Reduced Complexity MIMO Detection Algorithms based on Group Detection Techniques

A Thesis submitted toward the degree of Master of Science in Electrical and Electronic Engineering

By

Yohai Zmora

Submitted to the Senate of Tel-Aviv University
April 2008
Reduced Complexity MIMO Detection Algorithms based on Group Detection Techniques

A Thesis submitted toward the degree of
Master of Science in Electrical and Electronic Engineering

By

Yohai Zmora

This research was carried out in the Department of Electrical Engineering – Systems, under the supervision of Dr. Dan Raphaeli

April 2008
ABSTRACT:

In this thesis we study the different aspects of implementing a Group Detection (GD) based demodulator for a multiple-input multiple-output (MIMO) channel. MIMO channel decoding complexity for the optimal MAP detector increases exponentially with the product of the transmit antenna number and the number of bits per modulation symbol. This can rapidly become prohibitive even for simple schemes. Group Detection is a novel technique for reducing this complexity by partitioning the problem into groups and decoding each of them separately. GD enables the system designer to trade-off complexity and performance and to choose the point which best suites his application.

We attempt to explore the different aspects of GD implementation and the way different parameters such as the Group Partitioning and Group Processing Order affect the overall performance. Finally we suggest a new algorithm to maximize GD performances by using new criterions for both Group Partitioning and Ordering.
ACKNOWLEDGMENTS

I wish to express my appreciation and gratitude to Dr. Dan Raphaeli for his professional guidance and wise words.

I would also like to express my appreciation and gratitude for my family and friends for supporting me, standing by my side and pushing me forward.

Last but dependently not least, I would like to thank my wife Irit for being there always, for clearing the way, and giving me inspiration and support.
**TERMINOLOGY**

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWGN</td>
<td>Additive White Gaussian Noise</td>
</tr>
<tr>
<td>BICM</td>
<td>Bit Interleaved Coded Modulation</td>
</tr>
<tr>
<td>BER</td>
<td>Bit Error Rate</td>
</tr>
<tr>
<td>CSI</td>
<td>Channel State Information</td>
</tr>
<tr>
<td>GD</td>
<td>Group Detection</td>
</tr>
<tr>
<td>IC</td>
<td>Interference Cancellation</td>
</tr>
<tr>
<td>LSD</td>
<td>List Sphere Detector</td>
</tr>
<tr>
<td>MAP</td>
<td>Maximum A-Posteriori</td>
</tr>
<tr>
<td>MIMO</td>
<td>Multiple Input Multiple Output</td>
</tr>
<tr>
<td>MMSE</td>
<td>Minimum Mean Square Error</td>
</tr>
<tr>
<td>MSE</td>
<td>Mean Square Error</td>
</tr>
<tr>
<td>NM-SGD</td>
<td>Noise Minimizing Serial Group Detection</td>
</tr>
<tr>
<td>$N_0$</td>
<td>White noise spectral density</td>
</tr>
<tr>
<td>$N_G$</td>
<td>Group Size</td>
</tr>
<tr>
<td>$N_G$</td>
<td>Number of groups</td>
</tr>
<tr>
<td>$N_{Tx}$</td>
<td>Number of transmit antennas</td>
</tr>
<tr>
<td>$N_{Rx}$</td>
<td>Number of receive antennas</td>
</tr>
<tr>
<td>OFDM</td>
<td>Orthogonal Frequency-Division Multiplexing</td>
</tr>
<tr>
<td>OSIC</td>
<td>Ordered Successive Interference Cancellation</td>
</tr>
<tr>
<td>PCCC</td>
<td>Parallel Concatenated Convolutional Code</td>
</tr>
<tr>
<td>PGD</td>
<td>Parallel Group Detection</td>
</tr>
<tr>
<td>SISO</td>
<td>“Soft In Soft Out” or “Single In Single Out”</td>
</tr>
<tr>
<td>SM</td>
<td>Spatial Multiplexing</td>
</tr>
<tr>
<td>SNR</td>
<td>Signal to Noise Ratio</td>
</tr>
<tr>
<td>STC</td>
<td>Space Time Codes</td>
</tr>
<tr>
<td>V-BLAST</td>
<td>Vertical Bell Labs Space-Time Algorithm</td>
</tr>
<tr>
<td>ZF</td>
<td>Zero Forcing</td>
</tr>
</tbody>
</table>
**TABLE OF CONTENT:**

1  **INTRODUCTION** ..................................................................................................................... 1
   1.1  A REVIEW OF MIMO DETECTION TECHNIQUES: ................................................................. 2
   1.2  GROUP DETECTION PRIOR ART: .......................................................................................... 4
   1.3  THESIS CONTRIBUTION AND STRUCTURE ...................................................................... 7

2  **SYSTEM MODEL** .......................................................................................................................... 8
   2.1  MIMO-BICM TRANSMITTER: ................................................................................................. 8
   2.2  THE CHANNEL MODEL: ......................................................................................................... 9
   2.3  THE MIMO-BICM RECEIVER: ............................................................................................... 11
   2.4  REVIEW OF MIMO-BICM MAP DETECTION: ...................................................................... 13

3  **PRINCIPLES OF GROUP DETECTION** .................................................................................... 16
   3.1  GROUP DETECTION FRAMEWORK: ....................................................................................... 16
   3.2  GROUP SEPARATION: ............................................................................................................ 17
   3.3  GROUP LLR CALCULATION: .................................................................................................. 19
   3.4  BASIC GROUP PARTITIONING: ............................................................................................. 21

4  **ADVANCED ISSUES IN GD IMPLEMENTATION** ................................................................. 27
   4.1  PARALLEL GROUP DETECTION ............................................................................................. 27
   4.2  SERIAL GROUP DETECTION AND NOISE CANCELLATION ............................................... 30
   4.3  NOISE MINIMIZING SERIAL GROUP DETECTION (NM_SGD) .............................................. 34
      4.3.1  Soft Interference Cancellation and Symbol Statistics ....................................................... 36
      4.3.2  Symbol Ordering ............................................................................................................ 37
      4.3.3  Group Construction ........................................................................................................ 39
      4.3.4  Group Separation (and statistics initialization) ............................................................... 40
      4.3.5  The Full Algorithm Listing ........................................................................................... 41
      4.3.6  Comparison with prior work ........................................................................................ 42

5  **SIMULATION RESULTS** ........................................................................................................ 44
   5.1  UN-CODED PERFORMANCES: ............................................................................................. 44
   5.2  CODED PERFORMANCES – SINGLE ITERATION: .............................................................. 47
   5.3  CODED PERFORMANCES – ITERATIVE SCHEMES: ........................................................... 49
   5.4  COMPLEXITY AND DELAY ANALYSIS ............................................................................ 52

6  **SUMMARY AND CONCLUSIONS** ......................................................................................... 55

APPENDIX A – MANAGING SYMBOL STATISTICS IN NM-SGD .................................................... 57

BIBLIOGRAPHY ................................................................................................................................. 59
# Table of Figures:

- Figure 2.1: MIMO-BICM Transmitter ................................................................. 9
- Figure 2.2: MIMO-BICM Receiver ..................................................................... 11
- Figure 3.1: General Framework for GD .............................................................. 16
- Figure 3.2: Non-Overlapping Group Partitioning scheme .................................. 22
- Figure 4.1: Parallel Group Detection ................................................................. 27
- Figure 4.2: General structure of a NM-SGD receiver scheme ......................... 35
- Figure 4.3: Basic stages of NM-SGD ................................................................. 35
- Figure 5.1: Un-coded MIMO 4x4 QAM64 – Results ....................................... 45
- Figure 5.2: MIMO 4x4 rate 1/2 convolutional code - first iteration .................. 47
- Figure 5.3: MIMO 4x4 rate 1/2 convolutional code – iterative detection .......... 49
- Figure 5.4: MIMO 4x4 rate 1/2 UMTS Turbo code ........................................... 51
LIST OF TABLES

Table 3.1: Number of possible Group Partitions ..........................................................23
Table 3.2: Simplified partitioning for group size of 2 .....................................................25
Table 3.3: Simplified Partitioning for groups of size 4 ..................................................25
Table 5.1: Performance comparison for un-coded GD schemes ....................................46
Table 5.2: GD (un-coded) performance as function of Group Size...............................46
Table 5.3: Performance comparison for rate $\frac{1}{2}$ convolutional coded GD schemes ....48
Table 5.4: Performance comparison iterative detectors..................................................50
Table 5.5: Approximate Complexity Analysis ..............................................................53
1 INTRODUCTION

MIMO systems and channels are currently a hot topic for research and development because of their promises for extreme gains in spectral efficiency. However MIMO incredible performances come at the price of exponential demodulation complexity which quickly makes the optimal MAP solution unrealistic as the constellation and number of antennas increase.

The demand for MIMO based systems is fuelled by the ever increasing demand for spectral efficient transmission techniques. The pioneering work of Foschini [13] and Telatar [21] had shown the tremendous potential of the multiple-input multiple-output (MIMO) channel to break the traditional capacity limits of the AWGN channel model. In fact Information theory prediction suggest that in a rich scattering environment channel capacity would scale linearly with the minimum of the number of transmit and receive antennas.

This performance gains are achieved by exploiting the increased number of spatial dimensions created between the antennas. In spatial dimension we refer to the distinct paths through space the signal travels between the different antennas. Loosely speaking, schemes that exploit the spatial dimensions to improve the probability of error are said to maximize the diversity gain while schemes that use the spatial dimensions to increase the transmission rate are said to maximize the multiplexing gain. The main way to exploit the MIMO channel to increase the Diversity Gain is via Space Time Codes (STC) and especially Space Time Block Codes (STBC) ([1],[20]). This type of codes received a lot of attention due to their ability to create Diversity Gain in a simple easy to decode manner.

As important as Diversity Gain is for some application the real great promise from the MIMO concept relies in its high spectral efficiency which can only be tapped using Spatial Multiplexing (SM), that is the concurrent transmission from different antennas of different unrelated symbols. MIMO techniques had been demonstrated to reach spectral efficiency of over 40 bits/s/Hz [13]. This coupled with the fact that MIMO thrives in the high scatter environments of the home and city, makes it ideal for the
A review of MIMO detection techniques:

new generation of communication systems and it was adopted as the physical layer for both 802.11n (WLAN) and the 802.16 (WiMAX) standards.

Actual MIMO systems often combine the MIMO technique with other techniques in order to handle the demands imposed by the high bandwidth and the reach scatter environments. Probably the most common of these combinations would be the MIMO-OFDM combination. This combination utilizes OFDM ability to mitigate the multipath characteristics of the channel while giving us both high bandwidth and simultaneous channel estimation coupled together with the Multiplexing and Diversity gains of the MIMO technique.

In addition, since MIMO channels are almost by definition fading channels, MIMO system are almost always coupled with some type of Forward Error Correcting Code layer. There are however many schemes and designs for such layers, ranging from individually coded streams of data which are transmitted independently to BICM [6] systems which first code the entire data stream and then interleave the code frame bits before symbol mapping and transmission.

1.1 A REVIEW OF MIMO DETECTION TECHNIQUES:

All of these great MIMO advantages and abilities come at a price - detection complexity. As in any communication system the ideal receiver will have to jointly detect and decode both the channel and the code layer which is prohibitive even for the SISO case. However in the MIMO scenario even trying to implement just the ideal channel detector is often too complex. This is due to the fact that a Maximum Likelihood (ML) decision on the transmitted vector will have to account for all of the concurrent transmitted symbols possibilities. Such ML decision will therefore be exponential both in the number of bit per symbol and the number of transmitting antennas. We will review the full details of the optimal MAP detector as part of our System Model chapter.

There are many reduced complexity algorithms that were proposed in order to solve the MIMO demodulation problem. The most basic are simple linear channel equalizers which try to reverse the channel effect on the transmitted vector and estimate it, allowing us to use a traditional single symbol ML/MAP detector. The simplest of these linear equalizers is the Zero Forcing (ZF) equalizer which simply
Introduction

calculates the pseudo-inverse of the channel transmit matrix. However this can introduce significant noise enhancement and performance loss at low SNR. A somewhat more advanced equalizer is the Minimum Mean-Squared Error (MMSE) equalizer which takes into account the signal SNR while calculating the channel inverse.

The linear equalizers, while far from achieving capacity on their own have relatively low computational cost and therefore are often the starting point for many of the common MIMO detection techniques. These can be generally categorized as either Sphere search algorithms or Interference Cancellation/Suppression (IC/IS) techniques.

One of the first IC techniques introduced, and probably the most familiar one, is the V-BLAST [24] algorithm. V-BLAST represents one of the first practical solutions for MIMO detection and is the benchmark for many of the more advanced algorithms. Its detector is essentially a Decision Feedback Equalizer with smart symbol ordering. This is also sometimes referred to as OSIC – Ordered Successive Interference Cancellation. It performs iterations between a linear equalizer frontend (either ZF or MMSE) which decodes the highest SNR (or smallest MSE) symbol and an IC stage which cancels out the (hard) decoded symbol. Much of the decoding complexity of V-BLAST consists of the symbol reordering at each step. This however becomes negligible when working with a relatively static MIMO channel since it only depends on the channel matrix.

There are many other low-complexity Interference Cancellation techniques this range from Successive Soft interference Cancellation where the performance of V-BLAST are improved by preventing the error propagation of the hard decisions to a true iterative (“turbo”) Interference Cancellation ([5],[7]) techniques which use an initial estimate and the prior information in order to cancel the interference and softly detect the symbols.

The second group of MIMO detection algorithms falls under the headline of Sphere search algorithms [11]. These types of algorithms try to approximate the ML and Map decision by establishing a search radius around an initial estimate. This search can be visualized as finding the nearest lattice point to a noisy lattice observation. In-fact the lattice points can be regarded as the vector space spanned by the MIMO channel.
matrix, and the received vector is the noisy point on the lattice. The search radius of the algorithm defines a sphere around the received vector. If the search radius is permitted to span the entire lattice, the Sphere search for the closest lattice point will clearly give the ML result, however it will also result in usual ML complexity. In practice the Sphere Detector performs a limited search while continually updating the best lattice candidate. This will optimally result in a low complexity high performance algorithm. However in practice setting the search radius correctly for the optimal performances is a challenging task.

While the Sphere Detector gives us an approximate ML decision, we often need the soft LLR information given to us by the MAP algorithm especially when dealing with iterative algorithms. This requires a somewhat more complex algorithm in order to approximate the soft MAP result. The so called List Sphere Detector (LSD) [14], accomplishes this by utilizing the sphere search algorithm, but unlike the sphere detector it maintains a list of the best candidates at each stage. This of course result in higher complexity since each update now needs to propagate through the entire list, which makes the list size a vital parameter. Nevertheless [14] has shown that a schema using the LSD detector coupled with a turbo code in an iterative fashion can approach channel capacity. Generally speaking sphere search based techniques are of higher complexity and higher performances than IC based schemas.

1.2 GROUP DETECTION PRIOR ART:

As previously discussed there are many MIMO detection techniques which approximate the optimal ML/MAP detector. Each of these techniques is usually characterized by some tradeoff between performances and complexity. Group Detection (GD) is a novel technique which enables the system designer to control the detection complexity. The idea behind GD is simply to break down the detection problem into groups, and then to detect each group separately treating the other groups as noise. The system engineer can tailor the complexity performance tradeoff as he see fit, by changing one major parameter - the group detection size. This parameter gives us at one extreme, when the group size equals the transmitted vector size, the optimal MAP detector, and on the other, when group size equals one, the basic linear equalizer.
Introduction

GD was first proposed and detailed in the context of CDMA multi user detection by Varanasi at [23] later he and Fain at [10] had adapted it to the MIMO scenario and in particular shown how using GD can increase the diversity order compared to a linear detector. Essentially they had shown that while using a simple linear detector degrades the MIMO channel to a single AWGN channel with no diversity, and the full MAP detector enables us to enjoy the full MIMO channel diversity $M$ (which equals the number of transmit paths $N_{T}$ in the case of reach scatter), the GD detector gives us $M - |\tilde{G}|$ levels of diversity where $|\tilde{G}|$ is the number of elements left outside the group being detected.

Choi and Cioffi in [8] had suggested improving V-BLAST by using a partial GD approach. The detection is done using a partial "QR" factorization which splits the detection process into two distinct parts. First they detect an initial “grouped” transmit element, made of the best sub-channels, and only then continue with the traditional V-BLAST OSIC element by element detection. Using this method the authors claim to overcome the error-propagation problems of the OSIC method and reclaim the diversity for the sub-channels decoded first.

Li et al in [17] had implemented a simple ZF based GD using ML decisions. They were the first to address the issue of how to construct the group in order to best decode a desired antenna. They suggested grouping together the elements based on their correlation and signal strength in order to minimize the interference from outside the group. They have also combined GD with OSIC based detection but didn’t try to address the subject of overlapping groups.

Aoki et al in [2] built on [17] initial work and addressed specifically the subject of overlapping groups in a scheme of parallel independent GD detectors. However the authors only concentrated on how to make the decision which gives the best result. They suggested two criteria for choosing the best result for an overlapped element.

The work of Ming and Cheng in [18] demonstrates how GD can be combined with a Sphere Detector instead of the more common ML detector. This is an example of GD flexibility and versatility which help create a low complexity high performance system. It is very important to note that the concept of GD is separated from the actual detector used, and although the ML detector is often the obvious choice thanks to the
Group Detection Prior Art:

reduced complexity which makes it much more feasible, it is by no means the only choice.

In a more recent work Elkhazin et al in [9] used GD to develop two new MAP detectors for use as a SISO stage in an iterative detection for V-BLAST communication. The authors gave special attention to the case of an unbalanced channel where there are more transmit antennas than receiving. It is plain that in this type of scenario the extra diversity achieved by GD truly comes to play and improves the performances compared to linear detectors.

The authors developed a soft output GD detector based on Noise Whitening of the interfering groups and then demonstrated two detection schemes. One which used group overlapping and selected an optimal group for each bit being decoded and another scheme based on a static Group Partitioning with no overlap. The authors also tried to estimate the optimal partitioning and suggested a Group Partitioning scheme based on greedy criteria. We will go back and delve into more details of the first scheme, and compare it to ours in section 4.3.6.

Our work is based on the work of Levi in [16] which had developed a GD detector for BICM-MIMO channel using Information Theory principles. The author studied the problem of Group Partitioning and devised an optimal solution for groups of size 2 and an Ad-hoc simplified partitioning similar to that of [9] for groups of size 4. The author also gave special attention to the decoupling of the quadrature and inphase components of the transmitted signal, and proved that a decoupled scheme will always outperform a per antenna detection scheme. We will farther delve into Levi work and explain it in details when we will present the basics of performing GD in following chapters.
1.3 Thesis Contribution and Structure

This thesis aims to extend and continue the work of [16]. We, like Elkhazin et al in [9], use GD as SISO stage in an iterative receiver. However, we try to improve on [16] and [9] GD performance by using the diversity effect of overlapping group partitioning, while trying to maximize the information transfer from one group to the other by implementing an OSIC type group decoder.

We will investigate the different aspects of implementing different GD schemes designed to make use of different advantages of the technique. The schemes try to maximize the potential gains of the GD approach by using Information Theory motivations and insights. We will focus our discussion around three methods representing different tradeoffs of complexity, performance, and processing delay. We will start by describing the basic GD scheme which will serve us as our baseline this is the same scheme also used by [16].

We will then explore the effects allowing our groups to overlap. This result in a phenomenon we call “Group-Diversity”. We will analyze it by using a simple scheme we name PGD – Parallel Group Detection. This scheme will allow us to demonstrate the potential gain and advantages of schemes using “Group-Diversity”.

Next we will turn our attention to a novel and sophisticated OSIC style scheme for iterative (“turbo”) detection we call NM-SGD (Noise Minimizing Serial Group Detection). This scheme will serve us to demonstrate different aspects of implementing a serial processing GD scheme and will allow us to explore how to make the best use of prior information. As part of the NM-SGD scheme we will also introduce novel new criteria for ordering and group construction.

The rest of this work is organized as follows. We start with describing the system model and the optimal MAP detector. Second, the details of the basic GD framework and detector will be described including the different aspects of its implementation. Next we study of the effects of group-overlap and the so-called PGD algorithm. Now we come to our study of serial group detection and develop the principles of NM-SGD. Finally we conclude this work by examining the performances of the different schemes and suggest avenues for farther research.
2 SYSTEM MODEL

The basic system model for this work is based on the MIMO-BICM scheme. BICM or bit-interleaved coded modulation was proposed by [6] and [25] as an effective way to create code-diversity for fast fading channels. BICM tries theoretically to make the code diversity equal to the smallest number of distinct bits. This is achieved by bit-wise interleaving at the encoder output, and by using an appropriate soft-decision bit metric as an input for a soft input decoder.

The channel characteristic of MIMO and especially MIMO-OFDM is very attractive for the BICM notion. This is due to the rich multi-path environments and the frequency interleaving which makes the effective channel characteristic across subcarriers rather fast fading. The MIMO-BICM notion was extensively studied in the works of [19] and [4] and many others and is one of the leading implemented schemes today.

In this section we will start with a detailed description of the transmitter model used in this work, we will then describe the channel model and assumptions used for the simulations, finally we will describe the general framework of the receiver used and even go into a detailed description of the ideal soft-in-soft-out (SISO) MAP detector.

2.1 MIMO-BICM TRANSMITTER:

Consider the transmitter structure in Figure 2.1 for a MIMO-BICM [19] transmitter having $N_{TX}$ transmitter antennas. The data bits $b(k)$ assumed to be equally distributed and independent go through an encoder. We used three types of code layers during this work: the basic un-coded data stream, a simple Viterbi encoder and the more advanced Turbo encoder. The encoded data stream is divided into frames of $N_{FS}$ bits and passed to the interleaver ($\pi$) where it is randomly interleaved. The interleaved coded bits marked $c(k)$ are then divided into groups of $N_{Bi}$ bits and are mapped to the complex symbols $\hat{s}(k)$ by a complex constellation $\mathbb{C}$ of size $2^{N_{Bi}}$ using Gray code.
System Model

Figure 2.1: MIMO-BICM Transmitter

The mapped symbols are then de-multiplexed and transmitted in parallel through the $N_{tx}$ transmit antennas forming the transmit vector - $\mathbf{s}(k) = [\tilde{s}_1(k), \tilde{s}_2(k), ..., \tilde{s}_{N_{tx}}(k)]^T$ where $[ ]^T$ represents the vector transpose operation, and symbols in bold represent vectors.

2.2 THE CHANNEL MODEL:

We consider a flat-fading linear MIMO channel model with $N_{tx}$ transmitter antennas and $N_{rx}$ receive antennas sampled at symbol rate. The channel output at the $m$'th receiver antenna at the $k$'th time index is given by

$$\tilde{r}_m(k) = \sum_{n=1}^{N_{tx}} \tilde{h}_{mn}(k) \tilde{s}_n(k) + \tilde{n}_m(k)$$

(2.1)

where $\tilde{h}_{mn}(k)$ is the complex channel gain between the $n$'th transmitter antenna and the $m$'th receiver antenna, and $\tilde{n}_m(k)$ is an additive white Gaussian noise (AWGN) source of variance $\sigma^2$. We define the SNR at the channel output as

$$\text{SNR}_m = \sum_{n=1}^{N_{rx}} \mathbb{E}[|\tilde{h}_{mn}(k)|^2] \frac{E_s}{\sigma^2} = N_{rx} \mathbb{E}[|\tilde{h}_{mn}(k)|^2] \frac{E_s}{\sigma^2}$$

(2.2)

where $E_s = \mathbb{E}[|\tilde{s}_n(k)|^2]$ is the symbol energy which depends on the constellation $\mathcal{C}$ in use.
The channel model:

In order to simplify SNR calculation we normalize \( \tilde{h}_{mn} \) so that \( \mathbb{E}[\tilde{h}_{mn}(k)]^2 = 1/N_{Tx} \) which means that we generate \( \tilde{h}_{mn} \) as a normal distributed complex random variable with zero mean and \( 1/N_{Tx} \) variance.

It is often more convenient to represent the channel model in vector form as:

\[
\mathbf{\tilde{r}}(k) = \mathbf{\tilde{H}}(k)\tilde{\mathbf{s}}(k) + \mathbf{\tilde{n}}(k)
\]  

(2.3)

where \( \mathbf{\tilde{r}}(k) = [\tilde{r}_1(k), \tilde{r}_2(k), \ldots, \tilde{r}_{N_{Rs}}(k)]^T \), \( \mathbf{\tilde{n}}(k) = [\tilde{n}_1(k), \tilde{n}_2(k), \ldots, \tilde{n}_{N_{Rs}}(k)]^T \) and \( \mathbf{\tilde{H}} \) is the \( N_{Rs} \times N_{Tx} \) complex channel matrix which elements are \( \{ \mathbf{\tilde{H}} \}_{mn} = \tilde{h}_{mn} \).

Since in the reminder of this work we are mainly interested in a given channel realization at a fixed time index we will from now own omit the time index \( k \) for clarity of notation.

Finally it is often more convenient and in fact as we will demonstrate more advantageous to transform the complex channel equation into the real matrix equation

\[
\mathbf{r} = \mathbf{H}\mathbf{s} + \mathbf{n}
\]  

(2.4)

where \( \mathbf{r} = [\Re\{\mathbf{\tilde{r}}\}, \Im\{\mathbf{\tilde{r}}\}]^T \), \( \mathbf{s} = [\Re\{\mathbf{\tilde{s}}\}, \Im\{\mathbf{\tilde{s}}\}]^T \), \( \mathbf{n} = [\Re\{\mathbf{\tilde{n}}\}, \Im\{\mathbf{\tilde{n}}\}]^T \), and

\[
\mathbf{H} = \begin{bmatrix}
\Re\{\mathbf{\tilde{H}}\} & -\Im\{\mathbf{\tilde{H}}\} \\
\Im\{\mathbf{\tilde{H}}\} & \Re\{\mathbf{\tilde{H}}\}
\end{bmatrix}
\]  

(2.5)

is the \( N'_{Rs} \times N'_{Tx} \) real channel matrix where \( N'_{Rs} = 2N_{Rs} \), and \( N'_{Tx} = 2N_{Tx} \).

The channel simulation is based on the fast-fading model where a different independent channel realization is considered at each time instant. This is in contrast to the quasi-static model where the channel is considered static for the duration of a frame of symbols. We use the fast-fading model because it is more suitable to the MIMO-OFDM scenario, and also because our main focus is on the detector stage and we try to examine its performance under as many channel realizations as possible.
System Model

Our receiver is assumed to have full channel state information (CSI) which means it knows the correct values of the $H$ channel matrix. There are many ways the receiver can attain such knowledge such as the use of preambles, beacons and other common channel estimation techniques, which are outside the scope of this work.

2.3 The MIMO-BICM Receiver:

The general framework of an iterative MIMO-BICM receiver is shown in Figure 2.2. The $N_{\text{rx}}$ receive antennas each receive a complex symbol, see (2.1). The received symbols are then interpreted by the soft in soft out detector. The detector job is to inverse the channel effect on the transmitted vector and to detect the symbols originally transmitted. The detector is considered soft-output because its output is an a posteriori probability (APP) of a given symbol being the one originally transmitted rather than making a hard decision - which symbol was most likely transmitted.

Figure 2.2: MIMO-BICM Receiver

The a posteriori probability is often expressed as bit log likelihood ratios (LLRs)[14]. LLRs provide a convenient and efficient notation for describing the operation of iterative decoding algorithms, because simple add/subtract operations are sufficient to separate a priori or old information from new ("extrinsic") information obtained during a cycle of detection/decoding. The LLR of a given bit $b_n$ is expressed by

$$LLR[b_n] = \log \frac{P(b_n = 1)}{P(b_n = 0)}$$

It can be seen from (2.6) that the LLR sign can indicate the probable value of the bit (1 or 0 in our case), while the magnitude of the LLR can indicate the reliability of the decision.
The detector output can be expressed using LLRs in the form:

\[
\Lambda_1[b(n)] = \log \frac{p(b(n) = 1|x,H)}{p(b(n) = 0|x,H)} = \Lambda_1'[b(n)] + \Lambda_1''[b(n)]
\]  

where \( \Lambda_1'[b(n)] \) is the new *extrinsic* information gained about the received symbols (and bits), and \( \Lambda_1''[b(n)] \) is the *a priori* information provided by the channel decoder at the \( l \) iteration. This conforms to the generic form of the iterative module where the final decision \( L_D \) is based on the prior \( L_A \) and the new information \( L_E \). It is however important to notice that only the extrinsic information is passed on by the detector to the next iteration. This is an important principle of turbo detection, which exists in order to prevent positive feedback. Positive feedback might introduce unwonted error and bias the final result.

The SISO channel decoders produce an extrinsic LLR for each coded bit as

\[
\Lambda_{\text{e}}[b(n)] = \log \frac{p\left(b(n) = 1\mid \Lambda_{\text{e}}[b(j)], n \neq j\right)}{p\left(b(n) = 0\mid \Lambda_{\text{e}}[b(j)], n \neq j\right)} 
\]  

The calculated extrinsic LLRs are transferred iteratively between the detector and the decoder at what is referred to as Turbo processing or equalization \cite{3},\cite{15}. By passing the LLRs through the BICM interleaver and de-interleaver at each cycle we decouple the bits from each other and help make the decisions of the two detectors independent.

Finally after a given number of iterations or after it was determined that the decision had converged to its final state, the iterations stops and we can take a hard-decision over the final code-bits LLRs.
2.4 **Review of MIMO-BICM MAP Detection:**

The optimal APP detector – the Maximum A-Posteriori (MAP) detector can be derived from (2.7) by evaluating over all possible symbol vectors \( \mathbf{s} \). For clarity of notation we will omit the conditioning on the channel matrix \( \mathbf{H} \) from the following development.

First using Bayes rule we get:

\[
P\{b(n) = 1|\mathbf{r}\} = \frac{P\{\mathbf{r}|b(n) = 1\} \cdot P\{b(n) = 1\}}{P\{\mathbf{r}\}}
\]

(2.9)

Notice that the probability \( P\{\mathbf{r}\} \) is not a function of \( b(n) \) and serves only as a normalizing factor and therefore will be cancelled out when evaluated in (2.7).

The bit \( b(n) \) is determined by the symbol vector \( \mathbf{s} \). Let \( \mathcal{S} \) be the set of all possibly transmitted symbol vectors \( \mathbf{s} \), we define \( \mathcal{S}_i^n \) as the sub-set of \( \mathcal{S} \) having \( b(n) = i, i \in \{0,1\} \). Now we can write (2.9) as

\[
P\{b(n) = 1|\mathbf{r}\} = \sum_{\mathbf{s} \in \mathcal{S}_i^n} P\{\mathbf{r}|\mathbf{s}\} \cdot P\{\mathbf{s}\} \frac{P\{b(n) = 1\}}{P\{\mathbf{r}\}}
\]

(2.10)

Under the assumption of AWGN we can write:

\[
P\{\mathbf{r}|\mathbf{s}\} = \frac{1}{2\pi\sigma^2} \exp\left(\frac{-||\mathbf{r} - \mathbf{Hs}||^2}{2\sigma^2}\right)
\]

(2.11)

Finally combining (2.10) and (2.11) into (2.7) we get:

\[
\Lambda_i[b(n)] = \log \frac{\sum_{\mathbf{s} \in \mathcal{S}_i^n} p(\mathbf{s}) \exp\left(\frac{-||\mathbf{r} - \mathbf{Hs}||^2}{2\sigma^2}\right)}{\sum_{\mathbf{s} \in \mathcal{S}_0^n} p(\mathbf{s}) \exp\left(\frac{-||\mathbf{r} - \mathbf{Hs}||^2}{2\sigma^2}\right)}
\]

(2.12)

It is already possible to evaluate and solve the equation in this form, but let’s simplify it farther. Assuming the BICM interleaver is large enough we can safely assume the bit vector associated with \( \mathbf{s} \) is uncorrelated and therefore the bits are independent.
Review of MIMO-BICM MAP detection:

\[
p\{s\} = \prod_{j=1}^{N_TN_R} p\{b(j)\} = p\{b(n)\} \cdot \prod_{j \neq n} p\{b(j)\} = p\{b(n)\} \cdot p\{s^-\} \tag{2.13}
\]

We broke down the bit vector represented by \(s\) and defined a new vector \(s^-\) as the original vector without its \(n\) bit. Now placing (2.13) into (2.12) we notice that all the elements in the numerator have in common the element \(P\{b(n) = 1\}\) and symmetrically all the elements in the denominator have in common \(P\{b(n) = 0\}\) therefore we can now remove them out of the sum and get:

\[
\Lambda_{i}\left[ b(n) \right] = \log p\{b(n) = 1\} \cdot \frac{\sum_{s \in \mathbb{B}} \exp \left( -\frac{\|r-Hs\|^2}{2\sigma^2} \right) \cdot p\{s^-\}}{\sum_{s \in \mathbb{B}} \exp \left( -\frac{\|r-Hs\|^2}{2\sigma^2} \right) \cdot p\{s^-\}} \tag{2.14}
\]

It can already be seen we have reached the wanted form we described for (2.7), that is we have a prior element \((L_{\Lambda})\) which is summed together with a new extrinsic element. We can farther simplify (2.14) by trying to describe the bit probabilities using their prior LLRs:

\[
p(s^-) = \prod_{j \neq n} p\{b(j)\} \cdot \prod_{j=1}^{N_TN_R} p\{b(j)\} = \prod_{j \neq n} e^{LLR(b(j))} \cdot \prod_{j=1}^{N_TN_R} 1 + e^{LLR(b(j))} = \prod_{j \neq n} e^{LLR(b(i))} \cdot \prod_{j=1}^{N_TN_R} 1 + e^{LLR(b(j))} \tag{2.15}
\]

This result is direct from the definition of the LLR - (2.6) where the notation - \(J, i \in \{0,1\}\) simply means all the bit indices of \(s^-\) where \(b(j) = i\). Notice that the denominator of (2.15) doesn’t depend on the actual bit vector that we are trying to calculate its probability, and there for it will be canceled out in (2.14). Using (2.15) we get our final form:
System Model

\[
\Lambda_1[b(n)] = L_A(b(n)) + \log \frac{\sum_{r \in \mathcal{R}_1} \exp \left( -\frac{\|r - Hs\|^2}{2\sigma^2} \right) \exp \left( \sum_{j \in \mathcal{J}_1} L_A(b(j)) \right)}{\sum_{r \in \mathcal{R}_0} \exp \left( -\frac{\|r - Hs\|^2}{2\sigma^2} \right) \exp \left( \sum_{j \in \mathcal{J}_0} L_A(b(j)) \right)} 
\]  

(2.16)

Where \( L_A(b(n)) \) symbolize the prior LLR value of the \( n \) bit.

It can be clearly be seen that the calculation of (2.16) though simplified by the use of the LLR, is nevertheless of exponential complexity and requires to go over all the \( 2^{N_T N_s} \) possible values of the transmit vector \( s \). This can quickly become prohibitive even for relatively small number of transmit antennas and medium constellations.
3 PRINCIPLES OF GROUP DETECTION

This chapter aims to introduce the basic framework and principles which underline Group Detection and makes the approach viable. We will start by describing the general framework and components which comprise GD and describe some of the basic parameters which affect its performance. We will then go into the details of some of the basic elements and describe their implementation. Finally we will review some general implementation issues and prepare the ground for the advanced algorithms described in the next chapter.

3.1 GROUP DETECTION FRAMEWORK:

The general process involved in performing GD is illustrated in Figure 3.1. The process is broken into three distinct parts which will be detailed in the following sections. The detection starts by separating the received vector into groups. This is the fundamental process upon which GD is based. The separation is done based on the channel matrix and utilizes both prior information from the decoder, and optionally information gained by feedback from previous GD steps. The process of selecting the groups themselves is called “Group Partitioning” and determines both the size of the different groups, the elements each group contains, and even the order in which the groups will be processed. Finally, once the wanted group had been selected and separated from the rest, we can calculate the APP probability for the group and the appropriate bit LLR’s.

Figure 3.1: General Framework for GD
Principles of Group Detection

In the next two sections we will delve into an in-depth description of the Group Separation and Group LLR Calculation process. This two fundamental processes had been well documented and researched ([16],[9]) and are relatively clear to implement. Group Partitioning however still remains a somewhat open subject, which can be approached in many ways. As stated before there are many parameters involved in Group Partitioning, some such as group size have a clear and understood effect on performances and complexity, while others such as the grouping of different elements, and the group processing order are much harder to decipher.

Probably the most important parameter for GD implementation is the Group Size. It is this parameter which gives us the ability to control the performance verses complexity ratio of GD. By setting the group size to be minimal (one) we are reducing our LLR calculation complexity to its minimum, and essentially reproducing a simple linear filter. On the other extreme setting the Group Size to its maximum value will generate a group of the same size as the original input vector, and the detection problem will be that of the optimal (and exponential complex) MAP detector.

One of the main goals of this work is to explore the different issues involved in implementing different Group Partitioning schemes and the ways feedback information can be used to enhance GD performances. In the final section of this chapter we will review one elementary Group Partitioning scheme proposed by [16], which will serve as our base of comparison for the advanced techniques reviewed in the following chapter.

3.2 GROUP SEPARATION:

The most basic step of GD is the Group Separation process. The main idea here is to break down the received vector into groups to be processed separately. However since the received vector \( r \) is a super composition of all the transmit elements that were received concurrently at all the receive antennas there is no reason to break it down into its separate elements. What we really want is groups composed of several transmit elements each. This way we can in the next stage limit our MAP search only to the relevant transmit elements.
Group Separation:

So we want a simple, preferably linear, method to separate the received vector to groups of transmit elements:

\[ \bar{s}_{Gi} = W_{Gi}r \]  

(3.1)

where \( \bar{s}_{Gi} \) represents a vector grouping together the estimation of several transmit elements selected to be in the \( Gi \) group, and \( W_{Gi} \) is the linear separation matrix used to separate the \( Gi \) group out of the received vector.

Levi at [16] suggests an Information Theory approach for finding the optimal separation matrix \( W_{Gi} \) by trying to maximize the information gained:

\[ W_{Gi}^{opt} = \arg \max_{W_{Gi}} \{ I(s_{Gi}, \bar{s}_{Gi}) \} \]  

(3.2)

Levi showed that under the Gaussian assumption we can use a sub-matrix of the well known MMSE matrix to optimize equation (3.2).

\[ W^{MMSE} = \Sigma_{rs} \cdot \Sigma_{rr}^{-1} = \Omega H^r \left[ H \Omega H^r + \sigma_n^2 I \right]^{-1} \]  

(3.3)

\( \Sigma_{rs} \) - represents the cross-correlation matrix between the received vector \( r \) and the transmitted vector \( s \). \( \Sigma_{rr} \), is the auto-correlation matrix of the received vector. \( \Omega = \text{diag} \left( \text{var}(s) \right) \), is the diagonal matrix which elements represent the transmitted symbols variance, and \( \sigma_n^2 I \) is a diagonal matrix with the noise variance as its elements.

The actual separation matrix for a particular group \( G_i = \{ x_{g_i}, x_{g_1}, ..., x_{g_N} \} \) of transmit elements indexes of size \( N_G \leq N_{Tx} \) is a sub-matrix of the MMSE matrix created by grouping together the lines matching the group indexes and is denoted as \( W_{Gi} \).

It is important to note that under normal initial conditions the \( \Omega \) matrix is the identity matrix multiplied by the transmit power \( \sigma_s^2 I \) and therefore can be replaced in equation (3.3) by the scalar \( \sigma_s^2 \). However in the following iterations \( \Omega \) represents our prior knowledge and should be treated with care.
3.3 GROUP LLR CALCULATION:

Once the current group has been separated, it can be relatively easily detected in any of the common detection method used for MIMO detection. However for optimal performances we would like to develop the optimal MAP detector for use with GD.

That is a detector that calculates: $p(s_g | \hat{s}_g, H)$

For clarity of notation we will only examine the case of a single group $G$ and its complement made of all the elements left outside the group and denoted $\overline{G}$. We will start by examining the exact nature of the received input for the detector:

$$\hat{s}_g = W_g r = W_g H_g s_g + W_g (H_{G G} s_G + n)$$

We can see that the input is composed of two parts; the first is the part we want which is a direct function of our selected group $G$, the second denoted by $v_g$ is in fact the noise experienced by the group $G$ and is a function of both the WGN noise and the noise generated by the elements of $\overline{G}$. $H_g$ and $H_{G G}$ are the sub-matrixes of the channel matrix corresponding to the group $G$ and $\overline{G}$ accordingly.

In order to estimate the noise effect of $v_g$ we need the covariance matrix of $v_g$ given by:

$$\Sigma_{v_g v_g} = W_g [H_g \Omega_g H_g^T + \sigma^2 I] W_g^T$$

Now, recalling the evaluation of the General MAP decoder in the previous chapter, we need to compute the conditional pdf $f(s_G | \hat{s}_G)$. From equation (3.5) and under the Gaussian assumption on the inter group interference [16] it follows:

$$p(\hat{s}_g | s_g) = \frac{1}{(2\pi)^{N_g} |\Sigma_{v_g v_g}|^{\frac{1}{2}}} \exp \left( -\frac{1}{2} (\hat{s}_g - W_g H_g s_g)^T \Sigma_{v_g v_g}^{-1} (\hat{s}_g - W_g H_g s_g) \right)$$

Equation (3.6) is the multivariate Gaussian distribution of $\hat{s}_g | s_g$, where $|\Sigma_{v_g v_g}|$ represent the determinant of $\Sigma_{v_g v_g}$. It is clear we are dealing with a colored Gaussian
variable which significantly complicates the LLR calculation. It was suggested both by [16] and [9] to whiten the noise of equation (3.4) by using eigen value decomposition:

\[ \Sigma_{v_g,v_g} = U_G \Lambda_G U_G^T \]  

(3.7)

Where \( U_G \) is a \([G \times G]\) unitary matrix and \( \Lambda_G \) is a \([G \times G]\) diagonal matrix of the eigen values of \( \Sigma_{v_g,v_g} \). The noise whitening matrix can then be calculated by

\[ F_G = \Lambda^{-1/2} U_G^T \]  

(3.8)

Now we redefine our separation matrix as:

\[ W_G^{white} = F_G \cdot W_G^{MMSE} \]  

(3.9)

Notice that using our new definition the noise covariance becomes:

\[ \Sigma_{v_g,v_g} = I_{[\mathbb{C}]} \]  

(3.10)

Now we can rewrite (3.6) as:

\[ p(\hat{s}_o | s_o) = C \cdot \exp\left(-\frac{1}{2} \| s_o^{white} - W_G^{white} H_o s_o \| ^2 \right) \]  

(3.11)

where \( C \) is a constant value (which will be eliminated), and \( s_o^{white} \) is our group estimation using the whitening group separation matrix. Finally the group LLR can be evaluated as:

\[ LLR\left[ b(n) \right] = \log \frac{\sum_{i=1}^{l} p(s_o) \exp\left(-\frac{1}{2} \| s_o^{white} - W_G^{white} H_o s_o \| ^2 \right)}{\sum_{i=1}^{l} p(s_o) \exp\left(-\frac{1}{2} \| s_o^{white} - W_G^{white} H_o s_o \| ^2 \right)} \]  

(3.12)

Equation (3.12) represents the optimal MAP – GD detector. The complexity of its solution is exponential with the Group Size and the Constellation size. This is in fact the entire idea behind GD. By controlling the parameter of Group Size we can control the over-all detection complexity and adjust it to our resources.
Principles of Group Detection

While equation (3.12) is the optimal solution (under the Gaussian assumption) of the GD problem, it is by no means the only one. As stated before any MIMO detection scheme can be incorporated into GD and used to calculate the group LLR. Logically of curse we will prefer high performance and high complexity detectors, since the idea behind GD is to enable the implementation of detectors that would otherwise be unpractical.

In practice most GD implementation, ours included, do implement in one form or another the MAP detector as described in equation (3.12), but there are some exception such as [18] which implemented a Group Sphere Decoder.

3.4 Basic Group Partitioning:

As previously stated, unlike Group Separation and Group LLR Calculation which are relatively clear and straight-forward to implement, Group Partitioning is a large and diverse subject. The reason is that there are many implementation issues relevant to GD which falls under this category.

The most basic and straight forward issue is Group Size. As previously shown the Group Size is the basic parameter controlling the complexity and performances of GD algorithms. As a general rule we would like to work with the minimal Group Size still giving us the performance we need. By doing this we reduce the complexity and there by the resources needed for our implementation to its minimum.

Other parameters however are not as clear, or straight forward to implement. The main issues that arise are the actual Group Construction and the order of Group Processing. Group Construction relates to the selection of elements into groups. Group Processing relates to the order in which we process our groups and whether or not we use feedback and interference cancellation.

Obviously all the parameters previously mentioned influence both the performance and the complexity of the GD scheme. However there are also other issues which are affected. For example the total processing delay will be greatly extended if we opt to use serial group processing, which means processing the groups one by one feeding back each group solution to improve the next. Other issues which may arise include robustness, and precision.
Basic Group Partitioning:

It is one of the main goals of this work to explore the various aspects of GD implementation and suggest various schemes appropriate in different processing scenarios. We will start with the basic scenario presented by Levi at [16] and continue in the next chapters to explore more advanced scenarios.

Levi examined the basic scenario where the transmit elements are partitioned into groups of equal size $N_G$ with no overlap allowed:

$$
\Psi = \{G_1, ..., G_{N_G}\}
$$

$$
a \neq b \forall a \in G_i, b \in G_j, i \neq j
$$

The information flow in this scheme is depicted in Figure 3.2. Since there is no overlap between the groups, and since the MMSE matrix is the canonical separation matrix, it is possible to process all the groups concurrently. This makes the scheme ideal for low latency applications.

![Figure 3.2: Non-Overlapping Group Partitioning scheme](image)

Even with this basic simplification it is clear we are faced with huge amount of possible Group Partitions, as displayed in Table 3.1. We can theoretically try each and every partitioning option one after the other until we find the optimal one, but this obviously won't be practical. What we need is a method to determine based on our channel state information what would be the best group partition to detect with.
Table 3.1: Number of possible Group Partitions

<table>
<thead>
<tr>
<th>Transmit Scheme</th>
<th>Group Size</th>
<th>Possible Group Partitions</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N_{tx} = 2$</td>
<td>$N_G = 2$</td>
<td>3</td>
</tr>
<tr>
<td>$N_{tx} = 4$</td>
<td>$N_G = 2$</td>
<td>105</td>
</tr>
<tr>
<td>$N_{tx} = 4$</td>
<td>$N_G = 4$</td>
<td>35</td>
</tr>
<tr>
<td>$N_{tx} = 8$</td>
<td>$N_G = 2$</td>
<td>675675</td>
</tr>
<tr>
<td>$N_{tx} = 8$</td>
<td>$N_G = 4$</td>
<td>225225</td>
</tr>
<tr>
<td>$N_{tx} = 8$</td>
<td>$N_G = 8$</td>
<td>6435</td>
</tr>
</tbody>
</table>

Once again Levi suggested approaching the problem from Information theory point of view. The optimal partitioning scheme according to this would be the one maximizing the sum rate of the groups information.

By using the chain rule of mutual information the mutual information of the MIMO channel in GD scheme can be written as:

$$I(r; s) = I(r; \hat{s}_{G_i}) + I(r; \hat{s}_{G_{i+1}}} | \hat{s}_{G_{i}}, ..., \hat{s}_{G_{N_g}})$$  \hspace{1cm} (3.14)

Since in the general GD framework there is no exchange of information between different groups, the mutual information of equation (3.14) can't in general be realized. The actual mutual information (sum rate) of the general GD scheme can be expressed as:

$$I(r; s) \geq \sum_{i=1}^{N_g} I(r; \hat{s}_{G_i})$$ \hspace{1cm} (3.15)

This is simply the sum of the mutual information of the different groups under the assumption the groups are disjoint. The sum rate can be written as:

$$\sum_{i=1}^{N_g} I(r; \hat{s}_{G_i}) = \sum_{i=1}^{N_g} h(\hat{s}_{G_i}) - \sum_{i=1}^{N_g} h(\hat{s}_{G_i} | r)$$ \hspace{1cm} (3.16)

Assuming the transmit elements are i.i.d the first term of equation (3.16) does not depend at all on the selected Group Partitioning. We are left with trying to minimize the second term of (3.16) which represents the estimation error of each group.
Basic Group Partitioning:

Since the differential entropy of a Gaussian random vector is only a function of its covariance matrix, it follows we need to find the matrix corresponding to the estimation error. Levi had shown that this matrix can be determined from the MMSE estimation error covariance matrix given by:

$$\Sigma_{ee} = \Sigma_{ss} - \Sigma_{sr} \Sigma_{rr}^{-1} \Sigma_{rs}$$  \hspace{1cm} (3.17)

The estimation error covariance matrix for the group $G_i$ is a sub matrix of $\Sigma_{ee}$ and is denoted as $\Sigma_{ei,ei}$. The matrix is constructed from $\Sigma_{ee}$ by striking out the $k$'th rows and $k$'th columns $\forall k \notin G_i$. Using $\Sigma_{ei,ei}$ we can now express (3.16) as:

$$\sum_{i=1}^{N_x} I(r; s_{G_i}) = \sum_{i=1}^{N_x} h(s_{G_i}) - \frac{1}{2} \sum_{i=1}^{N_x} \log \left( \left| \Sigma_{ei,ei} \right| \right)$$  \hspace{1cm} (3.18)

Now we can express the overall optimization problem as:

$$\left\{ G_1^{opt}, ..., G_{N_y}^{opt} \right\} = \arg \min_{G_1,...,G_{N_y}} \left( \prod_{i=1}^{N_y} \left| \Sigma_{ei,ei} \right| \right)$$  \hspace{1cm} (3.19)

Equation (3.19) gives us a framework for calculating and determining the optimal group partitioning; however the complexity of the above search quickly becomes overwhelming and is unpractical (at least in the general case). Nevertheless using the internal structure of the covariance matrix $\Sigma_{ei,ei}$ Levi was able to find and prove the optimal solution for the case of 2x2 MIMO channel and groups of size 2 (real valued transmit elements). As it turns out the optimal scheme depends on the values of the channel correlation matrix:

$$\Sigma_{ii} = HH^T$$  \hspace{1cm} (3.20)

This symmetric matrix values correspond to the correlation between the channels experienced by the different transmitted symbols. This in-turn can be viewed as the amount of interference each symbol creates to its neighbors. In the case of groups of size 2 it is easy to use this insight to create a simplified group partitioning algorithm:
Table 3.2: Simplified partitioning for group size of 2

1) \( n = 1, \Phi_n = \{h_{i,j} \forall i < j < N_{T_X}\} \)
2) \( [i_n, j_n] = \arg \max_{(i,j) \in \Phi_n} (h_{i,j}) \)
3) \( G_n = \{i_n, j_n\} \)
4) \( \Phi_{n+1} = \{h_{i,j} \in \Phi_n \forall i \neq i_n, j \neq j_n, i \neq j, j \neq j_n\} \)
5) \( \text{If } (+n) \leq N_y \text{ then goto 2 else end} \)

While Levi proved the algorithm is optimal for the case of \( N_{T_X} = 4 \) (2x2, complex, MIMO channel), the algorithm basic intuition of trying to maximize the correlation of elements within the group is true for any channel and the algorithm presents a simple way to do so for any GD scheme with Group Size of 2.

The same motivation is true for any size group, but for groups larger than 2, finding the combination of elements which maximize the correlation is somewhat less straightforward. The problem relies in the fact that our correlation matrix only tells us the correlation between two elements and no more. Levi suggested the following greedy algorithm which attempts to maximize the correlation of elements inside a group of size 4:

Table 3.3: Simplified Partitioning for groups of size 4

1) \( n = 1, \Phi_n = \{h_{i,j} \forall i < j < N_{T_X}\} \)
2) \( [i_n, j_n] = \arg \max_{(i,j) \in \Phi_n} (h_{i,j}) \)
3) \( k_n = \arg \max_{(i,k) \in \Phi_n, i \neq i_n} (h_{i,k} + h_{i,j}) \)
4) \( l_n = \arg \max_{(i,j) \in \Phi_n, j \neq j_n} (h_{i,j} + h_{j,k} + h_{k,l}) \)
5) \( G_n = \{i_n, j_n, k_n, l_n\} \)
6) \( \Phi_{n+1} = \{h_{q,p} \in \Phi_n \forall q \neq i_n, k_n, l_n, p \neq i_n, j_n, k_n, l_n\} \)
7) \( \text{If } (+n) \leq N_y \text{ then goto 2 else end} \)

This is in fact a generic algorithm which can be adjusted for any size group with small modifications. However by definition the algorithm is not optimal, though the greedy approach will almost always improve the results compared to a totally random partitioning. A similar somewhat more sophisticated approach using a normalized version of the correlation matrix was used by [9].
Basic Group Partitioning:

An important thing to notice about Levi approach is the fact that the channel correlation matrix is a byproduct of the MMSE calculation. This means that the algorithm complexity is that of a simple search for maximal values in a list.

While the basic GD system described in this section is very attractive in the sense of its simplicity, low complexity and low latency, it is still far from optimal in its performances. The obvious drawbacks of the scheme lays in two distinct problems: the first is that there is no information exchange between the groups; the second is that restricting the groups to non-overlapping elements by definition creates less optimal groups.

In the next chapters we explore different algorithms which try to attend to these two main problems. Finally we will present a novel approach which tries to maximize the possible performances of GD while still remaining simple, adjustable, and of low complexity.
Advanced issues in GD implementation

4 ADVANCED ISSUES IN GD IMPLEMENTATION

In this chapter we try to examine and explore the behavior of GD based algorithms and schemes trying to overcome the two basic problems discussed in the last chapter. First we study what are the effects of allowing the group partitions to overlap. We devise a simple algorithm named Parallel Group Detection (PGD) and demonstrate the effect of group diversity.

We then attempt to find a way to transfer information between the detectors of different groups, for that order we devise a simple serial interference cancellation scheme similar to the classic V-BLAST detector and study its performances.

Finally we conclude by suggesting a new novel scheme named Noise Minimizing Serial Group Detection (NM-SGD) which attempts to use both effects in order to maximize the achievable results of GD.

4.1 PARALLEL GROUP DETECTION

In this section we will examine the effects of GD schemes allowing the elements inside different groups to overlap. Once more we are looking at a basic GD scheme similar to the one described in chapter 2. The new scheme depicted in Figure 4.1 differs in that instead of using the MMSE separation matrix only once, now it is used $N_s$ times in order to create several unrelated group partitioning.

![Figure 4.1: Parallel Group Detection](image-url)
This means that we now have $N_s$ times the groups we used to in the simpler method. More importantly this means each element from the transmitted elements was detected at least $N_s$ times as part of different Group Partitions. The basic question we ask is can we gain anything by using this scheme?

We will try to answer this question first by going back to Information Theory and try to analyze the suggested scheme. Let’s look first at the simple case where $N_s = 2$ and we are trying to decode the transmitted symbol $s_i$ which is part of group $G_1$ and $G_2$. That is we have: $\hat{s}_{G_1}$ and $\hat{s}_{G_2}$ as the result of our group separation, and we know that $s_i \in \hat{s}_{G_1}$ and $s_i \in \hat{s}_{G_2}$ there for the information we have about $s_i$ can be expressed as:

\[
I(s_i;r) = I(s_i;\hat{s}_{G_1}) + I(s_i;\hat{s}_{G_2} | \hat{s}_{G_1})
\]

The first thing we should notice about equation (4.1) is the term $I(s_i;\hat{s}_{G_1} | \hat{s}_{G_2})$. This is the conditional information added by $\hat{s}_{G_2}$ about $s_i$ and it is a direct result of using the PGD scheme and stand in contrast to equation (3.16). According to Information Theory we know that:

\[
I(s_i;\hat{s}_{G_1} | \hat{s}_{G_2}) \geq 0
\]

This is due to the fact that information is always a non-negative quantity. This interesting result shows that as expected the PGD scheme can only improve our results by introducing new information by the use of different Group Partitioning. We refer to this effect as “Group Diversity”, since like transmit diversity and other sorts of diversities the amount of Groups Partitioning being processed directly affect the amount of information we have on our desired symbol.

Trying to analyze equation (4.1) farther is however a difficult task. The problem relies in the exact evaluation of the term $H(s_i;\hat{s}_{G_1},\hat{s}_{G_2})$. This entropy will depend on the particular way we decide to combine the information our two group estimations give
Advanced issues in GD implementation

us. We can, however, easily give an upper-bound for our performances by considering the union of our two groups:

\[ I(s_1; \hat{s}_{G_1}) + I(s_1; \hat{s}_{G_2} | \hat{s}_{G_1}) \leq I(s_1; \hat{s}_{G_1 \cup G_2}) \]  

(4.3)

One should notice that at least in theory this upper-bound is achievable since thanks to the canonical nature of the MMSE matrix used for group separation we know that:

\[ \hat{s}_{G_1 \cup G_2} = \hat{s}_{G_1} \cup \hat{s}_{G_2} \]  

(4.4)

This suggests that we can always use the brute force approach and do a joint MAP search for the two groups. However, this will obviously result in the exponential complexity we are trying to avoid.

The question of combining the information from the different groups in order to make the most of the Group Diversity effect is a difficult one, and is opened to farther research. We opted for the simplest of solutions, even if just to examine the raw potential of the concept. We suggest simply adding up together the LLR of the bits from the different groups.

The question of the optimal choice of \( G_1 \) and \( G_2 \) in order to maximize the performances is also non-trivial. It depends obviously on the exact performance criteria we are trying to optimize and on our method of joining together our detectors. We can, however, get some insight by trying to maximize our upper-bound for the information gained for \( s_1 \).

The “trick” is to regard our combined group as a new “super” group which we are trying to construct in parts. The first group is comprised in the exact same method as previously described, the second will comprise from our wanted element \( s_1 \) and the elements not yet grouped in \( G_1 \) with the highest correlation to \( s_1 \), and so on.

This algorithm while simple isn’t too useful, usually there is more than one element we are trying to decode, and what we are really after is a scheme that will optimize the detection for all the elements, and not a particular one. This scheme is hard to find and makes a complex optimization problem. Luckily, if we examine the properties of
Serial Group Detection and Noise Cancellation

PGD a little closer we might find that in most practical situation there little need for such scheme, in fact there is no need for any complex Group Partitioning scheme.

One of the interesting points of Group Diversity and equation (4.1) is the fact we are continuously gathering information. In the general case of \( N_s \) ”iterations” this can be written as:

\[
I(s_1; r) = I(s_1; \hat{s}_{G_1}) + I(s_1; \hat{s}_{G_2} | \hat{s}_{G_1}) + \ldots + I(s_1; \hat{s}_{G_{N_s}} | \hat{s}_{G_1}, \hat{s}_{G_2}, \ldots, \hat{s}_{G_{N_s-1}})
\]  

(4.5)

However as one might expect the amount of information each new item adds continuously diminishes, this is due to the fact that every new information added to the sum will be depended on all the previous information, that is the \( N_s \) item in the equation is of the form \( I(s_1; \hat{s}_{G_{N_s}} | \hat{s}_{G_1}, \hat{s}_{G_2}, \ldots, \hat{s}_{G_{N_s-1}}) \).

This effect, however, also has a positive side. Because of the statistical nature of the addition operations, and because of the fact our information is limited, even if we don’t imply any smart Group Partitioning at all, and only use simple random partitioning, we will still get almost the same amount of information after a couple of adding operations.

This phenomenon not only simplifies PGD implementation, but also demonstrates its increased robustness. The fact we are decoding the same data in several different constellations enable us to cancel out all kinds of calculation accuracy problems and noise phenomena.

4.2 SERIAL GROUP DETECTION AND NOISE CANCELLATION

While enabling our groups to overlap, as demonstrated by the PGD scheme, does give us a marked increase in performance there is still a significant gap from the optimal MAP decoder. The main problem is that we are just combining the information gained from each group, but there is no real transfer of information between the groups. That is, the information gained from the first group detection is not used to help detect the next group. In other words, we are not using the knowledge we gained on the other symbols we already detected to help us when we come to detect a new symbol.
Advanced issues in GD implementation

Almost by definition, algorithms which use the first detection to help with the next are serial algorithms. The best known example for a serial MIMO detection algorithm is the one used as part of the V-BLAST[24] scheme. The V-BLAST detector is classified as Decision Feedback Equalizer with Ordering. At every stage the algorithm repeats the following basic steps:

1) **Equalization**: Usually ZF or MMSE equalizers which separates and estimate the original transmitted elements.

2) **Ordering**: Select which of the remaining elements will be decoded. The selection is classically done based on the element with the best SNR or lowest MSE.

3) **Detection**: The element selected is detected using a ML or MAP detector.

4) **Feedback**: A hard decision is taken from the detector and transferred to the equalizer. The decision is used to cancel out the transmit element that was detected and thereby reduce the problem dimensions.

The V-BLAST algorithm is one of the first and most successful MIMO detection algorithms ever suggested. Its strength lays in its relative simplicity; it enables us to reduce the detection problem to one element in a time while still maintain reasonable performances. Unfortunately the very thing which makes it so good is also one of its main problems. The serial nature and the hard decision feedback can introduce error propagation and corrupt several elements in a row.

The main way the V-BLAST detector deals with error propagation, is by symbol ordering. By reordering the symbols each stage according to their SNR or MSE the scheme can reduce the chance for error propagation to minimum. However this significantly increases the overall complexity of the scheme, and does not completely eradicates the problem.

The most interesting and relevant point in the V-BLAST detector to our GD scheme is the way it accomplishes its feedback. The feedback is done by cancelling out the transmitted elements already detected. This result in the following equation
Serial Group Detection and Noise Cancellation

\[ \bar{r}' = \bar{r}'-\hat{s}_j h_j \]
\[ = \sum_{n=1}^{N} s_n h_n - \hat{s}_j h_j + n \]  \hspace{0.5cm} (4.6)
\[ = \sum_{n=1}^{N} s_n h_n + n \]

\( \bar{r} \) is initialized as \( \bar{r}' = r = \sum_{n=1}^{N} s_n h_n + n \), and \( j \) is the iteration number (\( j > 0 \)). \( h_j \) is the column of the channel matrix representing the channel the \( n \) symbol experienced are and we are assuming the detector has made the correct decision \( \hat{s}_j = s_j \).

We can see in equation (4.6) how each iteration reduces the problem dimensions by removing a column from the channel matrix. Thereby also reducing the inter symbol interference experienced by the remaining symbols and improving their detection probability.

We would like to apply the same principles of the V-BLAST detector to our problem, and create a serial GD scheme. It seems intuitive to follow the exact same stages as the V-BLAST algorithm and simply adjust them to our familiar GD scheme and create a so called GD-BLAST detector performing the following stages:

1) **Equalization**: As usual we use our MMSE matrix to equalize and separate into groups.
2) **Ordering**: Based on a given Group Partitioning scheme select which of the remaining groups will be decoded. Once more the selection can be done based on the group with the best SNR or lowest MSE.
3) **Detection**: The selected group will be transferred to our Group Detector and its LLR will be calculated.
4) **Feedback**: A hard decision about the entire group will be calculated, and transferred to the equalizer. Once again the decision is used to cancel out the transmitted elements that were detected in order to reduce the problem dimensions.

Somewhat counter intuitively at first this simple algorithm fails to perform, and will be actually outperformed by the classical V-BLAST algorithm until relatively big Group Sizes will be reached. This however helps to shed light on some of the hidden
Advanced issues in GD implementation

properties of serial ordered detection algorithms and raises some of the important implementation points to take notice for.

The first and probably the most significant effect, is the Ordering Resolution. We define the Ordering Resolution as the size of the smallest part we can change its order. Equivalently we can look at it as the amount of different orders we can choose from. The ordering resolution for the classical V-BLAST is that of a single transmit element, on the other hand in our GD-BLAST scheme we are limited to the size of a single group which would probably be bigger. This means that no matter what is our current Group Partitioning scheme at each ordering stage V-BLAST will always have more choice and will probably manage to find a better candidate (SNR or MSE wise) to decode. This effect is made worse since the obvious groups ordering criteria is their average group SNR, which will naturally decline as the group is made larger seeing as by definition it will contain also elements with lower score. Therefore at each stage the V-BLAST scheme will always find a better candidate to be decoded then the proposed GD-BLAST scheme.

At first, we might expect this to be negligible effect as the better performances of the GD detector will compensate thanks to the bigger group size. However in practice, this is only true for relatively big Group Sizes. This is due in part to the fact that, thanks to the good ordering, the V-BLAST detector makes most of its initial decisions in relatively high SNR; thereby it avoids both error propagation and inter-symbol interference. This reduces the performance gap to the GD detector which has to content with lower average SNR.

It is interesting to note the effect transforming the channel to its real valued equivalent (equation (2.5)) has on the Ordering Resolutions. By braking down the complex transmitted elements into their real and imaginary parts we can effectively double our ordering resolution compared to the complex case. This can actually improve the performances of the classical V-BLAST scheme by several dB a phenomena also noticed by [12]. This emphasis the importance’s of the real channel equivalent for GD schemes. The breakdown of the complex elements gives us both higher degrees of freedom for the grouping process and for the ordering in the case of serial algorithms.
Noise Minimizing Serial Group Detection (NM_SGD)

The only way to use a serial GD based scheme and still maintain the maximal “Ordering Resolution”, is to allow group overlapping. In this way at each stage we can select a single transmit element we want to detect and then detect it using the best group we can. This also means that at each stage we remove out from our ordering possibilities only the element we focused our detection around. This will in turn obviously generate much more detection stages ($N_{tx}$) and higher overall complexity.

The second significant limiting factor of GD-BLAST is its error propagation. This is in part related to the ordering issue but is also an inherited problem of GD. Because as mentioned before the detection is done over an entire group at once, and therefore the detection suffers from lower average SNR and is more likely to introduce some error to one of the grouped symbols.

There are two possible ways to resolve the error propagation problem of the GD-BLAST scheme. First we can select to only cancel out the wanted element in the detection group. This solution however means we are throwing away all the information we gained about the other symbols in the group. The second solution is to use soft decisions and soft interference cancellation; this however has the drawback of never reducing the problem dimensions which may result in higher overall complexity.

4.3 **Noise Minimizing Serial Group Detection (NM_SGD)**

Using the principles shown by the PGD scheme and the GD-BLAST scheme we are finally ready to present a true high performance GD scheme. Our novel algorithm named “Noise Minimizing Serial Group Detection” utilize the Group Diversity effect coupled with serial processing while maintaining the maximal degree of Ordering Resolution. The scheme also tries to incorporate in a clever way any prior information available in order to make it a true iterative “turbo” processing scheme.

The general structure of the NM-SGD receiver is described in Figure 4.2. The scheme complies with the general structure of the BICM turbo receiver as shown in Figure 2.2 where the main difference is that we enable the NM-SGD detector to make internal iterations. The internal iteration enables us to fully utilize the transfer of information between the groups, and enable the symbols inside the groups that were detected first, to enjoy the information gained by the detection of the subsequent groups.
Advanced issues in GD implementation

Figure 4.2: General structure of a NM-SGD receiver scheme

As its name implies NM-SGD is a serial algorithm which means it repeatedly performs a set of detection steps for each element that was transmitted. The basic stages of NM-SGD are depicted in Figure 4.3. The figure depicts the flow of information between the different stages and the external decoder. The prior information is feed in the standard form of L values (LLR). The algorithm itself makes use of the somewhat more efficient and more useful symbol statistics (see appendix A for further information). Finally the symbol statistics are converted back to LLR form and transferred to the decoder.

Figure 4.3: Basic stages of NM-SGD

Every detection phase starts by updating the symbol statistics based on the current prior data, either in the form of L values from the decoder or in the form of the previous stage detection statistics. By updating our symbol statistics we are actually
incorporating our prior knowledge into scheme and enabling the separation to make use of it.

Once we have all the information about our symbols we can choose which one to focus on for the next phase of detection. The scheme is designed to iterate through each of the transmitted symbols, while ordering them according to the information gained by their detection. After we chose the next symbol to focus on, we need to choose the best group to decode it with. Again, this should be the group that will gain us the most information.

Once we have constructed our group partition it is time to perform our group detection. First we construct our group separation matrix, taking care to initialize the statistics for the detection group. Finally after the group has been separated, we can decode it using our standard GD detector.

Each of the scheme stages was fine tuned, and specific criteria had been developed in order to maximize the flow of information, and the over-all performance of the scheme. In the next sub-sections we will delve into the fine details of each of the main stages of the algorithm and explain the exact details and reasoning behind its implementation.

4.3.1 **SOFT INTERFERENCE CANCELLATION AND SYMBOL STATISTICS**

One of our main conclusions from our experiment with GD-BLAST at the last section was the need for soft interference cancellation. The reasoning was that the group decoder is more prone for error propagation since it decodes an entire group of symbols at once with various SNR/MSE.

The idea behind soft interference cancellation is that we don’t make a hard decision about our symbols at the end of each stage but instead we assess our knowledge of each symbol and only use what we know about it. By not making a final decision, we are preventing a possible error from propagating and allow more room for possible correction at the next stages.

In practice soft interference cancellation is based on the two basic symbol statistics: the symbol mean and variance. The symbol mean or in other words its expected value is the value we want to use when we cancel out our symbol since it represents our best
guess of its possible value. The symbol variance on the other hand represents the accuracy of our knowledge. If we are completely sure of our estimation our variance would be very close to zero, while if we are completely unsure, it should be at its initial value based on the constellation.

The symbols variance is very important to our scheme and is used in many of the stages from the calculation of the separation matrix to the decision of which element goes into which group. The expected value of the symbols is cancelled out from the received vector at the beginning of each iterative step in the following manner:

$$
\hat{r}^t = r - HE(\hat{s}) = H(s - E(\hat{s})) + n
$$

(4.7)

It is important to note that for practical reasons the scheme maintains and uses a table of symbol statistics internally, while its external input and output is in the form of standard LLRs. Therefore there is a need to convert from the two forms of statistics representation. More about the mathematics and calculation involved in maintaining the statistics inside NM-SGD can be found in appendix A.

### 4.3.2 Symbol Ordering

Classic serial algorithms for the MIMO channel like V-BLAST put much emphasis on the ordering phase. In fact serial methods are very sensitive to the issue of ordering since it is the main method of controlling error propagation - the great nemesis of all serial algorithms. That is why most serial MIMO schemes order the symbols to decode based on either their SNR or their MSE.

NM-SGD however is based on soft-feedback and therefore is less prone for error propagation regardless to the order. The target of our ordering scheme is to increase over-all performances by reducing the number of required iterations.

The classical ordering approach has good initial performances but suffer however from a basic flow. It doesn’t integrate all the existing prior knowledge into its decision. What we suggest is a greedy approach that maximizes the information we gain at each detection step. In order to do so, we look on our information equation (4.5) as the result of an iterative process where at each step we add a new term with new information to what we already know:
Noise Minimizing Serial Group Detection (NM_SGD)

\[
I^k(s;r) = I^{k-1}(s;r) + \frac{1}{\Delta I^k} \left( r_s \hat{s}_{G_k} \ldots \hat{s}_{G_{k-1}} \right)
\]

(4.8)

\( k \) is our current iteration \((k > 1)\), and \( I^k(s;r) \) is the information we have about our transmitted symbols at the \( k \) iteration, and its initial value is:

\[
I^1(s;r) = I(r_s \hat{s}_{G_1})
\]

(4.9)

At each new iteration, we gain new information marked \( \Delta I^k \), which is the term we want to maximize. As can be seen in (4.8) the new information naturally depends on all the information we already gathered. Unfortunately \( \Delta I^k \) is in general somewhat difficult to evaluate, but we have in hand two quantities which we know directly effect it.

The first is the classical MSE criteria. The MSE directly assess the accuracy of our result and therefore for the amount of good information we will gain by detecting a particular element. The MSE of a given symbol to be detected is directly available to us from the diagonal of the MMSE matrix:

\[
\text{MSE}(\hat{s}_i) = W_{i,i}^{\text{MMSE}}
\]

(4.10)

The second quantity is the symbol variance (see appendix A for calculation). The symbol variance is a direct estimate of the knowledge we have about a particular element. The lower the variance the more confident we are about a particular element and therefore the more information we have about it. Another maybe more accurate view is that the lower the variance of a particular element is, we have less to learn about it, meaning that there is less information unknown about it.

It is clear that the two quantities affect \( \Delta I^k \) in opposite ways. On the one hand the lower the MSE of the chosen element is, the detection is likely to result in more information, however, if the variance of the element is low, then most of the information gained by the detection would in-fact already be known and won’t increase \( \Delta I^k \) since it is conditioned on all the existing information. Therefore we suggest estimating \( \Delta I^k \) for each element by using the following criteria:
Advanced issues in GD implementation

\[ \Delta I^k (s_i) \propto \frac{\text{VAR}(s_i)}{\text{MSE}(s_i)} \]  

(4.11)

Where at each stage we will choose to focus our detection on the element \( \hat{s}_i \) which maximize this expression

\[ i = \max \arg_j \frac{\text{VAR}(\hat{s}_j)}{\text{MSE}(\hat{s}_j)} \]  

(4.12)

4.3.3 \textit{GROUP CONSTRUCTION}

In the ordering stage we only selected a single element which we wanted to focus our detection on. This was due to two basic reasons: The first was in order to preserve the Ordering Resolution we discussed at section 4.2; the second was simply because it isn’t realistic to maximize \( \Delta I^k \) for every possible selection of groups. Therefore our greedy approach is to maximize \( \Delta I^k \) first by choosing the best element to decode and then constructing the best group to decode it with.

Once again we take the informatics approach and aim to select the group for \( s_i \) in such a way that will maximize \( \Delta I^k \) that is:

\[ G_K = \arg \max_{G_i \in G} \Delta I^k \]  

(4.13)

Yet again, we have no practical way to directly calculate the contribution of each element to the overall information. So once more we try to asses it based on the quantities at hand. We already know from previous algorithms depicted that the criterion that most affect the group detector performance is the inter-symbol/group interference - \( v_G \).

What we want is to minimize the term \( v_G \) of equation (3.4). Classically we would try to reduce this, by grouping together elements with high cross correlation in order to keep the disturbance from outside the group to it minimum. However, when trying to minimize \( v_G \) we must also remember that its power is a direct factor of our knowledge of the disturbing symbols. After all, by correctly estimating the disturbing symbols we can cancel them out completely leaving only the white noise. Once more...
we turn to the symbol variance in order to assess our knowledge of the symbols and in turn their interfering power.

Our noise minimizing criterion is therefore a factor of two quantities the cross-correlation of the different symbols and the chosen symbol, and the variance of these symbols. The higher each of these quantities is for a particular symbol, the more power it would contribute to the noise if it was left outside the group. Our group is then constructed based on the following greedy criterion:

\[
G_k = \arg \max_{\hat{G}_k} \sum_{n \in G_k, n \neq i} \text{corr}(\hat{s}_n, \hat{s}_i) \text{var}(\hat{s}_n)
\]  

(4.14)

Where \(\text{corr}(\hat{s}_n, \hat{s}_i)\) is simply the matching element from the channel cross-correlation matrix - \(h_{n,n}\).

4.3.4 GROUP SEPARATION (AND STATISTICS INITIALIZATION)

Once we selected the group to be detected, it is now time to separate it from the rest of the received vector. Generally we should use equation (3.3), as is, to calculate our separation matrix. However, first we must do one vital action.

We must first initialize our prior knowledge for the detection group of symbols. This so called initialization process is actually common among “Turbo” processing schemes and is part of the “Turbo Principle”. The idea is to cancel the prior information for the bit/symbol we want to detect and only use the prior information for the rest of the bits/symbols. This is done in order to prevent bias and positive feedback which may degrade the results over several iterations. This type of problem can be seen in the work of Elkhazin et al in [9] where the authors had to resort to pre-scaling the LLR values.

If we remember equation (4.7) then we realize our modified received vector already contains our prior information because it passed soft interference cancellation and is currently only a combination of our estimation error and the White Gaussian noise. We must then perform:

\[
\mathbf{F}^k = \mathbf{r}^k + \mathbf{H}_{\hat{G}_k} \mathbf{E}(\hat{s}_{\hat{G}_k}) = \mathbf{H}_{\hat{G}_k} \mathbf{s}_{\hat{G}_k} + \mathbf{H}_{\hat{G}_k} \left( \mathbf{s}_{\hat{G}_k} - \mathbf{E}(\hat{s}_{\hat{G}_k}) \right) + \mathbf{n}
\]  

(4.15)
Advanced issues in GD implementation

where \( \bar{G}_k \) is the complement group of \( G_k \) meaning that it contains all the elements that do not belong to \( G_k \). By introducing back our estimation for our detection group \( G_k \) we are essentially assuming nothing about its value and there by performing our initialization. Equivalently we could skip equation (4.7) and directly perform

\[
\vec{F}^k = r - \bar{G}_k \cdot E(\bar{s}_{\bar{G}_k}) = H_{\bar{G}_k} s_{\bar{G}_k} + H_{\bar{G}_k} \left( s_{\bar{G}_k} - E(\bar{s}_{\bar{G}_k}) \right) + n
\]

Once we have correctly initialized our received vector according to the group we want to detect it is important to also initialize the variance of the symbols inside the detection group by assuming we know nothing but the original constellation distribution. This is important since the variance plays a significant role in equation (3.3) in the form of \( \Omega = \text{diag}(\text{var}(\hat{s})) \).

After initializing our detection group statistics we can continue in the standard fashion to calculate the separation matrix according to (3.3) and to decode the group using a standard group detector.

4.3.5 THE FULL ALGORITHM LISTING

Now that we described the logic and the math behind the different steps of the algorithm we are finally ready to give its full listing:

1) \( j = 1, k = 1 \), Initialize all statistics.
2) For every full iteration \( j \) do:
   3) \( G_{\text{detected}} = \{ \emptyset \} \)
   4) For \( k = 1 \) to \( N \) do:
       5) \( i = \text{max arg} \frac{\text{var}(\hat{s})}{\text{MSE}(\hat{s})} \)
       6) \( G_{\text{detected}} = G_{\text{detected}} \cup i \)
       7) \( G_k = \text{arg max}_{G_k} \sum_{n \in G_k, n \neq i} \text{corr}(\hat{s}_n, \hat{s}_i) \text{var}(\hat{s}_n) \)
       8) \( \vec{F}^k = r - H_{G_k} \cdot E(\bar{s}_{G_k}) = H_{G_k} s_{G_k} + H_{G_k} \left( s_{G_k} - E(\bar{s}_{G_k}) \right) + n \)
       9) Initialize variance for symbols of \( \hat{s}_{G_k} \).
       10) \( W_{\text{MMSE}} = \Omega \cdot H^T \cdot \left( [H \Omega H^T + \sigma_n^2 I]^{-1} \right) \)
       11) \( \hat{s}_{G_k} = W_{G_k} \vec{F}^k \)
       12) For every possible combination of \( s_{G_k} \) do:
           13) \( p(\hat{s}_{G_k} | s_{G_k}) = \mathcal{C} \cdot \exp \left(-\frac{1}{2} \| F_{G_k} \hat{s}_{G_k} - F_{G_k} W_{G_k} H_{G_k} s_{G_k} \|^2 \right) \)
4.3.6 *Comparison with Prior Work*

It is interesting to note some of the key differences in features and approaches used in NM-SGD relative to competing schemes. The most relevant one of those to compare with, is the one suggested by Elkhazin et al in [9] named RD-MAP.

To begin with, both schemes focus their detection stages around a single symbol, making them of similar complexity. Interestingly, although very different approaches for the development of the group construction criteria where taken, both algorithms use the same final criteria, and will produce the same group once we decided on which symbol to focus our detection on.

The actual detection process in both schemes is very similar, although RD-MAP does not take care to initialize its prior knowledge for the detection group, and therefore suffers from bias in its detector, which it tries to compensate for by pre-scaling the prior LLRs.

The main difference between the schemes lays in the fact that NM-SGD allows the transfer of information between the different groups. This fact is due to the serial nature of the algorithm and its unique ordering criteria. RD-MAP is in a sense a parallel algorithm, decoding and constructing each group as if no other group exists and therefore it only uses the prior information it received from the decoder.

Although, theoretically parallelism may allow faster implementation, in practice, in order to improve the initial detection results RD-MAP must wait to complete entire “turbo” iteration against the code layer, while NM-SGD can improve the results simply by using the available data from the other groups. This in turn means that NM-SGD can converge more quickly, and require less iteration against the code layer.

Farther more, the smart ordering criteria, and the fact that in each stage of the scheme we update the statistics for the entire group, means that NM-SGD also enjoys the “Group Diversity” effect, where each symbol is detected several times; each time in a different group. This unique feature of NM-SGD, help maximize the overall information gained by single iteration of NM-SGD and helps it converge very quickly.
Advanced issues in GD implementation

The serial nature of the algorithm, unfortunately also has a price – added complexity. Since each serial stage requires the soft cancellation of the previous one, it also requires the recalculation of the group separation matrix, which contributes to the over-all complexity relative to RD-MAP.
5 SIMULATION RESULTS

We will now present and discuss the simulation results of the different algorithms presented in the previous chapters. The simulations are based on the MIMO-BICM scheme presented in chapter 2, and more specifically on Figure 2.1 and Figure 2.2.

The simulations had been performed over the presented algorithms and several base-line methods in a variety of channel combinations and using different code layers. We also studied the effect of key parameters on the performances of the presented algorithms. We will present select results which best demonstrate the behavior and comparative performances of our algorithms.

5.1 UN-CODED PERFORMANCES:

We start by comparing the un-coded results of the key algorithms presented. Figure 5.1 presents the results for the basic algorithms discussed in this work for the case of a 4x4 (complex) MIMO channel using a QAM64 constellation.

We measure the error rate as function of the total SNR at the receiver. In the un-coded case the BER is measured between the bits at symbol modulator input to the bits at the receiver demodulator output. For base-line purposes, the algorithms compared include the commonly used MMSE linear equalizer and the V-BLAST algorithms. The graph also presents the results for the basic GD scheme, the PGD scheme and the NM-SGD algorithm. Each GD scheme is presented once with a group size (the S parameter) of 2 and once with a group size of 4. The PGD scheme presented is of a factor of 4 (the I parameter), meaning each element was decoded 4 separate times. Finally we also present the somewhat defunct GD-BLAST algorithm, just in-order to demonstrate its inherited problem.
Simulation Results

By analyzing the un-coded results we can demonstrate several effects common to our GD algorithms. First comparing the different GD algorithms we can see exactly how the increasing sophistication of the scheme increases the overall performance this is demonstrated in Table 5.1 by comparing the SNR needed to achieve a fixed BER ratio. The table only shows the performance for GD with Group Size of 4(S=4). We can see that the most sophisticated scheme - our proposed NM-SGD, sets the standard. It is followed by the PGD and V-BLAST schemes which perform very similar to one another, although they are very different in nature. V-BLAST is a complex serial processing algorithm, and the PGD is a simpler brute-force approach. The simple GD scheme comes last but still has more than one dB advantage over the common MMSE equalizer.
Un-coded performances:

Table 5.1: Performance comparison for un-coded GD schemes

<table>
<thead>
<tr>
<th></th>
<th>MMSE</th>
<th>GD</th>
<th>V-BLAST</th>
<th>PGD</th>
<th>NM-SGD</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNR for 6E-2 (dB)</td>
<td>24.6</td>
<td>23.4</td>
<td>23</td>
<td>22.8</td>
<td>22.1</td>
</tr>
<tr>
<td>Gap (dB)</td>
<td>2.5</td>
<td>1.3</td>
<td>0.9</td>
<td>0.7</td>
<td>0</td>
</tr>
</tbody>
</table>

The analysis also clearly shows how increasing the group size increases the scheme performances. This effect is also demonstrated in Table 5.2 by examining the performance of the different GD schemes at a fixed SNR.

Table 5.2: GD (un-coded) performance as function of Group Size

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Group Size 2</th>
<th>Group Size 4</th>
<th>Approx Gap</th>
</tr>
</thead>
<tbody>
<tr>
<td>GD</td>
<td>5.28E-2</td>
<td>4.21E-2</td>
<td>0.9dB</td>
</tr>
<tr>
<td>PGD</td>
<td>4.79E-2</td>
<td>3.37E-2</td>
<td>1.2dB</td>
</tr>
<tr>
<td>NM-SGD</td>
<td>2.86E-2</td>
<td>2.23E-2</td>
<td>0.6dB</td>
</tr>
</tbody>
</table>

On a side note, it is interesting to analyze the GD-BLAST scheme. The scheme demonstrate for us how important is the correct design of serial algorithms, and the effect of the reduced Order Resolution has on the performances of the detector. Even though its performances can increase as function of the Group Size, we can clearly see how the increased error propagation diminishes its performances to the level of the much simpler GD scheme, not to mansion the clear superiority of the V-BLAST scheme.
Simulation Results

5.2 CODED PERFORMANCES - SINGLE ITERATION:

The next stage of analyzing our GD schemes is to examine the effect the code-layer has on their performances. Our receiver is in general a “turbo” processing receiver which means it can perform several iterations between the demodulator and the decoder therefore we must make a clear distinction between the first iteration which doesn’t require prior processing abilities from our schemes to the rest of the iterations which also measure the ability of the demodulator to incorporate the decoder prior results.

Our receiver performances for the case of a 4x4 MIMO channel using a QAM64 constellation and rate ½, convolutional encoder with octal generators (133,171) is presented on Figure 5.2. The decoder used is a “Soft In Soft Out” (SISO) standard MAP decoder design in order to make use of the prior (soft) information feed to it by the detector.

![Figure 5.2: MIMO 4x4 rate 1/2 convolutional code - first iteration](image-url)
Coded performances - Single Iteration:

It is very interesting to notice the profound change of performances compared to the un-coded scheme results. We can now clearly see the major fundamental gains the GD principle provides. The advantage of using a broader MAP detector has clear effect over the accuracy of the L values which manifest them self clearly in the performance gap between the GD schemes and the more traditional MMSE and V-BLAST alternatives. In fact the performance gap between the GD schemes and the traditional schemes is over 3.5 dB at the 10E-3 region.

Also very interestingly the performance gaps between the GD schemes had been dramatically reduced as can be seen in Table 5.3. This is effect is due to the nature and capabilities of the code layer in use. Generally speaking the code layer is more sensitive to the slight variations and inaccuracies of the different detection schemes. This is best seen in the PGD algorithm, where the robust nature of summing up the L values from 4 different detection results in a more accurate L value which in turn results in performances which manage to equal and even out-perform the more sophisticated NM-SGD detector.

<table>
<thead>
<tr>
<th></th>
<th>Traditional</th>
<th>GD</th>
<th>NM-SGD</th>
<th>PGD</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNR for 1E-3 (dB)</td>
<td>&gt;24</td>
<td>21.1</td>
<td>20.6</td>
<td>20.3</td>
</tr>
<tr>
<td>Gap (dB)</td>
<td>&gt;4</td>
<td>0.8</td>
<td>0.3</td>
<td>0</td>
</tr>
</tbody>
</table>

The NM-SGD scheme on the other hand suffers from a slight degradation in performance due to a small bias introduced by its serial nature. The bias is in fact a form of soft error propagation and is mostly due to numerical and accuracy problems propagated through the algorithm serial stages. However, it is important to remember that the PGD increased performance come at the price of increased over-all complexity compared to the NM-SGD scheme.
5.3 CODED PERFORMANCES – ITERATIVE SCHEMES:

Finally we are ready to analyze our receiver when it performs several complete iterations between its detector and decoder. The results for the NM-SGD scheme under different performance parameters are shown in Figure 5.3.

Figure 5.3: MIMO 4x4 rate 1/2 convolutional code – iterative detection

Figure 5.3 clearly shows how the NM-SGD scheme improves its performances after each iteration as is also demonstrated by Table 5.4 which shows the performance gap between different iteration at a fixed BER of 1E-3.
Coded performances – Iterative Schemes:

### Table 5.4: Performance comparison iterative detectors

<table>
<thead>
<tr>
<th></th>
<th>PGD</th>
<th>NM-SGD (S=2,I=1)</th>
<th>NM-SGD (S=2,I=1)</th>
<th>NM-SGD (S=2,I=1)</th>
<th>NM-SGD (S=2,I=2)</th>
<th>NM-SGD (S=4,I=1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNR for 1E-3 (dB)</td>
<td>19.8</td>
<td>20.5</td>
<td>17.2</td>
<td>16.3</td>
<td>17.1</td>
<td>18.4</td>
</tr>
<tr>
<td>Gap (dB)</td>
<td>3.5</td>
<td>4.2</td>
<td>0.9</td>
<td>0</td>
<td>0.9</td>
<td>2.1</td>
</tr>
</tbody>
</table>

An interesting effect demonstrated by Table 5.4 and Figure 5.3 is the fact that increasing the Group Size (S) or the number of internal iterations (I) within the detector, does not increase the overall performance but actually degrades them.

The author of [16] has also mentioned a similar trade-off when dealing with schemes based on group detection and interference cancellation. He noted that once we gained enough information about our symbols the interference cancellation starts to “kick in” and the inter symbol/group interference is reduced, it then becomes better to use small groups for detection since, in that way, we allow for better use of our gained information. A similar effect is also true for the NM-SGD scheme where the algorithm converges rapidly and manages to gain most of its information in the first stages of its initial pass on the symbols.

The strong convergence of NM-SGD also cancels out the need for internal iterations since those only serve to increase the error propagation from the numerical mistakes previously mentioned. We therefore suggest that NM-SGD is optimally used with a single internal iteration and group size of 2.

Figure 5.4 shows our receiver performance using a stronger code layer comprised of a “Turbo” Parallel Concatenated Convolutional Code (PCCC) defined in the UMTS standard [22] and data frames of 2048 bits.
Figure 5.4: MIMO 4x4 rate 1/2 UMTS Turbo code

The first thing to notice about the figure is how much closer we are now to achieving capacity. According to [14] the unrestricted capacity of the 4x4 MIMO channel for rate 1/2 should be achieved around 11.5dB SNR. As can be seen in the figure we are just 1dB away from this limit.

Another interesting thing showing from the figure is the fact that at low SNR, NM-SGD outperforms PGD and the other GD schemes even in its first iteration. We suspect this is the result of slight bias in the PGD L values caused by the fact we are simply adding them and not using some more sophisticated method which will account for the conditional information between the groups.
Finally, Figure 5.4 also compares two different types of turbo implementations trade-offs. In the first we define our code layer so it would make 16 internal turbo iterations and only 3 external iterations between the detector and the decoder (marked with “NM-SGD (TI=16, S=2)(x3)”). At the second tradeoff we make 5 external iterations but only 4 internal turbo decoder iterations at each (marked with “NM-SGD (TI=4, S=2)(x5)”). The results show a slight advantage to the second approach even though we estimate it to be of roughly the same overall complexity or less. Nevertheless finding the optimal configuration will require a more thorough analysis and may require the use of “exit charts” analysis.

5.4 Complexity and Delay Analysis

Our discussion and analysis of our GD schemes wouldn’t of course be complete without analyzing their complexity. It is however important to note once more that the idea behind GD is that we can control the complexity of the given scheme by changing the Group Size parameter (S).

The Group Size parameter affects the complexity in two related ways. First and foremost it directly determines the complexity of the Group LLR calculation. This can be stated in the case of the optimal detector as $2^{N_G-N_b}$ where $N_G$ is the Group Size and $N_b$ is the number of bit per symbol.

The second way the group size affects the complexity is through the Group Separation process. While the MMSE matrix calculation is canonic, and therefore doesn’t depend on the group size. The number of the needed (or possible) groups will decrease as the group size increases.
Simulation Results

Table 5.5 summarize our approximate complexity analysis of our discussed schemes. We had broken down the complexity into two main parts. The complexity of the actual detection process on the one hand. And the complexities of the accompanying process like equalization and ordering.

Table 5.5: Approximate Complexity Analysis

<table>
<thead>
<tr>
<th></th>
<th>MAP</th>
<th>MMSE</th>
<th>V-BLAST</th>
<th>GD</th>
<th>PGD</th>
<th>NM-SGD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detection</td>
<td>$2^{N_{s}\cdot N_{s}}$</td>
<td>$N_{Tx} \cdot 2^{N_{s}}$</td>
<td>$N_{Tx} \cdot 2^{N_{s}}$</td>
<td>$\frac{N_{Tx} \cdot 2^{N_{s}}}{N_{G}}$</td>
<td>$\frac{N_{Tx} \cdot 1^{\infty}}{N_{a}} \cdot \frac{1}{2} \cdot 2^{N_{s}/N_{G}}$</td>
<td>$\frac{N_{Tx} \cdot 2^{N_{s}}}{N_{G}}$</td>
</tr>
<tr>
<td>Equalization</td>
<td>$N_{Rc}^3$</td>
<td>$\sim N_{Tx}^3 N_{Rc}^3$</td>
<td>$N_{Tx} + N_{Rx}^3$</td>
<td>$N_{Rx}^3$</td>
<td>$\sim N_{Rx} N_{Rc}^3$</td>
<td></td>
</tr>
</tbody>
</table>

It is important to note though that the complexity of the equalization is a function of a matrix inversion process which, in practice, is hard to estimate and therefore is taken to be $O(N_{Rc}^3)$. In addition while we assumed the equalization is repeated for every stage for the serial algorithms, in practice there are iterative methods of calculating these equalization matrixes requiring only a fraction of the calculations.

The analysis shows that for a fixed Group Size the GD scheme is the simplest while the PGD scheme complexity depends on the factor $I^{PGD}$ which determines the number of overlapping groups and increases the scheme complexity in a linear factor. The NM-SGD complexity is much closer to that of the GD scheme and is increased mainly by the repeated equalization process. However in practice since NM-SGD Group Size will be normally limited to 2 while $I^{PGD}$ is in practice at least 3 or 4, NM-SGD will result in much lower over-all complexity.

However, in practical systems complexity isn’t everything, there other practical considerations such as delay and memory requirements. While these will depend heavily upon the actual implementation, there are, however, several limiting factors we would like to raise to consideration.

First, it is apparent that in general the delay of serial schemes will be greater than alternative schemes. If we assume our basic Group LLR calculator has a delay of $T_{G}$ seconds, then, at best, NM-SGD will finish an internal iteration within $N_{Tx} \cdot T_{G}$ seconds.
seconds. PGD however is only limited by the amount of available Group LLR calculators which means it can output its result in nearly $T_G$ seconds.

Another important factor to remember is the memory requirement of implementing a true iterative “turbo” processing scheme since, not only, does it requires us to remember all the symbols or bit statistics but also the channel matrix for each channel use within the boundaries of the code word.
6 SUMMARY AND CONCLUSIONS

In this thesis we tried to analyze and present the deferent aspects of using Group Detection based schemes to detect MIMO channels and more specifically MIMO-BICM channels. The main idea behind Group Detection is to break down the complexity of the detection and to control it by dividing the detection into smaller groups. By dividing the detection into groups we create a trade-off between performances and complexity, allowing the designer of the receiver to choose the point most appropriate to his application and resources.

We reviewed several factors beside Group Size which affect the performances of GD schemes including: Group Overlap, Information feedback, Ordering Resolution and others. We focused our discussion to three basic schemes each representing a different tradeoff of complexity, performance and usability.

The basic GD scheme represents the simplest and most straight forward way of implementing GD schemes. By not-allowing our groups to overlap we are keeping our complexity to the bare minimum while still enjoying GD performances. The basic GD scheme however requires special attention to the subject of Group Partitioning and its performances is greatly affected by the way the groups are constructed. We presented a relatively simple scheme for approaching the optimal performances under these conditions.

The PGD scheme introduces us to the concept of “Group Diversity” and our ability to gain information by decoding the same symbol in different groups. We established the mathematical principle of this effect, and showed a simple method of utilizing it. We also showed that the scheme enjoys extra robustness while keeping potential delays to the minimum.

Finally, the novel NM-SGD scheme tried to maximize the available performance gain from the GD concept. This was done by utilizing a serial detection scheme combining both soft interference cancellation and iterative (“turbo”) processing using prior information. By maintaining the maximal Ordering Resolution and constructing the Groups in a way that will minimize the noise experienced by the group we manage to achieve both high performances and rapid convergence.
The three schemes presented provide a wide overview of the different building blocks of practical GD based detectors. We showed the versatility of the approach noting the ability of combining it with existing complexity reducing approaches such as sphere detectors and interference cancellation schemes.

In addition we would like to point out that there is still room for optimization both in the complexity of the different schemes by using adaptive approaches which change the group size and the interference cancellation/suppression to meet the actual needs of the current symbol or group. And also by improving the way different detectors are summed together for the “Group Diversity” effect.
APPENDIX A – MANAGING SYMBOL STATISTICS IN NM-SGD

In this section we will go into more details about the way NM-SGD manages it symbol statistics. This is an important implementation issue which affects both the efficiency and the performances of the algorithm.

To begin with, it is important to note the two kinds of basic statistic values we have in our scheme. The first is the L value or LLR value defined as:

\[
LLR[b_n] = \log \frac{P(b_n = 1)}{P(b_n = 0)}
\]  \hspace{1cm} (A.1)

where \( b_n \) is the \( n \)th bit of the particular symbol being processed. This is obviously a bit level statistics, and it can take any value from \(-\infty\) to \(\infty\). Our second type of statistic relates to the probability of a particular symbol taking on a particular value:

\[
P_i[s] = P(s = S_i)
\]  \hspace{1cm} (A.2)

For our symbol constellation \( \mathbb{C} \) of size \( 2^{N_b} \), \( S_i \) represents a particular symbol value from our constellation \((S_i \in \mathbb{C}),\ 0 < i < 2^{N_b} \) and \( 0 \leq P_i[s] \leq 1 \). We will refer to the form of statistics represented in equation (A.2) as the explicit form.

When we want to represent the APP detector decision for a particular transmitted symbol \( s \) we have two choices: either maintain \( N_b \) L values, representing the possible bit values of the symbol bit vector, or maintain \( 2^{N_b} \) symbol statistics in their explicit form. These two forms of statistics are equivalent and, as the following section demonstrate, exchangeable.

At first glance it would seem redundant to maintain the symbol statistics in their explicit form since it is clear that the LLR form is much more memory efficient. However, on closer examination of the NM-SGD algorithm, as listed in section 0, we can see that the explicit form of the symbol statistics is necessary in order to calculate the following essential values:
Complexity and Delay Analysis

\[ E[s] = \sum_{i=1}^{N_B} p_i S_i \]  
(A.3)

\[ \text{Var}[s] = E[(s - E[s])^2] = E[s^2] - E[s]^2 = \sum_{i=1}^{N_B} p_i S_i^2 - E[s]^2 \]  
(A.4)

Farther more, by examining the algorithm, we realize that all our calculations involve symbols rather than bits. The only situations we are interested in bits is when we are preparing our final output to the interleaver which obviously works at the bit level. Consequently it is clearly much more efficient to maintain a table of the explicit symbol probabilities and only use LLR values at the beginning, where we get our prior bit data, and at the end, where we output our new extrinsic values.

The calculation of the LLR values from the symbol statistics was already demonstrated in section 2.4 so we will only focus our development here on the calculation of symbol statistics from the LLR values:

\[ P_i[s] = P(s = S_i) = \prod_{n=1}^{N_S} P\left(b_n = b_n^S\right) = \prod_{h_i=0}^{e^{LLR\left(b_n\right)}} \frac{1}{1 + e^{LLR\left(b_n\right)}} \cdot \prod_{h_i=0}^{e^{LLR\left(b_n\right)}} \frac{1}{1 + e^{LLR\left(b_n\right)}} = 1 \cdot \frac{\sum_{h_i=1}^{e^{LLR\left(b_n\right)}} e^{LLR\left(b_n\right)}}{\prod_{n=1}^{N_S} 1 + e^{LLR\left(b_n\right)}} \]  
(A.5)

where \( b_n^S \) is the \( n \)th bit of the \( S_i \) symbol, and the second line is a direct result of the definition of the LLR (equation (A.1)). We can see that the denominator of the final term of (A.5) does not depend on \( S_i \) and would be the same for the possible symbols.

In order to simplify processing farther, and for the use of regular MAP estimation algorithms such as Max-Log it is common to use the log form of the symbol statistics rather than the explicit form used in equation (A.2). This, however, has little effect on our development.
BIBLIOGRAPHY


אבסטרקט

בעבודה זו אנו חוקרים את המאפיינים השונים של מימוש פענוח בקבוצות עבור ערוצי מימונים מרובי אנטנות (MIMO). סיבוכיות פענוח בערוצי MIMO גדלה בצורה אקספוננציאלית ביחס למספר האנטנות והбитים המאופננים בסימבול. אף על פי שהMITTEDを持っている את כל הלוח fragment של פענוח השפה, פענוח בקבוצות היא טכניקה מקורית המאפשרת הפחתה ושליטה בסיבוכיות של המפענח על ידי פירוק הבעיה וחלוקתה למספר קבוצות בלתי תלויות כאשר כל אחת מהן מפוענחת בנפרד. פענוח בקבוצות מאפשר לochondーンליסט ביקורת על_balance בין סיבוכיות המפענח לבין איכותו ונותן לו לבחור בנקודה המתאימה ביותר למערכת שלו.

במהלך עבודה זו ננסה לחקור ולאפיין את צורות המימוש האפשריות של פענוח בקבוצות, ואת ההשפעה של פרמטרים שונים על הביצועים. נחקור את השפעת צורת החלוקה לקבוצות ושל הסדר פענוח הקבוצות, ולאלגוריתם Larsen והוא המנשה למקסס את ביצועי פענוח בקבוצות.

אני שים בקריטירווגים שונים עבר ויינת קבוצות הסידור.
הפקולטה להנדסה ע״ש יוסי ואלדר פליישמן
בית הספר לתארים מתקדמים ע״ש זנדמן-סליינר

אלגוריתמים לפענוח מופחת סיבוכיות
של ערוצים מורבי גליסות ויציאות
המזרונים על FUNOVA בקברוזות

חיבר: יהודה גרשטנברג גמר לקראת תואר "מוסמך חינכתי" בהנדסת חשמל ואלקטרוניקה
ע״י

יוחאי זמורה

העבודה נקראה במשכת המחלקה להנדסה חשמל – מערColumnInfo

נсим טסית"ח
הפקולטה להנדסה עד"ש איבי ואלדר פליישמן
בית הספר לתארים מתקדמים עד"ש זנדמן-סליינר

אלגוריתמים לפענוח מופחת סיבוכיות
של ערוצים מרובי Giriş שicients ויציאות
המובילים על פענוח בקבוצות

יח norske 멜 hydrogen נמר克拉חאת תואר " المقدس אוניברסיטת" הובדש התשמ"א אלקטרוניקה

יוחאי זמורה

ניס היפות"ח