Energy Efficient Design of an Adaptive Switching Algorithm for the Iterative-MIMO Receiver

Noor Zahrinah Binti Mohd Tadza (Nina)

A thesis submitted for the degree of Doctor of Philosophy.
The University of Edinburgh.
May 2015

Abstract

An efficient design dedicated for iterative-multiple-input multiple-output (MIMO) receiver systems is now imperative in our world since data demands are increasing tremendously in wireless networks. This puts a massive burden on the signal processing power especially in small receiver systems where power sources are often shared or limited. This thesis proposes an attractive solution to both the wireless signal processing and the architectural implementation design sides of the problem. A novel algorithm, dubbed the **Adaptive Switching Algorithm**, is proven to not only save more than a third of the energy consumption in the algorithmic design, but is also able to achieve an energy reduction of more than 50% in terms of processing power when the design is mapped onto state-of-the-art programmable hardware. Simulations are based in Matlab™ using the Monte Carlo approach, where multiple additive white Gaussian noise (AWGN) and Rayleigh fading channels for both fast and slow fading environments were investigated. The software selects the appropriate detection algorithm depending on the current channel conditions. The design for the hardware is based on the latest field programmable gate arrays (FPGA) hardware from Xilinx®, specifically the Virtex-5 and Virtex-7 chipsets. They were chosen during the experimental phase to verify the results in order to examine trends for energy consumption in the proposed algorithm design. Savings come from dynamic allocation of the hardware resources by implementing power minimization techniques depending on the processing requirements of the system. Having demonstrated the feasibility of the algorithm in controlled environments, realistic channel conditions were simulated using spatially correlated MIMO channels to test the algorithm's readiness for real-world deployment. The proposed algorithm is placed in both the MIMO detector and the iterative-decoder blocks of the receiver. When the final full receiver design setup is implemented, it shows that the key to energy saving lies in the fact that both software and hardware components of the **Adaptive Switching Algorithm** adopt adaptivity in the respective designs. The detector saves energy by selecting suitable detection schemes while the decoder provides adaptivity by limiting the number of decoding iterations, both of which are updated in real-time. The overall receiver can achieve more than 70% energy savings in comparison to state-of-the-art iterative-MIMO receivers and thus it can be concluded that this level of 'intelligence' is an important direction towards a more efficient iterative-MIMO receiver designs in the future.

Contents

Chapter 1
Introduction

Wireless communication has become the fastest growing segment of the communications industry. It has gone through remarkable advancement in the 20^{th} century and along with it, electronic circuit design is also progressing at an exponential rate. Recent innovations in wireless communication technology and computing have led to the current proliferation of devices, each with specific applications, form factor, functionality and battery lifetime. The explosive growth in wireless systems coupled with the proliferation of electronics devices indicate a bright future for wireless networks, both as stand-alone and as a part of a larger networking infrastructure. However, many technical challenges remain in designing robust wireless networks and devices that deliver the performance necessary to support emerging applications. One major challenge materializes in the form of power. With approximately 14 billion electronic devices are connected online; personal ones, such as mobile phones, laptops, set-top boxes, modems, and/or on a larger scale; base stations, wireless hotspots and femtocells, the communication sector has become one power hungry industry. The devices are estimated to waste around US$ 80 billion each year due to inefficient designs. This trend could lead to an estimated loss of around US$ 120 billion by the end of 2020 [1]. Therefore, solutions are sought to overcome the current predicament. This introductory chapter provides a brief review of wireless communications and describes the motivation behind the work that has been undertaken, the technical challenges, and finally the possible contributions this work aims to accomplish.

1.1 Motivation of Work

Due to the large number of devices available, just by reconfiguring the design for each individual device chipsets to be more efficient, would have tremendous impact on the global energy usage. With the adoption of best available technologies, chipsets are able to possess a higher degree of software and hardware flexibility to be more efficient in radio systems. It is said that such devices could perform exactly the same tasks while consuming around 65% less power [1]. Therefore, motivation of this work is to tackle the power consumption problem head on starting from each individual device.

There are two sides to the coin, the **wireless communication** side, which deals with the tremendous data demands, and the other, the **computer architecture** side, where a more efficient implementation is sought for better hardware deployment. On the **wireless communication** side, traffic volume according to regions as depicted in Figure 1.1, taken from the report in [1], shows that data demand is increasing over the years. It is predicted that by the end of 2017, with the fastest growing inclination, the data for Asia Pacific will be more than triple, reaching to about 45 exabyte (EB) in just 5 years. In other regions, demands are also rising year by year. The total world demand for data per year amounts to more than 120 EB per month.

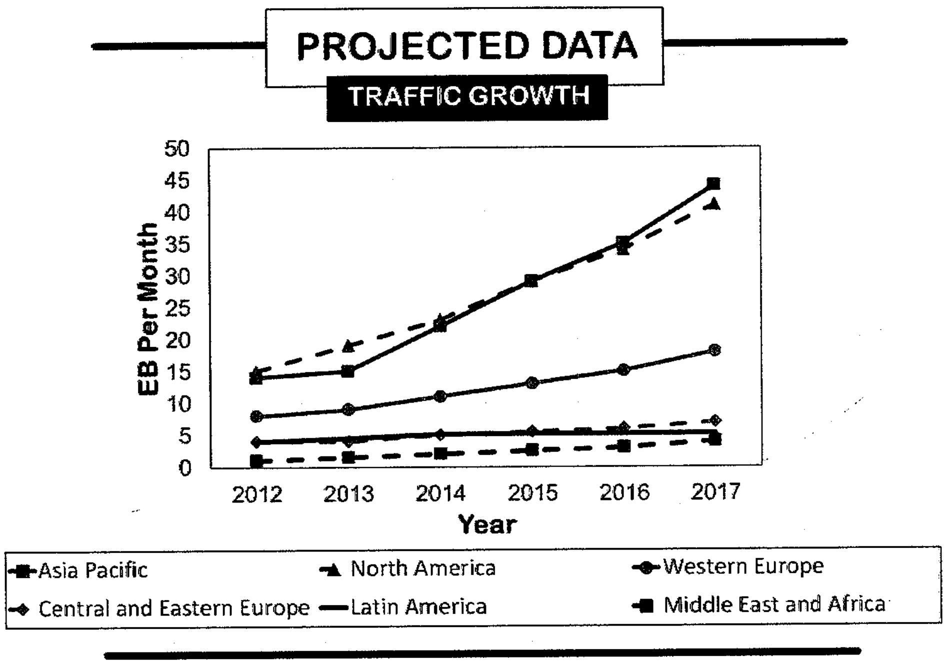

Figure 1.1: *Projected data traffic growth*

In order to cater for this trend in data demand, a significant breakthrough came in the late 1980s when the adaptive use of multiple-input multiple-output (MIMO) antenna systems was proposed. By using multiple antennas at both transmit and receive sides, parallel channels that utilize the same radio spectrum space can be created. MIMO manipulates this to increase the capacity of a channel so more data can be transmitted at one time. While minimizing power usage in these devices in wireless networks is imperative, more priority is given to the receivers since they handle massive computation processing. With billions of devices available, the total power consumption would be massive. Moreover, the receivers are usually limited in power source where they are operated using a battery, which has a limited lifetime. This brings us to the subject of **computer architecture**. Future wireless receivers aim at supporting a wide variety of wireless communication standards, such as the Long-Term Evolution (LTE), Universal Mobile Telecommunications System (UMTS), wireless local area network (WLAN), and Global System for Mobile (GSM). Key enabling technology for the enormous success of wireless communication is the progress in integrated circuit (IC) technology. It started in the late 1950s with the production of the first metal-oxide-semiconductor field-effect transistors (MOSFET) and with the idea of complementary metal-oxide-semiconductor (CMOS) circuits [2]. IC follows the trend given by Moore's law, which states that the number of transistors in a dense integrated circuit has doubled approximately every two years. Electronic design automation (EDA) software tools help handle larger and faster chips, fabrication technologies for supporting new technology nodes, and verification strategies for the increased circuit complexity. The progress in CMOS IC technology made it possible to pack more and more transistors onto the same area of silicon. This progress allowed to realize increasingly complex functions on a small piece of silicon. With this, the realization of a fast Fourier transform (FFT), a real-time detection and decoding algorithms, or an entire wireless baseband processor on a single chip became feasible.

Figure 1.2 shows the potential energy savings that can be achieved with growing technology in programming and IC circuitry. It depicts the proportion of savings that can be accomplished to compute a given operation, and that the devices of today do not fully reap these benefits in the designs. By the year 2015, just by implementing power minimization techniques to evoke a more efficient hardware design, 70% of potential energy savings can be gained, and this trend continues to rise up to a point where, in 2025, it is predicted that around 87% of energy usage can be conserved if more efficient designs are implemented in these devices. In order to have a more efficient design, flexible software and hardware implementation are needed for the whole receiver. To achieve this flexibility, the processor circuit and signal processing software need to have certain adaptivity whereby they possess a level of 'intelligence'. In principle, this would allow the exchange between transmission standards and algorithms at boot or even dynamically at run-time. This could be in the form of a system that is able to adapt to the detection algorithm on-the-fly to the current operating scenario according to the requests of the system. Current radio communication devices have incorporated digital signal processing (DSP)-based programmability for some receiver blocks. However, many computationally intensive parts still require dedicated hardware for performance and efficiency reasons. This issue is particularly crucial for MIMO transceivers, where the volume of incoming data is multi-fold, and therefore the energy required to process would be immensely large.

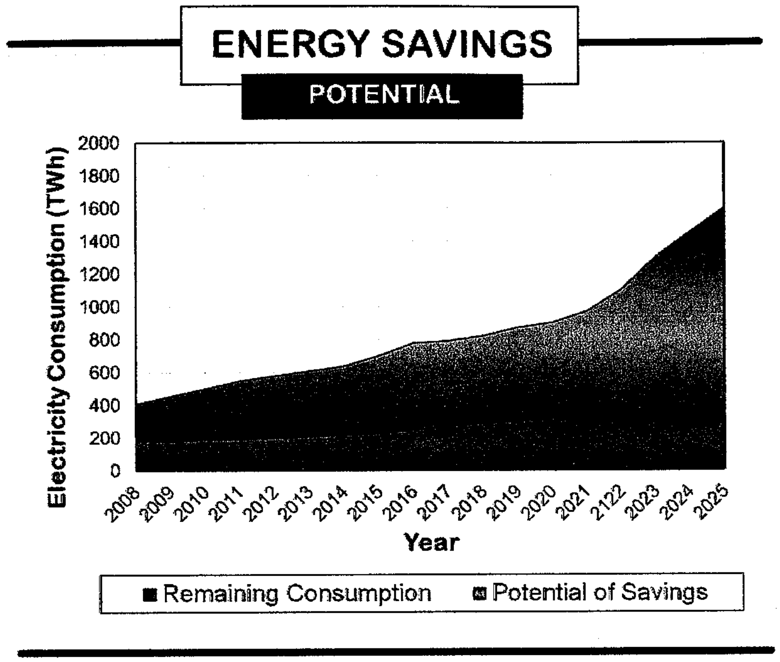

Figure 1.2: *Potential energy savings trend [1]*

This aspect of **computer architecture** and the power management schemes have not been fully exploited. Even though the technology exists, several power minimization techniques are not properly optimized on devices that support MIMO. This thesis therefore proposes a more efficient design for a receiver that rivals the state-of-the-art available in the market today. With the combination of both fields of knowledge, another setback to take into account when designing an efficient hardware capable of transmitting large amounts of data is that when a signal propagates through a wireless channel, it experiences random fluctuations in time if the transmitter or receiver is moving, due to changing reflections and attenuations. Thus, the characteristics of the channel appear to change randomly with time, which makes it difficult to design reliable systems with guaranteed performance. This is imperative to keep in mind in order to confirm the applicablity of the new design in realistic situations.

In summary, technological advances in the following areas are needed to overcome the challenges this work aims to tackle:

- Algorithmic design for the MIMO detection and decoding algorithms that support efficiency in implementations.

- Hardware design suitable for low-power handheld computer and communication receiver terminals, which can be implemented on current and future communication systems.

- Measurements and models for wireless indoor and outdoor channels in order to verify the design suited for real-life deployment.

Given these requirements, the work draws from many areas of expertise, which includes the area of communications, signal processing, software and hardware design, and power management schemes. Moreover, given the fundamental limitations of the wireless channels and the explosive demand for its utilization, communication between these interdisciplinary groups is necessary to implement the most rudimentary shell for the thesis work.

1.2 Thesis Contributions

The objective of this work is to design an efficient iterative-MIMO receiver fit for current and upcoming wireless communication standards. The main contributions of this work are distributed in three separate chapters. The chapters integrate into one another to culminate in achieving the main objective of the thesis, which is to design an efficient adaptive algorithm that possesses a level of 'intelligence' for iterative-MIMO receivers. Each stage of the work leads to the next logical progression from experimental to design practicality, as detailed below:

- An Adaptive Switching Algorithm that adapts to real-time channel conditions to minimize the power and energy consumption of iterative-MIMO detection systems is proposed. This is realized in the form of a threshold control unit, which selects the minimum complexity detector capable of meeting the desired bit-error-rate (BER) performance. The adaptive algorithm shows promising BER performance on par with the current available detection schemes with lower resource utilization. An evaluation of the new algorithmic design shows convincing dynamic and static power savings compared to baseline detectors.

- Realistic power and energy saving trends of the Adaptive Switching Algorithm are computed for the chosen hardware circuitry. Detailed power and energy analysis and the assessment of potential benefits of specific power minimization techniques show more promising results compared to the others. The combination of both the algorithmic design and the hardware design adaptivity results in tremendous gains in the overall proposed design.

- The performance of the Adaptive Switching Algorithm in realistic conditions shows significant power and energy savings with slight BER degradation. The proposed algorithm is suitable to be used as a link between the detector and iterative decoder blocks in the receiver, as a stopping criteria tool to help determine the number of decoding iterations needed per transmission. Hardware design implementation for the proposed algorithm maintains the performance of the Adaptive Switching Algorithm total receiver design in spatially correlated channels with a lower hardware utilization complexity to boot.

1.3 Thesis Outline

The thesis is structured into several chapters covering different stages of the work, following a logical flow of information, starting with the development from theoretical concepts and continuing on with the three main contributions of the research; the proposed Adaptive Switching Algorithm, the design performance of the proposed algorithm on hardware and finally, the performance of the hardware design in realistic channel conditions to test its readiness for real world applicability. The structure of each chapter is described below:

Chapter 2 is divided into two parts, viz. the **wireless communication** and the **computer architecture**. The **wireless communication** part explains the total iterative-MIMO systems and provides additional background on the detecting and decoding techniques. For a reader who is familiar with modern wireless communication systems, this part will serve mainly as a refresher as it introduces the concept of MIMO systems that provides the foundation of the research. The **computer architecture** part presents the different hardware types available and various power minimization techniques labeled as state-of-the-art, each of which promises significant power savings. The combination of the two fields of knowledge provides the comprehensive understanding required as basis for the work described in this thesis.

The proposed novel innovation of the Adaptive Switching Algorithm introduced in **Chapter 3** proves to be suitable for the sole purpose of saving power and energy consumption of the overall receivers in both slow and fast fading environments. The algorithm works by switching between thresholds pre-calculated between the transmitters and receivers during each transmission in real-time. This novel idea is the first of its kind to produce an 'intelligent' system based on switching from a high to a low complexity detector, exploiting full information of the current channel conditions of a MIMO system. The adaptivity shows that promising savings can be gained in comparison to non-adaptive iterative-MIMO detectors.

Having shown the potential power and energy savings that can be achieved within the receiver design with the proposed algorithmic design of the Adaptive Switching Algorithm, the next stage of work as described in **Chapter 4** extends those findings by incorporating the novel idea of the Adaptive Switching Algorithm onto hardware design, to promote its applicability in implementations as well. With efficient design, the proposed algorithm shows that significant power and energy savings can be gained when different power minimization techniques are utilized. A comprehensive power and energy performance analysis of the Adaptive Switching Algorithm is investigated for the iterative-MIMO systems, with the primary goal of minimizing additional power and energy consumption within the receiver. The work is then extended to examine the potential benefits of several power minimization techniques during the implementation of the Adaptive Switching Algorithm. An in depth investigation shows that power and energy usage can be further optimized when the design for the proposed algorithm is designed on state-of-the-art hardware.

After having demonstrated in the preceding chapters that the Adaptive Switching Algorithm could save significant complexity, power and energy consumption in both algorithmic and hardware design implementation in experimentally controlled conditions, its effectiveness in real-world situations is then verified in **Chapter 5**, whereby the proposed algorithm is executed under spatially correlated channel conditions. The performance of the Adaptive Switching Algorithm in these channel conditions shows that significant energy savings can be gained with slight BER degradation as the correlation between the transmitters and receivers increases. The chapter describes how forwarding the proposed algorithm threshold information to the decoder, which by providing the same necessary information used in the detector as a stopping criteria for the decoder, helps limit the number of iteration(s) required during each transmission. Significant power and energy savings are achieved for the full Adaptive Switching Algorithm receiver in comparison to state-of-the-art hardware, with lower hardware utilization complexity to boot.

The concluding remarks about this work, as presented in **Chapter 6**, enumerates the major contributions while identifying the novel aspects and improvements in comparison to other research that has been carried out in the same area. Special attention is also paid to the specific areas that could potentially be studied in future work. An appendix that contains a list of publications originating from this work is attached and included as references throughout the thesis.

Chapter 2
Background

2.1 Chapter Contribution

The work described in this thesis evolves around designing an efficient iterative-MIMO receiver that is suitable for state-of-the-art wireless communication standards. This chapter aims to provide comprehensive knowledge in the areas of **wireless communications** for software design and **computer architecture** for the hardware design implementation. The combination of each field of specialization gives the background information required to help the reader in understanding the nature of the work. The chapter begins by introducing the **wireless communication** system under consideration and the blocks within the iterative-MIMO systems i.e. the detector and the decoder. After a brief description regarding each block, the chapter progresses to the other area of specialization, namely the **computer architecture**. Several power minimization techniques in hardware are discussed in detail to shed light on the state-of-the-art methods currently available in the market. The chapter concludes by summarizing the chosen methods in this thesis for detecting and decoding and the reason behind them. It also pinpoints the best power minimization techniques to investigate in this study. Both information will lead to better understanding of the upcoming technical chapters.

2.2 Wireless Communication

Wireless communication is the transfer of information between two or more points that are not connected by an electrical conductor. The most common wireless technologies use radio. Figure 2.1 illustrates the different antenna configurations for wireless communication links. Single-input single-output (SISO), shown in Figure 2.1(a) is effectively a standard radio channel. This type of configuration has one transmitter and one receiver. Due to its simplicity, SISO requires no extra processing for manipulating the diversity that may be used. The disadvantage of SISO is that it is vulnerable to interference and fading. Moreover, the throughput is dependent on the channel bandwidth and the signal-to-noise ratio (SNR), which means it is bounded by Shannon's law. The single-input multiple-output (SIMO) version is depicted in Figure 2.1(b) and the multiple-input single-output (MISO) is shown in Figure 2.1(c). Due to the usage of multiple antennas, there are several advantages that can be gained when compared to their SISO counterpart. SIMO or MISO is able to increase the receive SNR by coherently combining the wireless signals to achieve **array gain**. Moreover, **diversity gain**, which can be classified as transmit or received diversity, are used to combat fading. The receive diversity does this by enabling the receiver to receive signals from a number of independent channels. Transmit diversity on the other hand, generates redundant data from the multiple transmitters for the one receiver to choose from. This is when the signal is transmitted over multiple (ideally) independent fading paths in time, frequency, or space. This allows the receiver to select the optimum signal to extract the required data. The advantages of using multiple transmitters are that it creates redundancy in coding and moves processing from the receiver to the transmitter. This is highly beneficial for the receiver. The lower processing requirement, which leads to lower power consumption, will have a positive impact on the size needed for multiple antennas, as well as the cost and battery lifetime. In addition, the usage of multiple antennas exploits the spatial dimension to increase the separation between users by directing signal energy towards the intended user. This is **interference reduction**. Lastly, **spatial multiplexing gain** in the multiple antenna setup provides additional data capacity by utilizing the different paths to increase the data throughput capability [3] [4] [5].

By combining the configurations, MIMO may exploit all the advantages provided by the configurations of others [6], from the aforementioned techniques of **array gain**, **diversity gain**, **spatial multiplexing gain** and **interference reduction**. MIMO, as illustrated in Figure 2.1(d), uses multiple antennas at both the transmitters and receivers. It enables a variety of signal paths to carry the data, choosing separate paths for each antenna to enable multiple signal paths to be used. It is found that the signal can take many paths between a transmitter and a receiver. Additionally, by moving the antennas even by a small distance, the paths used by the signal will change. The variety of paths available occurs as a result of the number of objects that appear to the side or even in the direct path between the transmitter and receiver. By using MIMO, these additional paths provide additional robustness to the radio link by improving the SNR, or by increasing the link data capacity. As a result, it is able to considerably increase the capacity of a given channel by increasing the number of receive and transmit antennas. MIMO increases the throughput of the channel linearly with every pair of antennas added to the system. Moreover, as spectral bandwidth is becoming an ever more valuable commodity for radio

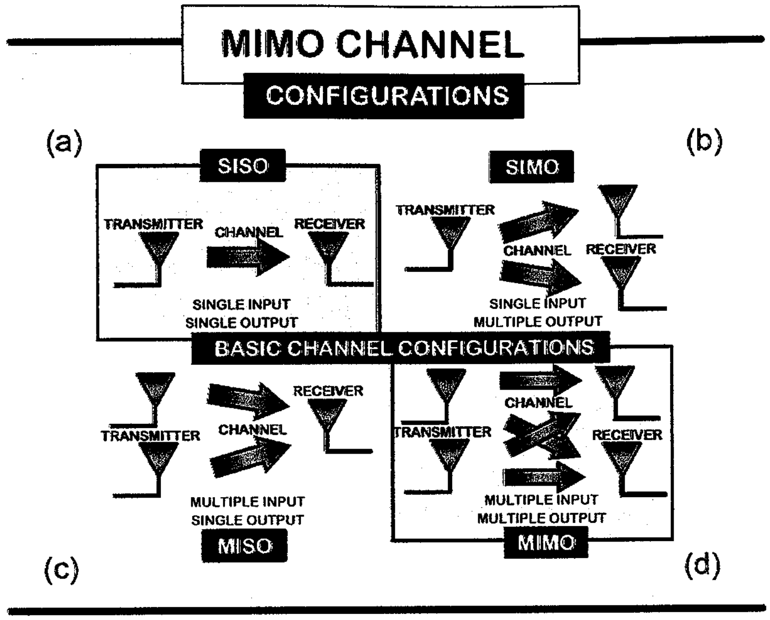

Figure 2.1: *Channel transmission configurations*

communications systems, MIMO is one of the techniques needed to properly exploit available bandwidth more effectively as well. Hence, depending on the purpose of the MIMO system, an appropriate trade-off needs to be found. Due to the increasing demand of data mentioned in the previous chapter, **spatial multiplexing** provides the capacity to cater for this need. The aim of this work is therefore, to find the right trade-off in a system that incorporates **spatial multiplexing**, between the complexity or power consumption and the system performance.

2.2.1 Iterative-MIMO System Architecture

A typical iterative-MIMO architecture is illustrated in Figure 2.2. An in-depth explanation of the full iterative-MIMO system can be found in the next section, however, as an overview, the system can be partitioned into three segments; the **transmitter**, the **channel** and the **receiver**. The **transmitter** is made up of several components. The hard data bits, u, first go through the channel encoder. The channel encoder appends extra data bits to make the data transmission more robust to interferences on the transmission channel. There are many coding schemes available and they can basically be categorized into two major types; linear block codes and convolutional codes. In a typical iterative-MIMO system, the latter is used, specifically the turbo encoder, where two convolutional codes are used in parallel with some kind of interleaving in between. This gives the encoded e bits, which are interleaved. These are being passed through to the constellation modulator where the bits are mapped onto a digital scheme such as the quadrature amplitude modulation (QAM) or the phase-shift keying (PSK). By representing the transmitted bits a as a complex number and modulating a cosine or sine carrier signal with real ($\Re$) and imaginary ($\Im$) parts respectively, the symbols can be sent with two carriers on the same frequency. Once the symbols are modulated, they are split into several streams depending on the number of transmitters used before being transmitted over a **channel**. The transmission **channel** is essentially a path between two nodes in a network.

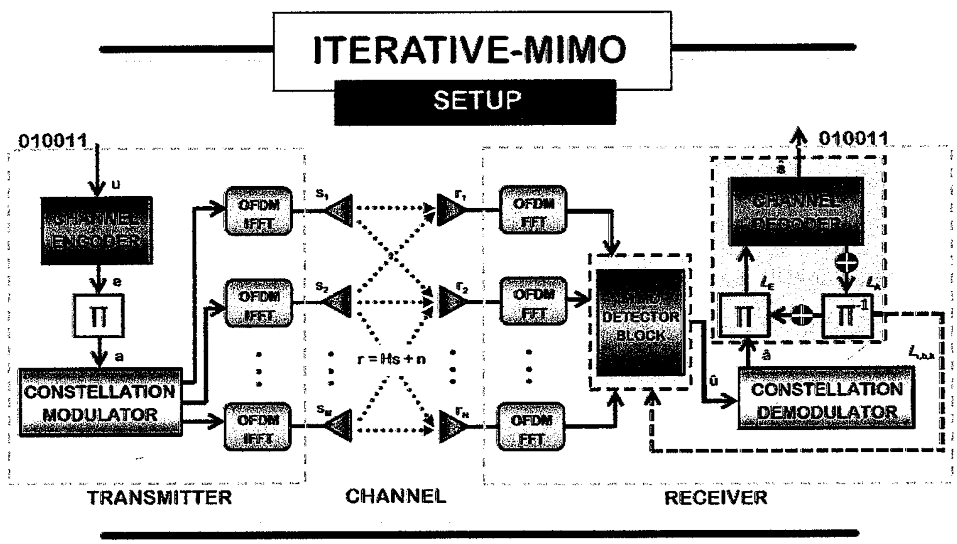

Figure 2.2: *Iterative-MIMO system channel*

Consider a spatial multiplexing MIMO-orthogonal frequency-division multiplexing (OFDM) system with M transmitters, N receivers, and $M \geq N$. The channel can be represented by the matrix described in Equation (2.1).

$$\mathbf{r} = \mathbf{H}\mathbf{s} + \mathbf{n} \tag{2.1}$$

where the channel matrix $\mathbf{H} \in \mathbb{C}^{M \times N}$ with independent elements $h_{i,j} \sim \mathcal{CN}(\mu, \sigma^2)$, for $1 \leq i \leq M$ and $1 \leq j \leq N$ representing a block fading propagation environment, with $\mu = 0$ and $\sigma^2 = 1$, $\mathbf{s} = (s_1, s_2, \ldots, s_M)^{\mathrm{T}}$ is the transpose vector of the M-dimensional transmit symbol vector with $E[| \, s_i \, |^2] = M^{-1}$, $\mathbf{n}$ is the $\mathbb{C}^{N \times 1}$ additive independent and identically distributed (i.i.d.) circular symmetric complex Gaussian noise vector normalized so that its covariance matrix is the identity matrix, i.e. $\mathbf{n} \sim (0, N_0 \mathbf{I}_N)$ of $h_{i,j} \sim \mathcal{CN}(0, N_0)$ and $\mathbf{r} = (r_1, r_2, \ldots, r_N)^{\mathrm{T}}$ is the transpose N-vector of received symbols. Throughout this thesis, the SNR is defined as the average SNR per receive antenna according to Equation (2.2).

$$\mathrm{SNR} = \frac{M \mathbf{E}_s}{N_0} \tag{2.2}$$

where $\mathbf{E}_s$ is the energy per transmit symbol s. The received symbols, $\mathbf{r}$, are then processed by the **receiver**. From Figure 2.2, first, the symbols are multiplexed into a single stream before being detected by the MIMO detector to give $\hat{\mathbf{u}}$ bit streams.

In the **receiver**, the detection can be solved in many ways. In order to optimally solve the MIMO detection problem, an exhaustive search for the best solutions can be performed over all signal constellations. The number of possible signal constellations increases exponentially with the number of antennas and the number of bits per modulation symbol. Maximum-Likelihood (ML) detection finds the minimum constellation point in Equation (2.1) within the received symbols. It is given by:

$$\hat{\mathbf{s}}_{ML} = \arg\min_{\mathbf{s} \in \mathcal{O}^M} \| \, \mathbf{r} - \mathbf{H}\mathbf{s} \, \|^2 \tag{2.3}$$

where $\mathcal{O}$ denotes the constellation size of a specific modulation. The ML detector is optimal and fully exploits all available degree of freedom. Even though ML produces the best BER performance, due to its use of exhaustive search, it can have immense complexity for direct implementation. The complexity grows exponentially with the transmission rate φ, since the detector needs to go through 2^{φ} hypotheses for each received vector. For example, for the case of a 4×4 iterative-MIMO system employing 16-QAM, the detector would need to search a total of $S = 16^4 = 65{,}536$ candidates in order to find the correct transmitted vector. For 64-QAM, this number rises to more than $S = 64^4 = 16{,}777{,}216$. This makes an exhaustive search infeasible for a hardware implementation [7]. As the optimal exhaustive search is far too complex for hardware implementations, many suboptimal detection algorithms exist with a big range in communications performance and complexity. Several efficient suboptimal detection techniques have therefore been proposed or adapted from the field of multi-user detection.

Even though these techniques are much less computationally demanding than the ML detector, they are often unable to exploit a large part of the available degree of freedom, and thus, their performance tends to be significantly poorer than that of ML detection. However, this trade-off can be made for efficient hardware designs.

Back to Figure 2.2, after the detection, the symbols are then forwarded to the constellation demodulator where the symbols are demapped to get $\hat{\mathbf{a}}$ before going to the turbo decoder, with two constituent decoders working together with deinterleavers in between them. This iterative decoders then produce the hard output for the received symbol bits. Within the **receiver** is where the focus of the work lies. This involves around minimizing power and energy consumption within the iterative-MIMO receiver, particularly, by re-designing the MIMO detector and the iterative decoder parts of the system. The sections below explain different types of detectors and decoders available, and their advantages and disadvantages are highlighted to showcase parts that need to be improved for a better performance in power and/or energy consumption. Finding the right trade-off between communications performance with implementation complexity, and understanding the implications on the whole receiver is one of the major challenges in the design of iterative-MIMO receivers.

2.2.2 MIMO Detectors

MIMO detection algorithms can be seen as a "tree search" problem, as shown in Figure 2.3. This is realized by inverting the channel matrix $\mathbf{H}$ using the $\mathbf{QR}$-decomposition to decompose matrix $\mathbf{H}$ into a unitary matrix $\mathbf{Q}$ of dimension $M \times M$ and an upper-triangular matrix $\mathbf{R}$ of dimension $M \times N$ according to:

$$\mathbf{H} = \mathbf{QR} \tag{2.4}$$

The system model in Equation (2.1) can be left-multiplied by the Hermitian transpose of $\mathbf{Q}$, which is the $\mathbf{Q}^H$, to give:

$$\hat{\mathbf{y}} \triangleq \mathbf{Q}^H\mathbf{r} = \mathbf{Rs} + \mathbf{n} \tag{2.5}$$

When the problem is visualized as a "tree search", the ML detection rule as given in Equation

(2.3) can be approximated as:

$$\hat{\mathbf{s}}_{ML} \approx \arg\min_{\mathbf{s} \in \mathcal{O}^M} \| \hat{\mathbf{y}} - \mathbf{R}\mathbf{s} \|^2 \tag{2.6}$$

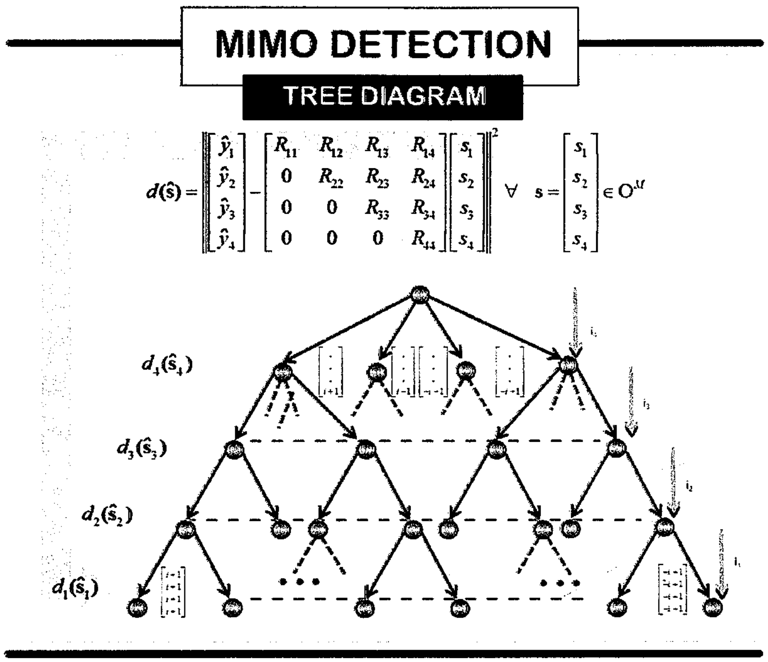

Figure 2.3: *MIMO detection as a tree diagram for* 4*-QAM modulation on a* 4×4 *MIMO system*

Figure 2.3 depicts the search traversing down level i, looking through j nodes until the solution is found, where the $\mathcal{O}$ is the number of constellation points in respective modulation scheme. Since $\mathbf{R}$ is upper-triangular, the minimization in Equation (2.3) corresponds to a "tree search" problem, where the nodes on level i are associated with a partial symbol vector $\mathbf{s} = [s_i, ..., s_M]^{\mathrm{T}}$ and with a corresponding squared partial Euclidean distance (ED), $d_i(\mathbf{s})$. The squared partial ED is given by:

$$d_i(\mathbf{s}_i) = d_{i+1}(\mathbf{s}_{i+1}) + |D_i(\mathbf{s}_i)|^2 \tag{2.7}$$

with $i = M, M-1, ..., 1$. The distance increments $|D_i(\mathbf{s}_i)|^2$ are computed as:

$$|D_i(\mathbf{s}_i)|^2 = |\hat{y}_i - \sum_{j=i}^{M} R_{ij}s_j|^2 \tag{2.8}$$

Therefore, the squared ED for the ML solution is given as:

$$d_{ML} = \min_{\mathbf{s}\in\mathcal{O}^M}(d_1(\mathbf{s}_1)) \tag{2.9}$$

and the ML solution is the associated $\mathbf{s}_1$. With this illustration in mind, the task of a MIMO-detector is to find the vector $\mathbf{s}_1$ that leads to the smallest d_i, i.e. the leaf node with the smallest squared partial ED.

To this end, a vast amount of literature exists that presents algorithms and approximations to process the tree in a clever way in order to find the estimate $\hat{\mathbf{s}}$ with less computational effort than an exhaustive search. The trade-off between the different approaches consists of implementation complexity, BER performance, and throughput.

2.2.3 Hard-Output MIMO Detection

The output of a MIMO detection algorithm is either a hard-output decision (the estimate $\hat{\mathbf{s}}$), or an *a posteriori* probability (APP) for each bit of the transmitted symbol vector. The latter helps further improve the performance of a MIMO detector. This soft-output iterative-MIMO detection algorithms were introduced in [8], and will be described in the next section. A hard-output MIMO detector delivers an estimate $\hat{\mathbf{s}}$ of the transmitted symbol vector $\mathbf{s}$. Starting point is the input-output relation as given in Equation (2.1). Several algorithms exist to obtain the estimate $\hat{\mathbf{s}}$. In general, these are divided into linear detection, successive interference cancellation (SIC) detection, and ML detection methods.

2.2.3.1 Linear Detectors

A linear detector first separates the data streams with a linear filter and then decodes each stream independently. The computational complexity of linear hard-output MIMO detection is small in comparison to other detection schemes. However, the BER performance is significantly worse compared to ML detection. Examples of linear detectors are Zero Forcing (ZF) and minimum mean square error (MMSE) filters apply an inverse of the channel to the received signal in order to restore the transmitted signal [9]. These linear filters can be implemented at a low complexity, however, their performance is very low as well.

The **ZF** detector inverts the effect of the channel matrix, $\mathbf{H}$. The corresponding channel filter matrix $\mathbf{G}_{ZF}$ is given by Equation (2.10).

$$\mathbf{G}_{ZF} = (\mathbf{H}^H\mathbf{H})^{-1}\mathbf{H}^H \tag{2.10}$$

where $\mathbf{G}_{ZF}$ is the Moore-Penrose pseudoinverse of $\mathbf{H}$. Left-multiplying Equation (2.1) with $\mathbf{G}_{ZF}$ yields the ZF estimate of:

$$\hat{\mathbf{y}}_{ZF} = \mathbf{G}_{ZF}\mathbf{r} = \mathbf{s} + \mathbf{G}_{ZF}\mathbf{n} \tag{2.11}$$

to obtain the symbol-vector estimate $\hat{\mathbf{s}}$, the equalized noise $\mathbf{G}_{ZF}\mathbf{n}$ is ignored and each element of $\hat{\mathbf{y}}_{ZF}$ is mapped to the closest constellation point according to Equation (2.12).

$$\hat{s}_i = [\hat{y}_i]_O, \qquad \text{for} \quad i = 1, ..., M \tag{2.12}$$

The ZF detection removes the co-channel interference and it is the ideal detector when the channel is noiseless, i.e. $\mathbf{n} = 0$. However, in a real system, the noise is enhanced and correlated by $\mathbf{G}_{ZF}$, which is the main reason for the poor BER performance of ZF detection. This phenomenon is known as noise-enhancement [10].

The **MMSE** detector considers the noise power in the interference cancellation and therefore shows a slightly better performance. It reduces the effect of noise-enhancement by minimizing the total error, including the noise term, according to Equation (2.13).

$$\mathbf{G}_{MMSE} = \underset{\mathbf{G}\in\mathbb{C}^{M\times N}}{\arg\min} \parallel \mathbf{Gr} - \mathbf{s} \parallel^2 \tag{2.13}$$

The MMSE estimator matrix $\mathbf{G}_{MMSE}$ can be computed as in [10] to give Equation (2.14).

$$\mathbf{G}_{MMSE} = (\mathbf{H}^H\mathbf{H} + \frac{M}{\text{SNR}}\mathbf{I}_M)^{-1}\mathbf{H}^H \tag{2.14}$$

Left multiplication of Equation (2.1) by $\mathbf{G}_{MMSE}$ yields:

$$\hat{\mathbf{y}}_{MMSE} = \mathbf{G}_{MMSE}\mathbf{r} = \sqrt{\frac{N}{\mathbf{E}_s}}\mathbf{s} + \mathbf{G}_{MMSE}\mathbf{n} \tag{2.15}$$

where the term $\sqrt{\frac{N}{E_s}}$ is the mean (over fading) received energy of the signal transmitted by each antenna, which is the residual noise caused by the co-channel interference. The detection step is carried out, similar to ZF detection, by mapping $\hat{\mathbf{y}}_{MMSE}$ to the closest constellation point analogous to Equation (2.12). The MMSE detector suffers less from the noise-enhancement and therefore achieves the better BER performance in comparison to ZF detection. The computational complexity remains approximately the same as for ZF detection with the exception of the former needing an estimate on the SNR.

2.2.3.2 SIC Detectors

The SIC technique was initially adopted by the Vertical-Bell Laboratories Layered Space-Time (V-BLAST) system [3]. In contrast to the basic ZF and MMSE filters, SIC detects the transmitted streams sequentially. It chooses the substream with largest SNR and removes the interference of each detected stream before continuing the detection process. The performance of the SIC algorithm is generally better than ZF and MMSE filters. The starting point for SIC detection is the **QR**-decomposition of the system model in Equation (2.5).

The matrix, **R**, has the property of being upper-triangular and the M^{th} stream can be detected according to:

$$\hat{s}_i = \left[\frac{\hat{y}_M}{R_{M,M}}\right]_{\mathcal{O}} \tag{2.16}$$

The remaining streams are detected according to the following recursion:

$$\hat{s}_i = \left[\frac{1}{R_{i,i}}(\hat{y}_i - \sum_{j=i+1}^{M} R_{ij}\hat{s}_j)\right]_{\mathcal{O}}, \qquad \text{for} \quad i = M-1, ..., 1. \tag{2.17}$$

SIC detection resembles the procedure of ZF detection. However, the streams are processed sequentially, one after another. This allows slicing the estimate $\hat{y}_i$ to $\hat{s}_i$ immediately after its computation and using the result to cancel out its influence on the subsequent streams. SIC can be visualized as a single tree-traversal from top to bottom always selecting the node with the smallest partial ED. The symbol vector leading to the leaf node is returned as the SIC estimate.

2.2.3.3 ML Detectors

Under the assumption that all transmit symbol vectors are equally likely, ML decoding is the optimum hard-output MIMO detection method in terms of minimizing the symbol BER [10]. The task of an ML detector is to go through all the possible constellation points and level of antennas exhaustively until the minimum node with the smallest ED is found.

A **brute-force ML** detector computes the ED for all possible transmitted vector symbols. The ML solution then corresponds to the vector symbol with the smallest ED. In [11], it was shown that the implementation of the detector is feasible at a throughput of 50 megabit per second (Mbps) for a 4×4 MIMO system with quadrature phase-shift keying (QPSK) modulation, i.e. for $4^4 = 256$ possible vector symbols.

2.2.3.4 Sphere Decoding (SD)

Due to the ML detection problem complexity being extremely high, the brute force manner can also be solved by the sphere decoding (SD) algorithm. SD traverses the tree in a clever way such that the search complexity is significantly reduced by searching over only those lattice points that lie within a hypersphere of radius Φ around the received signal $\mathbf{r}$ [10]. From a "tree search" point-of-view, the ML solution corresponds to the leaf associated with the smallest ED, as shown in Equation (2.9). To find this leaf, SD traverses the tree in a depth-first manner. The hypersphere around $\mathbf{r}$ corresponds to a pruning criterion in Equation (2.18).

$$d_i(\mathbf{s}_i) < \Phi^2 \tag{2.18}$$

Complexity reduction is achieved by pruning those nodes from the tree that violate the sphere constraint. Whenever a node is computed with a partial ED, $d_i(\mathbf{s}_i) \geq \Phi^2$, that branch is pruned and no longer followed. In order to further reduce search complexity, some optimizations on algorithmic level can be applied such as **radius reduction.** The Φ is initialized to $\Phi = \infty$ in order to guarantee to find at least one leaf node. Once the first leaf node is computed, the radius is updated according to $\Phi \leftarrow d_1(\mathbf{s}_i)$. Now, whenever a new leaf is found that fulfills sphere constraint, Φ is updated again. The reduction of Φ allows for more rigorous tree pruning while still finding the ML solution and therefore leads to a reduced average number of visited nodes. Another technique of reducing complexity is **enumeration**, where each node in the tree has several child-nodes. The processing order of these child-nodes considerably influences search complexity, especially if **radius reduction** is applied. A scheme proposed by Schnorr and Euchner [12] and modified for finite lattices in [13] visits the nodes of the same parent node in ascending order of their partial EDs. SD with Schnorr-Euchner **enumeration** and **radius reduction** is usually denoted as Schnorr-Euchner SD. A drawback of SD is the variable run-time, due to variable search complexity, which renders detection latency unpredictable.

2.2.3.5 Close-to-ML Detection

The variable number of nodes that need to be visited in SD and the still considerable implementation complexity lead to a variety of algorithms that approximate the performance of SD. The price for the reduced implementation complexity or for the constant run-time is slightly worse but still close-to ML BER performance. Therefore, **reduced complexity sphere decoding** aims at decreasing the computational effort to compute a partial ED. To this end, the computation of the squared l^2-norm in Equation (2.7) is approximated by the l^1-norm or the $l^{\tilde{\infty}}$-norm, respectively [14]. The l^1-norm of a vector $\mathbf{x}$ is defined as:

$$\| \mathbf{x} \|_1 = |\Re(\mathbf{x})| + |\Im(\mathbf{x})| \tag{2.19}$$

and the $l^{\tilde{\infty}}$-norm of a vector $\mathbf{x}$ is defined as:

$$\| \mathbf{x}_{\tilde{\infty}} \|_1 = \max\{|\Re(\mathbf{x})|, |\Im(\mathbf{x})|\} \tag{2.20}$$

By application of the l^1-norm, Equation (2.8) becomes:

$$|D_i(\mathbf{s}_i)| = |\Re(D_i(\mathbf{s}_i)| + |\Im(D_i(\mathbf{s}_i)|\} \tag{2.21}$$

and the partial ED in Equation (2.7) can be computed according to:

$$d_i(\mathbf{s}_i) = d_{i+1}(\mathbf{s}_{i+1}) + |D_i(\mathbf{s}_i)| \tag{2.22}$$

With this approximation, the squaring operation in Equation (2.8) is saved, which helps to reduce both delay and circuit area in a potential implementation. For the $l^{\tilde{\infty}}$-norm, the distance increment in Equation (2.8) is computed according to:

$$|D_i(\mathbf{s}_i)| = \max\{|\Re(D_i(\mathbf{s}_i))|, |\Im(D_i(\mathbf{s}_i))| \tag{2.23}$$

and the partial ED in Equation (2.7) becomes:

$$d_i(\mathbf{s}_i = \max(d_{i+1}(\mathbf{s}_{i+1}), |D_i(\mathbf{s}_i)|)) \tag{2.24}$$

In [14], it was shown that the application of the $l^{\tilde{\infty}}$-norm is beneficial in terms of the number of visited nodes as well as in terms of circuit area and clock frequency, while the BER performance is only slightly reduced compared to ML detection performance.

The **K-Best** detector is another algorithm that provides a close-to-ML solution. The K-Best algorithm for MIMO detection was first proposed in 2002 [15]. From a "tree search" point-of-view, it resembles a breadth-first "tree search". On each level of the tree, only the K nodes with the smallest partial EDs are further extended. Compared to SD, the throughput of the K-Best algorithm is constant. However, the BER performance is slightly degraded compared to SD and strongly depends on the chosen K. The K-Best algorithm is well suited for very-large-scale-integration system (VLSI) implementation due to the regular data path and the simple control flow. Architectural transformations like pipelining and resource sharing can easily be applied.

Another algorithm for hard-output MIMO detection is the fixed-throughput **fixed-complexity sphere decoding (FSD)** algorithm [16]. It achieves close-to ML BER performance and, like the K-Best algorithm, it exhibits a constant throughput. The FSD algorithm overcomes the problem of the variable complexity and the sequential behaviour of SD by searching only over a fixed but well-defined number of lattice vectors. A common configuration is to visit all nodes on the top level (i.e., on $i = M$) and only one node per parent node on the lower levels. A decisive factor that significantly contributes to the close-to ML BER performance of FSD is the order in which the streams are processed. The ordering is determined according to the number of nodes that are visited on the same layer. On the layers where all nodes of a parent node are visited, the stream with the largest noise amplification is chosen; on the other levels, the streams are selected in ascending order of their noise-amplification. In [16], the ordering is called FSD ordering and was obtained via V-BLAST ordering and computed according to [17].

The number of operations, floating-point operations per second (FLOPS) or algebraic operations, required by a detection algorithm is expressed in the "big O" notation. However, its practical meaning may be limited. In particular, for MIMO systems of moderate size, constants and lower order contributions to the computational cost may also be relevant. Matlab™ provides counting of FLOPS. Though this technique is obsolete, it provides a general overview of the complexity of each detection algorithm, where at this stage to be sufficient. Table 2.1 tabulates the FLOPS counts for each detection algorithm using Matlab™ environment running a packet size of $1,024$ utilizing 4-QAM on a 4×4 AWGN channel. SD and K-Best algorithms have variable complexity whereby they are highly dependent on the size of the search radius Φ and the expanded node K. In this case, $\Phi = \infty$ and K is set to be 3.

High Performance			Low Complexity		
Detector	Type	KFLOPS	Detector	Type	kFLOPS
ML	Fixed	28.7	ZF	Fixed	1.7
SD	Variable	24.4	MMSE	Fixed	1.9
K-Best	Variable	21.1	SIC	Fixed	4.2
FSD	Fixed	16.8	V-BLAST/ZF	Fixed	4.8

Table 2.1: *Different algorithm complexity of MIMO detectors measured in kFLOPS*

Figure 2.4 shows the frame-error-rate (FER) curves for the addressed hard-output MIMO detection algorithms. The simulation results are for a 4×4 MIMO-OFDM system with a convolutional code rate of $\varphi = 1/2$. Each OFDM symbol consists of 64 subcarriers using 16-QAM. For the simulation results, perfect channel state information and perfect synchronization are assumed. The simulation results clearly show the large difference between hard-output low complexity linear ZF and MMSE or SIC detection and high performance K-Best and FSD in relation to the ML detection respectively. Since the algorithms of V-BLAST/ZF and FSD show similar innerworkings (FSD requires the V-BLAST ordering), in the next chapter, a slightly modified version of the FSD algorithm incorporation with the V-BLAST/ZF, is presented to be the basis of the proposed efficient algorithm.

Better BER performance can be achieved by incorporating the APP in the detection. Figure 2.5

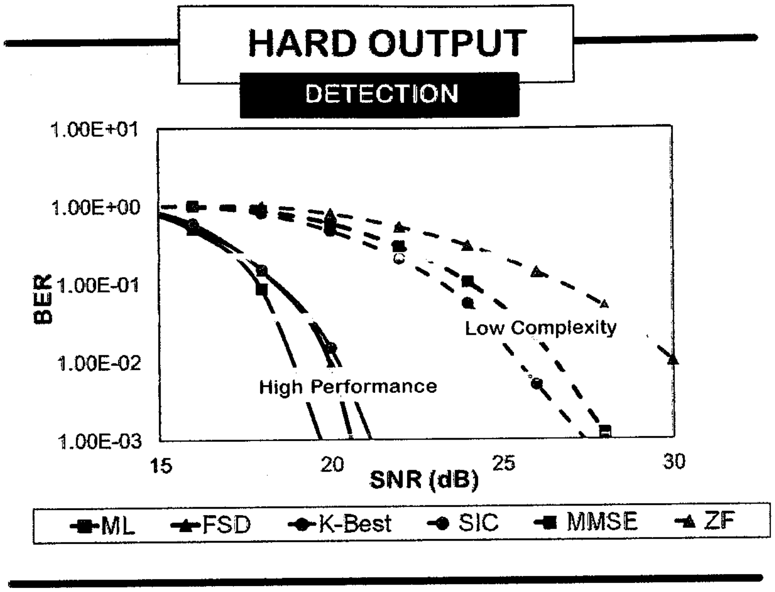

Figure 2.4: *BER performance comparison high performance and low complexity hard decoding* 16-*QAM with convolutional coding of* $\varphi = 1/2$

shows the BER performance for an optimum iterative soft-input soft-output MIMO detector with 4 iterations, for an optimum APP detector, and for an ML hard-output detector [8]. It can be seen that the BER performance for a convolutional coding with code rate of $\varphi = 1/2$ in binary phase-shift keying (BSPK) for additive white Gaussian noise (AWGN) channel shows significant improvement over the hard decoding equivalent. With an iterative-MIMO detector, the best BER performance can be achieved. However, the associated performance gains come at the cost of a substantially increased implementation complexity. This work will utilize the soft-output in the receiver.

2.2.4 Soft-Output MIMO Detection

As already shown in Figure 2.5, better BER performance in a coded MIMO-OFDM system compared to hard-output detection can be achieved by computing the APP for each hard bit, b, that associated to the transmitted symbol vector s. Therefore, the aforementioned detection algorithms have to be adjusted to utilize the given soft-input information. The APPs are usually expressed as log-likelihood ratio (LLR) [18] [19] and are computed according to:

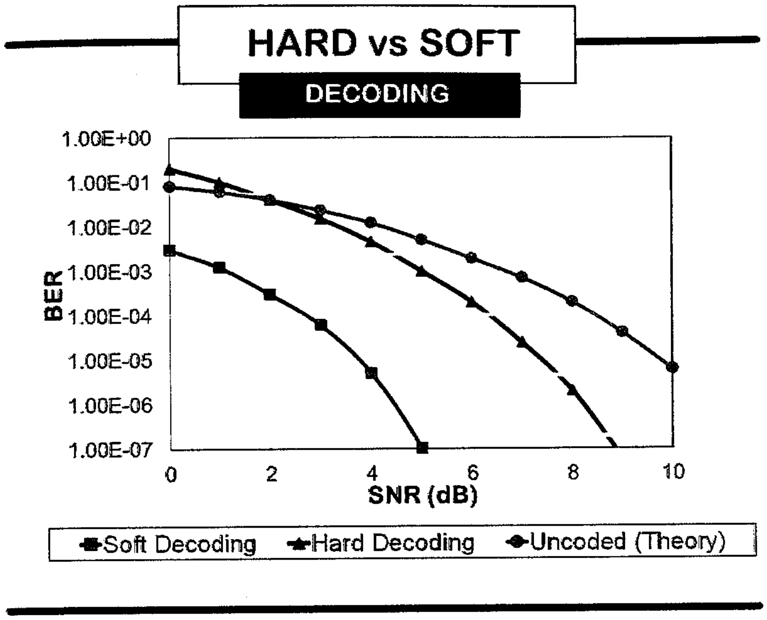

Figure 2.5: *BER performance comparison between hard and soft decoding BPSK with convolutional coding of* $\varphi = 1/2$

$$L_{i,b} \triangleq \frac{\ln P(s_{i,b} = +1|\mathbf{r},\mathbf{H})}{\ln P(s_{i,b} = -1)|\mathbf{r},\mathbf{H})} \tag{2.25}$$

for all bits b on level $i = 1, ..., M$. The sign of the LLR value $L_{i,b}$ shows whether bit $s_{i,b}$ is more likely to be $+1$ or 1 and the magnitude of $|L_{i,b}|$ indicates the probability of the estimate. The channel decoder takes advantage of the APPs and improves the estimate on the transmitted bits.

2.2.4.1 Soft-Output ML Detector

In [19], Equation (2.25) can be computed according to:

$$L(i,b) = \frac{\ln \sum_{\mathbf{s}\in\mathcal{Z}_{i,b}^{(+1)}} P_{\mathbf{r}}(\mathbf{r}|\mathbf{s},\mathbf{H})}{\ln \sum_{\mathbf{s}\in\mathcal{Z}_{i,b}^{(-1)}} P_{\mathbf{r}}(\mathbf{r}|\mathbf{s},\mathbf{H})} \tag{2.26}$$

under the assumption of equally distributed transmit symbols s. The sets $\mathcal{Z}_{i,b}^{(+1)}$ and $\mathcal{Z}_{i,b}^{(-1)}$ are

References

[1] International Energy Agency, "More Data, Less Energy: Making Network Standby More Efficient in Billions of Connected Devices." `http://www.iea.org/publications/freepublications/publication/MoreData_LessEnergy.pdf`. Accessed January 14, 2015.

[2] F. Wanlass and C. Sah, "Nanowatt Logic using Field-Effect Metal-Oxide Semiconductor Triodes," Technical Report, Xilinx Inc., December 2001.

[3] G. J. Foschini and M. J. Gans, "On the Limits of Wireless Communications in a Fading Environment when using Multiple Antennas," in *IEEE Wireless Personal Communications*, vol. 6, pp. 311–335, March 1998.

[4] E. Telatar, "Capacity of Multi-Antenna Gaussian Channels," in *European Transactions on Telecommunications*, vol. 56, pp. 619–630, December 1999.

[5] A. J. Paulraj and T. Kailath, "Increasing Capacity in Wireless Broadcast Systems using Distributed Transmission/Directional Reception (DTDR)," patent, The Board Of Trustees Of The Leland Stanford Junior University, September 1994.

[6] A. Goldsmith, E. Biglieri, R. Calderbank, A. Constantinides, A. Paulraj, and H. V. Poor, *MIMO Wireless Communications.* Cambridge: Cambridge University Press, 2007.

[7] D. Garrett, L. Davis, S. ten Brink, B. Hochwald, and G. Knagge, "Silicon Complexity for Maximum Likelihood MIMO Detection using Spherical Decoder," in *IEEE Journal on Solid State Circuits*, vol. 39, pp. 1544–1552, September 2004.

[8] C. Studer, A. Burg, and H. Bölcskei, "Soft-Output Sphere Decoding: Algorithms and VLSI Implementations," in *IEEE Journal on Selected Area in Communications*, vol. 26, pp. 290–300, February 2008.

[9] D. Wübben, R. Bohnke, V. Kühn, and K. D. Kammeyer, "MMSE Extension of V-BLAST Based on Sorted QR Decomposition," in *IEEE Proceedings of Vehicular Technology Conference*, vol. 1, pp. 508–512, October 2003.

[10] A. J. Paulraj, R. Nabar, and D. Gore, *Introduction to Space-Time Wireless Communications.* Cambridge University Press, 2003.

[11] A. Burg, N. Felber, and W. Fichtner, "A 50 Mbps 44 Maximum Likelihood Decoder for Multiple-Input Multiple-Output Systems with QPSK Modulation," in *IEEE International Conference on Electronics, Circuits and Systems*, vol. 1, pp. 332–335, December 2003.

[12] C. P. Schnorr and M. Euchner, "Lattice Basis Reduction: Improved Practical Algorithms and Solving Subset Sum Problems," in *Math Programming*, vol. 1, pp. 181–191, July 1993.