\
d_{k|k+1}^2 &= d_{k+1}^2 - u_{k+1,k+1}^2 (x_{k+1} - \hat{x}_{k+1|k+2})^2 \\
\end{align*}
$$

Figure 5.2: Flowchart of the sphere decoding algorithm, $s$ is distance between constellation points.
Chapter 6

Simulation Results

In this chapter, some simulation results are presented. In order to compare the performance of the sphere decoding algorithm to other decoding algorithms, we use the same definition of the signal to noise ratio (SNR) in all the simulations. The SNR is defined as

$$\text{SNR} = \frac{E\{\|HC\|^2\}}{E\{\|V\|^2\}}$$

$$= \frac{E\{\text{Tr}\{HCC^*H^*\}\}}{E\{\text{Tr}\{VV^*\}\}}$$

$$= \frac{\text{Tr}\{E\{H^*H\}\}E\{CC^*\}\}}{E\{\text{Tr}\{VV^*\}\}}.$$  \hspace{1cm} (6.1)

Since $H$ and $C$ are assumed to be statistically independent. Note that

$$E\{H^*H\} = NI_M$$  \hspace{1cm} (6.2)

and

$$E\{CC^*\} = E\left\{\sum_{q=1}^{Q} A_q x_q \sum_{q' = 1}^{Q} (A_{q'} x_{q'})^*\right\}$$

$$= E\left\{\sum_{q=1}^{Q} \sum_{q' = 1}^{Q} (A_q A_{q'}^*)(x_q x_{q'})\right\}$$

$$= \sum_{q=1}^{Q} (A_q A_q^*) E\{x_q x_q^*\}$$

$$= \sum_{q=1}^{Q} (A_q A_q^*) E\{x_q^2\}.$$  \hspace{1cm} (6.3)

The weight matrices $\{A_q\}$ are normalized so that $E\{\text{Tr}\{CC^*\}\} = L$. Recall that each entry of $V$ is an independent $\mathcal{N}(0, \sigma_v^2)$ random variable, which
follows that the expected noise variance is
\[ E \operatorname{Tr}\{VV^*\} = NL\sigma_v^2. \]  
(6.4)
The SNR definition in (6.1) can be simplified as
\[ \text{SNR} = \frac{1}{\sigma_v^2}. \]  
(6.5)

6.1 Fixed Radius versus Adaptive Radius

The sphere decoding algorithm in [13] depends on the initial search radius. If the initial search radius is too large, there will be too many lattice points in the sphere and we may still require an exponential search; if the radius is too small there will be no points in the sphere. We compared the performance of the original sphere decoding algorithm with the initial radius defined in (5.15), against the modified version of the algorithm. For this simulation, we are using the V-BLAST scheme with \( M = 2 \) and \( N = 2 \). The transmitted codeword is modulated using a 16-QAM modulator. Illustrated in Figure 6.1, the performance of the modified version of the sphere decoding algorithm is the same as the original version, but its computational complexity shown in Figure 6.2 is much lower then the original and the maximum likelihood decoder. FLOPS in the figure is calculated using FLOPS function in MATLAB, which counts the approximated floating point operations that algorithm needed to complete the task.

As seem from the figure, the sphere decoding algorithm with fixed search radius seems to have low decoding complexity in the low SNR region. But this is because the algorithm cannot find any lattice points inside the sphere, and the search is prematurely abandoned. As the SNR increase in the simulation, the initial search radius becomes smaller and smaller, which leads to lower decoding complexity.
Figure 6.1: Comparison of the performance between maximum likelihood decoder, the sphere decoding algorithm with fixed initial radius and the modified sphere decoding algorithm.

Figure 6.2: Comparison of decoding complexities.
6.2 V-BLAST

In this simulation, we compared the performance of the sphere decoding algorithm against other decoders including V-BLAST, zero-forcing and the maximum likelihood detection algorithm in V-BLAST scheme, where $M = 3$ and $N = 3$. Both the sphere decoders and the maximum likelihood decoder outperform the V-BLAST decoder as well as zero-forcing decoder, as shown in Figure 6.3.

![Figure 6.3: The performance of the decoding algorithms in V-BLAST scheme.](image)

Illustrated in Figure 6.4, the sphere decoding algorithm has lower computational complexity than the maximum-likelihood decoding algorithm.

In Figure 6.5, we change the number of transmitter and receiver antennas from $2 \times 2$, up to $5 \times 5$, the complexities of the sphere decoder with fixed initial radius are much lower than the maximum likelihood decoder even in the low SNR region.
Figure 6.4: Computing complexity in the V-BLAST scheme.

Figure 6.5: Complexity as function of transmitter and receiver antennas.
6.3 Alamouti Code

The Alamouti code has already an efficient decoding algorithm. So in this case, the performance of the sphere decoder is compared with a $2 \times 2$ Alamouti code scheme using exhaustive maximum likelihood search algorithm. As illustrated in Figure 6.6 that all the decoders have almost the same bit error probabilities. The computational complexities of the sphere decoding algorithms are lower than the maximum likelihood decoder as shown Figure 6.7.

![Figure 6.6: Bit error probability of Alamouti code.](image1)

![Figure 6.7: Computing complexity of Alamouti code.](image2)
6.4 Linear Dispersive Codes

As the number of transmitter antennas increases, it is hard to keep a complete orthogonal design of the Space-Time Block Codes. We used linear dispersive codes as in (3.7). The weighting matrices \( \{A_q\} \) in the code were obtained from the work in [16]. For this simulation, we simulated a MIMO system with 8 transmitter antennas and 8 receiver antennas, 12 binary bits is encoded in each codeword. The performance of the sphere decoder is the same as the maximum likelihood decoder, but the sphere decoding algorithm has a significantly lower complexity as shown in Figure 6.8.

![Figure 6.8: Computing complexity for linear dispersive codes.](image)

6.5 Random Generated Weighting Matrices

In order to demonstrate the effectiveness of the sphere decoding algorithm, we also used a random generated weight matrices \( \{A_q\} \) in (3.7). In this simulation, we used a 4 × 4 MIMO system, the 16-QAM modulator is used to modulate the input-data and encoded using randomly generated weighting matrices. While the error probability is the same as the maximum likelihood decoder shown in Figure 6.9, the sphere decoder has lower complexity than the maximum likelihood decoder as evident from Figure 6.10.
Figure 6.9: Performance of the decoders with randomly generated weighting matrices, $M = 4$, $N = 4$.

Figure 6.10: Computing complexity for randomly generated weighting matrices, $M = 4$, $N = 4$. 
Chapter 7

Channel Estimation Errors

In previous chapters, we always assumed that we have perfect channel knowledge at the receiver, which allows us to compare the performance of different decoders. However, the channel information is typically not perfect. A channel estimator extracts from the received signal approximate channel coefficients during the transmission. One method to accomplish this is to transmit pilot tones prior to the transmission, by turning off all transmitter antennas except the $i$th antenna at some time instance and sending a pilot signal using $i$th antenna. The fading coefficients $h_{i,j}$ are then estimated. Another way to estimate the channel fading coefficients is to embed the pilot bits inside the signal or send an orthogonal sequence to signals as pilot signals, one from each transmit antenna [17].

7.1 Error Model

The impact from the channel estimation errors will degrade the performance of the system. To study the impact of the channel estimation errors on the sphere decoding algorithm, we introduce the error model at the receiver

$$\hat{H} = \rho H + \sqrt{1 - \rho^2} E$$

(7.1)

where $H$ represent the true channel matrix and $E$ denotes the channel estimation error. The elements of $E$ are assumed to be zero mean, unit-variance and complex Gaussian. Here, $\rho \in [0, 1]$ is a measurement of how accurate the channel estimation is. The value $\rho = 1$ indicates no estimation error at all. When $\rho$ decrease, so does the accuracy of the channel estimation decreases.

7.2 Simulation Results

As shown in Figure [7.1, 7.3, 7.2] and [7.4], the channel estimation errors with different $\rho$ have given the sphere decoding algorithm with adaptive radius
almost the same bit error probability as the maximum-likelihood decoder.

As it is clear from the figures, only in high SNR region, the original sphere decoding algorithm starts to perform poorly. This poor performance is caused by the length of the initial search radius. The search radius in (5.15) depends only on the noise variance. This cause a problem since channel estimation error is the biggest contributor of the errors in the simulation at the high SNR region.

![Figure 7.1: Bit error rate with channel estimation error, $\rho = 0.9$.](image1)

![Figure 7.2: Bit error rate with channel estimation error, $\rho = 0.99$.](image2)
Figure 7.3: Complexity with channel estimation error, $\rho = 0.9$.

Figure 7.4: Complexity with channel estimation error, $\rho = 0.99$. 
Chapter 8

Conclusions and Future Work

8.1 Conclusions

In this master’s thesis work, we have focused on a multiple antenna system in a flat fading environment. We compared the performance of four decoders with different techniques, including zero-forcing, V-BLAST, maximum likelihood and sphere decoding algorithm. The zero-forcing and V-BLAST decoder are simple decoders with low complexities. But these two decoders have low performance compared to the maximum likelihood decoder, which in some sense is the optimal decoder, but it has high decoding complexity.

We used the concept of a linear system model to show that the sphere decoding algorithm offers near maximum likelihood performance with lower decoding complexity. For the original version of the sphere decoding algorithm, the initial radius of the sphere for the search is important to determine, since the complexity depends on the search radius. During the development process, a modified version of the sphere decoding algorithm has investigated, which operate on the assumption that after each lattice point found to be inside the sphere, the radius is decreased to the distance between the best solution and the received signal vector. In general, the sphere decoding algorithms outperform the decoder with V-BLAST and zero-forcing techniques, with lower complexity than the maximum likelihood decoder in high order constellation and large numbers of antenna.

We have also investigated the impact on the performance due to the channel estimation errors. In the computer simulation, the performance of the sphere decoding algorithms are very close to the maximum likelihood decoder. In the high SNR region, it seems that the original version of the sphere decoding algorithm has poor performance. This is because the initial search radius has not taken channel estimation errors into consideration. In a real situation, this case will not be very common, since when noise
variance is small, the channel estimator will be much accurate. With the modified algorithm, the sphere decoding algorithm is independent of the initial search radius, which leads to a high performance algorithm with lower computational complexity compared to the maximum likelihood decoding algorithm.

8.2 Future Work

The theoretical ground of the sphere decoding algorithm has been lay down in [5], and we investigate the performance of this algorithm in data simulation. It is very interesting to investigate the performance of the algorithm in real live application. The complexity study of the algorithm in this work has been based on statistical model. A worst-case scenario has been never studied, this limitation will be more noticeable in the application.

The computational complexity of the sphere decoding algorithm with fixed radius depends on the initial search radius. We have used the noise variance to determine the length of the radius. In the computer simulations with channel estimation errors, it seems that the system has poor performance in the high SNR region. A new method should be investigated to find a new way to determine the initial search radius, which has to take the channel estimation errors into consideration.

Furthermore, a more complex channel model can be studied for this algorithm, such as channel with memories. Then it is interesting to compare its performance with decoders that use decision feedback.
Appendix A

The Kronecker Product

The Kronecker product and the vec operator are introduced in this chapter. They both exhibit many useful properties. In this thesis only a brief introduction is given and for further information the reader is referred to [18].

A.1 The vec Operator

The vec operator transforms a matrix to a vector by stacking its columns on top of each other. If $X$ is a matrix of order $(m \times n)$ and $x_i$ is the $i$th column of $X$ the vec operator is defined by

$$\text{vec} \ X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}.$$  

(A.1)

From (A.1) it immediately follows that vec $X$ is an $mn$ column vector.

One obvious but still useful property of the vec operator is that $\|X\|_F^2 = \|\text{vec} \ X\|_F^2$. It is also obvious that vec $(X + Y) = \text{vec} \ X + \text{vec} \ Y$.

A.2 The Kronecker Product, Definition

The Kronecker product between a matrix $X$ of order $(m \times n)$ and a matrix $Y$ $(o \times p)$ is defined as:

$$X \otimes Y = \begin{bmatrix} x_{11}Y & x_{12}Y & \ldots & x_{1n}Y \\ x_{21}Y & x_{22}Y & \ldots & x_{2n}Y \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1}Y & x_{m2}Y & \ldots & x_{mn}Y \end{bmatrix}.$$  

(A.2)

Where $x_{ij}$ denotes the $i$th, $j$th element of $X$. Hence, $X \otimes Y$ is a matrix of order $(mo \times np)$. 
A.3 A Useful Property

The following property of the Kronecker product and the vec operator is essential to the proposed algorithm:

**Theorem 1** For the matrices $X$, $Y$ and $Z$ the following property holds:

$$\text{vec}(XYZ) = (Z^T \otimes X) \text{vec} Y$$  \hspace{1cm} (A.3)

A proof of this theorem can be found in [18]. This property of the Kronecker product makes it easier to maintain the structure of the data model while iterating.
Bibliography


