THE UNIVERSITY of EDINBURGH

Edinburgh Research Explorer

Adaptive Switching Detection Algorithm for Iterative-MIMO systems to Enable Power Savings

Citation for published version:
Tadza, NZBM, Laurenson, D & Thompson, J 2014, 'Adaptive Switching Detection Algorithm for Iterative-MIMO systems to Enable Power Savings' Radio Science., 10.1002/2013RS005323

Digital Object Identifier (DOI):
10.1002/2013RS005323

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Author final version (often known as postprint)

Published In:
Radio Science

General rights
Copyright for the publications made accessible via the Edinburgh Research Explorer is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy
The University of Edinburgh has made every reasonable effort to ensure that Edinburgh Research Explorer content complies with UK legislation. If you believe that the public display of this file breaches copyright please contact openaccess@ed.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.

OPEN ACCESS

Download date: 20. Feb. 2015

Radio Science, Volume ???, Number , Pages 1–33,

Adaptive Switching Detection Algorithm for Iterative-MIMO systems to Enable Power Savings

N Tadza,[1] D Laurenson,[1] and J S Thompson[1]

Corresponding author: N Tadza, Institute for Digital Communications, School of Engineering, University of Edinburgh, EH9 3JL, Edinburgh, UK. (n.tadza@ed.ac.uk)

[1]Institute for Digital Communications,
School of Engineering, The University of
Edinburgh, EH9 3JL, Edinburgh, UK.

This paper attempts to tackle one of the challenges faced in soft input soft output Multiple Input Multiple Output (MIMO) detection systems, which is to achieve optimal error rate performance with minimal power consumption. This is realized by proposing a new algorithm design that comprises multiple thresholds within the detector that, in real time, specify the receiver behavior according to the current channel in both slow and fast fading conditions, giving it adaptivity. This adaptivity enables energy savings within the system since the receiver chooses whether to accept or to reject the transmission, according to the success rate of detecting thresholds. The thresholds are calculated using the mutual information of the instantaneous channel conditions between the transmitting and receiving antennas of iterative-MIMO systems. In addition, the power saving technique, Dynamic Voltage and Frequency Scaling, helps to reduce the circuit power demands of the adaptive algorithm. This adaptivity has the potential to save up to 30% of the total energy when it is implemented on Xilinx®Virtex-5 simulation hardware. Results indicate the benefits of having this ‘intelligence’ in the adaptive algorithm due to the promising performance-complexity trade-off parameters in both software and hardware co-design simulation.

1. Introduction

The ability to increase throughput without requiring more computational power has always been a topic of interest amongst the wireless communication research community. Multiple Input Multiple Output (MIMO) promises high throughput without additional transmit power [*Goldsmith et al.*, 2007], however, minimizing the receiver's power, which is often limited, is still under intensive study. Current base stations, proliferations of femtocells and/or wireless access points also need to exercise being 'green'. The energy source is often shared amongst millions of devices. There are substantial potential of power savings to be gained in these small mains powered devices. In this paper, a field programmable gate array (FPGA) is used as a platform to show the inner workings of the adaptive algorithm. It is chosen due to its robustness, its reprogrammable capabilities and its potential for further energy savings by parallelization. The results obtained can be translated onto any hardware platform such as an application-specific integrated circuit (ASIC), which is more common in mobile devices. Fundamentally, a soft-MIMO receiver is divided into two parts, the MIMO detector and the soft decoder working together to achieve the best performance. The received data is processed through the detector before being passed into the decoder. Most publications focus on saving power using the signal-to-noise ratio (SNR) [Wu, 2011], channel matrix condition number [Matthaiou et al., 2008] or reducing the number of turbo decoding iterations [Zhang et al., 2009] for the receiver. Condition numbers of the channel matrix would only take into account the input and output matrix of the transmitter and the receiver. This is not sufficient as a switching metric since it disregards the noise level. SNR on the other hand, does not compute the relationship between the antennas in a MIMO system. If the channel is deemed good, due to high SNR values; strongly correlated antennas would not make for a good transmission condition. This is because the correlated system provides insufficient diversity in the MIMO system. Therefore, mutual information (MI) is implemented due to its consideration of the diversity of a MIMO system i.e. the transmitters and the receivers as well as the noise level. This gives a maximum amount of information regarding a channel with minimal complexity in comparison to using either condition number or SNR alone. This paper therefore, shifts the attention to the detector using MI as the threshold control; in hope to gain energy savings earlier on the processing stages i.e. by avoiding both detection and decoding processing. This iterative-MIMO scheme, which combines a spatial multiplexing MIMO detector and an outer forward error correction soft decoder with an interleaver in-between [*Ariyavisitakul*, 2000] [*Sellathurai and Haykin*, 2002], dubbed Bit-Interleaved Coded Modulation (BICM), [*Hochwald and ten Brink*, 2003], has very high computational complexity as the receiver detects and decodes symbols by searching through possible transmit symbols. Moreover, this is done iteratively in soft-MIMO detection systems by the decoder.

This paper focuses on saving energy consumption in the MIMO detector, where it predicts symbols transmitted by each antenna by examining the channel noise and constellation modulation scheme. It should be noted that, though out of scope of the paper, after the process of detecting, the symbols are passed to the outer decoder before a hard decision can be made.

There are many types of different detection algorithms available, which can be generalized into "Nulling and Cancelling" methods, such as the Zero Forcing (ZF) [*Winters et al.*, 1994] and the Minimum Mean Square Error Estimation (MMSE) [*Li et al.*, 2006] techniques as well as the "Tree Search" algorithms, for instance, the Maximum Likelihood (ML), Sphere Decoding (SD) [*Fincke and Pohst*, 1985], and the Fixed Sphere Decoding (FSD) [*Barbero and Thompson*, 2008] routines. For simple detectors, ZF and MMSE provide low complexity, however, they give poor performance in terms of bit-error-rate (BER). Linear detection methods, combined with nulling and cancelling, seem to give a better BER whilst maintaining the low complexity. This is why the combination of V-BLAST and ZF is chosen. On the other hand, for close to ML performance, tree search algorithms such as FSD, layered orthogonal lattice detector (LORD), smart ordered candidate adding algorithm (SOCA) and K-Best result in high complexity in order to meet the performance criteria. This drains a lot of power in order to decode data packets, which is particularly wasteful in good channel conditions. In poor channel conditions, FSD has been chosen as a detection method as it is independent of the search radius, meaning, the complexity is fixed and minimal in comparison to other tree search algorithms. The novelty of this paper lies in the fact that the algorithm switches between high and low complexity detectors to give a bigger gain in energy savings. Ultimately, using different detectors would only slightly alter the thresholds that need to be implemented, confirming that MI is adaptive to any system for determining the threshold for switching.

The computational power required to implement tree search MIMO detection every time a symbol is transmitted is unnecessary in some channel conditions. As each detection algorithm has a different performance and complexity, choosing between them depends on the system's unique requirements. To construct an adaptive implementation that could fit on available hardware in the market, this study combines two detection algorithms. The Fixed Sphere Decoding (FSD) and the Vertical Bell Laboratories Layered Space Time/Zero Forcing (V-BLAST/ZF) techniques are incorporated into an adaptive approach that has the ability to selectively operate according to the received signal conditions. These two detection algorithms are chosen due to their fixed data throughput, potential for hardware parallel implementation and low complexity.

The proposed adaptive algorithm therefore prevents the receiver from performing extensive computation under very low or very high SNR conditions, which ultimately yields significant savings in energy. The algorithm utilizes multiple thresholds to intelligently switch MIMO detection schemes according to the current environment. This 'intelligence' is the key to efficient energy utilization in the receiver. The results of this work will be presented in terms of overall energy savings from both software and hardware standpoints.

1.1. Contributions

The main contributions of this paper are summarized as follows:

- An adaptive switching algorithm that adapts to real time channel conditions by selecting to minimize the power 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 BER performance.

- The adaptive algorithm shows promising BER performance on a par with the current available detection schemes with lower computational complexity.

- An evaluation of the new design in a Xilinx®Virtex-5 FPGA shows convincing dynamic and static power savings compared to baseline detectors.

2. Background

2.1. System Model

Consider an iterative-MIMO system comprising M transmit antennas and N receive antennas based on BCIM, transmitting frames of K_u bits as shown in Figure 1. At the transmitter, the K_u bits are encoded using an iterative encoding method such as convolutional or turbo coding [*Hagenauer et al.*, 1996] of rate R_c, where $K_u = K_e \cdot R_c$. The K_e coded bits are then interleaved giving K_a bits, which are mapped into independent Quadrature Amplitude Modulation (QAM) constellations, $\mathcal{O}$, of P points, forming a sequence of $K_s = K_e / \log_2 P$ symbols. The symbols are separated into M substreams blocks of $M \cdot K_{ch}$ symbols are transmitted in each channel realization, K_{ch}. These are transmitted over Rayleigh fading channels. In other words, a frame of K_e coded bits requires a transmission of $K_s/(M \cdot K_{ch})$ blocks of data. Consequently, the received symbols, indexed by a sample time, k, can be written as

$$\mathbf{r}[k] = \mathbf{H}[k]\mathbf{s}[k] + \mathbf{n}[k] \tag{1}$$

where the channel matrix $\mathbf{H} \in \mathbb{C}^{M\times N}$ is assumed to be perfectly known at the receiver with independent elements $h_{i,j} \sim \mathcal{CN}(0,1)$, for $1 \leq i \leq M$ and $1 \leq j \leq N$ representing a block Rayleigh fading propagation environment, $\mathbf{s} = (s_1, s_2, \ldots, s_M)^{\mathrm{T}}$ is the transpose vector of the M-dimensional transmit symbol vector with $E[\mid \mathbf{s_i} \mid^2] = M^{-1}$, $\mathbf{n}$ is the $\mathbb{C}^{M\times 1}$ additive independent and identically distributed (i.i.d.) circular symmetric complex Gaussian noise vector of $h_{i,j} \sim \mathcal{CN}(0, \sigma^2)$ with $\sigma^2 = N_0$ and $\mathbf{r} = (r_1, r_2, \ldots, r_N)^{\mathrm{T}}$ is the transpose N-vector of received symbols. The set of all transmitted symbols form an M-dimensional complex constellation $\mathcal{O}^M$ of P^M vectors, which specifies the dimensionality of the system.

2.2. MIMO Detection

The channel $\mathbf{H}$ is assumed to be known at the receiver through a preceding training period. This generates and saves data in the channel estimation block regarding the modulation schemes and the channel condition statistics. MIMO algorithms solve (1) by separating parallel data streams transmitted by antennas. They can generally be categorized into four types as described below.

2.2.1. Maximum Likelihood (ML)

ML detection finds the minimum constellation point in (1) within the received symbols. It is given by

$$\hat{\mathbf{s}}_{ML} = \underset{\mathbf{s}\in\mathcal{O}^M}{\arg\min} \parallel \mathbf{r} - \mathbf{Hs} \parallel^2 \tag{2}$$

The ML detector is optimal and fully exploits all available diversity. 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 R_c, since the detector needs to go through 2^{R_c} 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 $K_s = 16^4 = 65536$ candidates in order to find the correct transmitted vector. Several efficient suboptimal detection techniques have therefore been proposed or adapted from the field of multiuser 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 diversity, 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.

2.2.2. Zero Forcing (ZF) - Linear Detection

This method neglects the constraint $\mathbf{s} \in \mathcal{O}^M$ in (2) and uses different criteria to find the nulling vectors, the most common being the ZF or MMSE approach [*Golub and Loan*, 1983]. Generally, the symbol $\hat{\mathbf{s}}$ is given by a transformation of the received vectors $\mathbf{r}$ in the form of

$$\hat{\mathbf{s}} = Q(\mathbf{H}^{+}\mathbf{r}) \tag{3}$$

where $\mathbf{H}^{+}$ is the Moore-Penrose pseudoinverse matrix that depends on channel $\mathbf{H}$ and Q is a quantizer that maps the argument into the closest point in $\mathcal{O}^M$. Even though this method has low complexity, it does have a major drawback of having a rather poor performance in terms of BER.

2.2.3. V-BLAST - Ordered Successive Interference Cancellation

V-BLAST [*Golden et al.*, 1999] method, gives slightly better BER performance in comparison to linear detection. However, due to the error propagation, it is still suboptimal in performance. This is often overlooked due to its practicality during implementation. V-BLAST is a recursive procedure that works by minimizing the influence of noise by re-ordering the channel matrix according to the signal strength received. The algorithm simply makes a first detection of the most powerful signal, consequently subtracting that signal from the overall detected symbols. It then continues the same process by proceeding to the detection of the second most powerful signal, and so forth. Assuming the ordered set to be

$$\S \equiv \{k_1, k_2, \ldots, k_M\} \tag{4}$$

the detection algorithm operates on $\mathbf{r}_i$, given in (5), while computing the decision statistics $y_{k_1}, y_{k_2}, \ldots y_{k_M}$, which are then quantized to form estimates of the received symbols $\hat{\mathbf{s}}_{k_1}, \hat{\mathbf{s}}_{k_2}, \ldots \hat{\mathbf{s}}_{k_M}$. The detection order is determined by the information about the channel conditions readily available within the estimation block. After computing (3), the detection process uses linear combinatorial nulling and symbol cancellation to successively compute the received vectors.

$$\mathbf{r}_{i+1} = \mathbf{r}_i - \hat{\mathbf{s}}_{k_i}(\mathbf{H})_{k_i} \tag{5}$$

When combined with the ZF method, it shows some improvement in BER while still maintaining low complexity. The complete V-BLAST/ZF detection algorithm is summarized in Table 1, where $\mathbf{G}$ denotes the Moore-Penrose pseudoinverse of the current channel $\mathbf{H}$, and therefore, $(\mathbf{G}_i)_j$ is the j^{th} row of $\mathbf{G}_i$, $Q(\cdot)$ is a quantizer to the nearest constellation point, $(\mathbf{H})_{\bar{k}_i}$ is the k_i^{th} column of $\mathbf{H}$, $\mathbf{H}_{\bar{k}_i}$ denotes the matrix obtained by zeroing the columns $k_1, k_2, \ldots, k_i$ of $\mathbf{H}$, and $\mathbf{H}^+_{\bar{k}_i}$ denotes the pseudoinverse of $\mathbf{H}_{\bar{k}_i}$. This type of detection scheme is best deployed in high SNR environments.

2.2.4. Sphere Decoding (SD) and Fixed Sphere Decoding (FSD)

SD reduces the complexity of the ML detection problem [*Viterbo and Boutros*, 1999] [*Pohst*, 1981] [*Agrell et al.*, 2006] by introducing a constraint within the search called the sphere radius, R.

$$\hat{\mathbf{s}}_{SD} = \underset{\mathbf{s} \in \mathcal{O}^M}{\arg\min} \parallel \mathbf{r} - \mathbf{H}\mathbf{s} \parallel^2 \leq R \tag{6}$$

The search can be visualized as a tree, traversing down each node until it encounters one with Euclidean Distance (ED) that is larger than R, where it will eliminate that branch from the search. The minimum symbol is acquired once it has traversed down through every path reaching the end i.e. the leaf node(s). The SD has major drawbacks when it comes to hardware implementation due to having variable complexity and its sequential nature. The complexity of the SD depends on the noise level and the channel conditions. Moreover, the linearity of the search prevents parallelism for newer hardware design implementation. Parallelization has been proven to minimize energy consumption in circuit designs due to a workload being shared across multiple computational resources, so that the circuit can produce the same amount of throughput at a lower frequency of operation. [*Chen et al.*, 2010] [*Esmaeilzadeh et al.*, 2011] [*Kumar et al.*, 2003]. Therefore, [*Barbero and Thompson*, 2008] proposed a modified version, the FSD, in order to overcome both shortcomings. FSD is a combination of brute-force enumeration and a low complexity, approximate detector. Much like the SD, FSD traverses down the tree whilst calculating the ED; however, instead of having a radius constraint R, FSD determines in advance the number of lattice points $\hat{\mathbf{s}}$ around received signal $\mathbf{r}$ it would pass through, evaluating $\mathbf{r}$ independent of the noise level, giving it a fixed throughput. The algorithm makes use of the fact that [*Barbero and Thompson*, 2008] the diagonal entries of $\mathbf{R}$ from the $\mathbf{QR}$-decomposition of the channel matrix satisfy

$$E[\mathbf{r}_{11}^2] < E[\mathbf{r}_{22}^2] < \cdots < E[\mathbf{r}_{NN}^2] \tag{7}$$

Thus, the number of candidates at antenna level k denoted by n_k should follow

$$E[n_N] \geq E[n_{N-1}] \geq \cdots \geq E[n_1] \tag{8}$$

The main idea of FSD is to assign a fixed but distinct number of candidates to be searched per antenna level. The FSD is considered a promising algorithm for soft-MIMO detection. Since its introduction, the reduction of complexity in FSD has received significant attention [*Barbero et al.*, 2008] [*Lei et al.*, 2010] [*Liu et al.*, 2011] [*Li et al.*, 2009] [*Wu and Thompson*, 2011].

2.3. Iterative Decoding

An iterative decoder [*Hagenauer et al.*, 1996] is used right after the MIMO symbols have been detected, where soft information extrinsic log-likelihood ratio (LLR) values are exchanged iteratively between the outer decoders with interleaving/de-interleaving operations in between until the desired performance is achieved [*Berrou et al.*, 1993]. The idea behind soft detection is to generate *a posteriori* probability

(APP) values in the form of LLR information, $L_E(b_k|\mathbf{r})$, about the interleaved bits, $\mathbf{b}$, for $1 \leq k \leq K_e$, while taking into account the channel observations $\mathbf{r}$ and the *a priori* LLR information, $L_A(b_k)$, coming from the outer decoder. For the system under consideration, assuming that the bits b_k are statistically independent due to the interleaving operation and making use of the Max-log approximation, $L_E(b_k|\mathbf{r})$ can be approximated by

$$\begin{aligned} L_E(b_k|\mathbf{r}) \approx \frac{1}{2} \max_{\mathbf{b} \in \mathcal{L} \cap \mathbb{B}_{k,+1}} & \left(\frac{- \parallel \mathbf{r} - \mathbf{Hs} \parallel^2}{\sigma^2/2} + \mathbf{b}_{[k]}^{\mathrm{T}} \mathbf{L}_{A[k]} \right) \\ - \frac{1}{2} \max_{\mathbf{b} \in \mathcal{L} \cap \mathbb{B}_{k,-1}} & \left(\frac{- \parallel \mathbf{r} - \mathbf{Hs} \parallel^2}{\sigma^2/2} + \mathbf{b}_{[k]}^{\mathrm{T}} \mathbf{L}_{A[k]} \right) \end{aligned} \tag{9}$$

for $1 \leq k \leq K_e$, where, without loss of generality, $K_e = M \cdot \log_2 P$ has been assumed to simplify the index notation. In (9), $\mathbf{b} = (b_1, b_2, b_3, \ldots, b_{K_e})^{\mathrm{T}}$, $\mathbf{b}_{[k]}$ denotes the subvector of $\mathbf{b}$ omitting b_k, $\mathbf{L}_A = [L_A(b_1), L_A(b_2), \ldots, L_A(b_{K_e})]^{\mathrm{T}}$, $\mathbf{L}_{A[k]}$ denotes the subvector of $\mathbf{L}_A$ omitting $L_A(b_k)$, $\mathbb{B}_{k,+1}$ and $\mathbb{B}_{k,-1}$ represent the sets of 2^{K_e-1} bit vectors $\mathbf{b}$ having $b_k = +1$ (logical 1) and $b_k = -1$ (logical 0) respectively, $\mathcal{L} \cap \mathbb{B}_{k,+1}$ and $\mathcal{L} \cap \mathbb{B}_{k,-1}$ denote the subgroups of vectors of $\mathcal{L}$ that have $b_k = +1$ and $b_k = -1$ respectively. The list of candidates $\mathcal{L} \subset \mathcal{O}^M$ is detector specific and subject to the overall performance and complexity of the iterative-MIMO receiver, since $\parallel \mathbf{r} - \mathbf{Hs} \parallel^2$ needs to be computed for all $\mathbf{s} \in \mathcal{L}$. Although iterative decoding does contribute to the overall complexity of a MIMO receiver, numerous studies have been done in reducing the total complexity of iterative decoding [*Li et al.*, 2013] [*Mathana et al.*, 2009] [*Wu*, 2011] [*Zhongfeng et al.*, 2009], therefore, this paper focuses on minimizing energy consumption in the MIMO detector. It should be noted that some of the complexity of iterative decoding will be avoided due to the proposed adaptive algorithm design; however, this is out of scope of this paper.

2.4. Power Savings

Energy consumption in mobile devices with battery powered sources is a major limiting factor in circuit designs. Fundamentally, energy is consumed in both dynamic and static aspects as specified by (10). Most publications like [*Mirsad et al.*, 2009], [*Andrei et al.*, 2009] and [*Salehi et al.*, 2009] have successfully reduce the dynamic power consumption, however, in newer chip technologies, the static power consumption is said to be high, [*Telikepalli et al.*, 2006], therefore, this work investigates ways to reduce both types, dynamic and static energy consumption, in a circuit design, while ensuring the algorithm performance is sufficient. This will ensure the adaptive algorithm is properly optimized to meet power budget of the design. In order to evaluate the overall power savings gained by the adaptive algorithm, both software and hardware savings should be analyzed.

$$E_{total} = E_{dynamic} + E_{static} + E_{I/O} + E_{transceiver} \tag{10}$$

There are multiple ways to exploit energy savings in circuit designs and different energy has different approach to execute these. For example, savings in $E_{dynamic}$ are achieved by deploying the Dynamic Voltage and Frequency Scaling (DVFS) technique [*Rabaey et al.*, 2009] while on the other hand, savings in E_{static} depend on the manufacturing process, the temperature, and the voltage, V .

2.4.1. Dynamic Energy

Dynamic energy, $E_{dynamic}$, spent within CMOS technology is due to toggling of transistors and is a function of clock frequency, f, which can be varied within some limit (before the circuit fails to function due to overheating), the value of V, and the capacitance. The $E_{dynamic}$ is given by the relation [*Abusaidi et al.*, 2008] below

$$E_{dynamic} = \frac{nCV^2 f}{t} \tag{11}$$

where n is the number of toggling transistors, C is the circuit capacitance, V is the voltage swing, f is the toggling frequency and t is the time it takes to complete an operation. DVFS has shown significant energy savings when applied to circuit designs, evident in [*Larson and Gustafsson*, 2011], [*ARM Industry*, 2009] and [*Kim et al.*, 2008]. Much like the adaptive algorithm, DVFS has the ability to adjust its parameters to match the computational demand of the current workload. If the workload requirement is high, DVFS will increase the V, to supply the circuit so that it can operate at a higher f in order to meet the desired data throughput within a particular time period. The opposite is also true; when the workload is minimal, the circuit could operate on a much lower f, which ultimately, according to (11) will decrease the overall $E_{dynamic}$ as the task time lengthens. This adaptivity is appealing to the design of the adaptive algorithm since now both hardware and software possess the same level of adaptivity and 'intelligence'. Both approaches will in turn yield significant overall energy savings.

2.4.2. Static Energy

Static energy, E_{static}, is consumed due to transistor leakage and is highly dependent on the manufacturing process, the ambient temperature of the circuit, and the value of V. According to the study by [*Telikepalli et al.*, 2006], E_{static} seems to dominate the overall power consumption within a circuit as the chip size shrinks. Therefore, E_{static} can no longer be neglected when designing new algorithms into new chip technology.

3. Adaptive Algorithm Methodology

Current MIMO detectors usually lack adaptivity whereby all receivers behave exactly the same way regardless the received signal characteristics. This 'one size fit all' architecture does not work well in some situations, since different users experience distinct channel conditions. For example, a stationary user who is physically near to a transmitter would often have a better data throughput than one who is further away. Doppler rates determined by motion in the environment also play a part in determining the current condition of the channel. To decode symbols in bad channel conditions would prove to be pointless since the data would not be likely to be decoded successfully anyway. Therefore, having 'intelligence' in the detector that could modify its behavior according to current channel conditions would be ideal. This adaptivity in the algorithm is controlled by the MI calculation between the transmitters and receivers. It is well-known that MI of a MIMO channel is given by (12) and the information required, $\mathbf{H}$ is already available within the channel estimation block. Different values of initial received soft information may lead to significantly different behavior during the iterative decoding process. The study done by [*Zhang et al.*, 2009], which compares the performance of iterative decoders using different received soft LLR information metrics, discovered that by computing the MI, the number of iterations in turbo decoding can be found using the highest complexity ML MIMO detection method. [*Zhang et al.*, 2009] also proves that the best approximation of the received symbols obtained are lossless and that the exact LLR values are sufficient enough statistic of **r** about **s**. Therefore, using this information and the principle of exploiting MI calculation in (12), the paper applies this approach for the first time to a MIMO detector to further save energy consumption in the overall receiver. With any given channel model in (1), and a Gaussian constellation with $E[|\ \mathbf{s_i}\ |^2] = M^{-1}$, the MI for the ML method is

$$\bar{I}(\mathbf{H}_k) \triangleq \log_2 \det(I + \frac{\mathbf{H}_k \mathbf{H}_k^T}{N_0}) \qquad (12)$$

The values of MI spread at specific SNR conditions. Figure 2 illustrates the accumulated MI performance of the detector as a function of probability of receiver fails and successes. The system is simulated using a 4×4 MIMO system with 16-QAM modulation symbols transmitting 1024 bits per packet of 10000 channel realizations utilizing an iterative-MIMO decoder of $1/2$ code rate in a fast fading environment. Threshold 1 can be obtained in Figure 2(a), which shows the FSD performance. Below a certain MI threshold of approximately 2200, the receiver is certain to fail when trying to decode a symbol message. Therefore, the best cause of action for the receiver is to request a retransmission i.e. Automatic Repeat Request (ARQ), from the transmitter rather than to attempt decoding where it is unlikely to succeed, wasting significant computational energy, which is the limitation of today's system designs. On the other hand, the V-BLAST/ZF performance is shown in Figure 2(b), where a value of about 7100 for threshold 2 can be seen. The receiver will decode the symbol message with very high probability above this MI value, therefore, a simpler detection method will suffice in detecting the symbol, i.e. the V-BLAST/ZF method. In addition, the area in-between the two thresholds shows that the receiver would sometimes fail to decode. Thus, a more powerful detection method is needed to assist the receiver in decoding the message. This is done by deploying the FSD algorithm in the MIMO detector. By obtaining these thresholds, the design of the adaptive algorithm can be described in Table 2. It should be noted that, the thresholds obtained are catered specifically for 16-QAM modulation scheme on a 4×4 MIMO system, however, the idea behind adaptive algorithm can be adjusted to fit any communication systems. The same analysis can be applied to all other modulation and coding schemes, with the exception of having different threshold values when calculated using (12).

4. Results and Analysis

The effectiveness of the adaptive algorithm can be measured using the performance and complexity trade-off metrics. This section describes these efficiencies from both hardware and software perspectives.

4.1. SOFTWARE - Performance

The performance can be quantified by calculating the number of errors in a total frame i.e. the BER analysis. The system design has been set to tolerate a BER of 10^{-3} or less in high SNR regions. In the system model used, the BER is depicted in Figure 3. The adaptive algorithm gives similar performance to the FSD and performs much better than the V-BLAST/ZF algorithm in low SNR regions. In very high SNRs, i.e. 10 dB and above, the less complex algorithm of V-BLAST/ZF is adopted and the BER performance is below the set error tolerance line. The FSD does give a much better performance than the tolerance line, however, this level of performance is unnecessary and only adds extra complexity for the hardware. When the SNR is below 0 dB, the receiver abandons the detection process (subsequently avoiding the complexity of the iterative decoding process as well, gaining substantial power savings) and requests a retransmission from the transmitter, whereas the area above the set threshold, circa 0 dB to 6 dB, the adaptive algorithm provides much higher chances of successful processing in comparison to the V-BLAST/ZF method.

4.2. SOFTWARE - Complexity

By obtaining the thresholds, the total number of usage of each MIMO detection algorithm throughout the span of the SNR is shown in Figure 4, depicting transmissions of 10000 packets of 1024 bits per frame. It clearly shows that below an SNR value of 0 dB i.e. threshold 1, no processing is taking place. In addition, in high SNR regions, V-BLAST/ZF is utilized. This figure concurs with Figure 3, where the performance coincides with the algorithm switching rate of sucessfulness. From this, another part of the parameter, the complexity measurement of the software can be determined.

Complexity measurement gives an important overview of the hardware before implementation and provides an initial indication of power savings in the design. A preliminary complexity analysis of the adaptive algorithm is determined by the multiplier counts in the code. Assuming that the complexity of channel ordering is the same for both detection schemes, the multiplier counts between the FSD and V-BLAST/ZF detection schemes for a transmission of one symbol for 4×4 M-QAM deploying FSD is M-times more complex than the V-BLAST/ZF. Figure 5 plots the percentage complexity results against the SNR of the channels, where 100% equals the complexity of FSD, while the V-BLAST/ZF requires only 25%. The complexity of the adaptive algorithm can be calculated by averaging over MI values shown at certain SNR in the figure and it is much lower than the FSD, i.e. 62% of the multipliers required. Most energy savings can be gained during the 'No Decoding' phase since no processing is required in this region. Furthermore, energy are saved during the utilization of V-BLAST/ZF algorithm i.e. where MI > 7100, which gives a total of only 25% multiplier usage.

4.3. HARDWARE - Performance and Complexity

Xilinx®Virtex-5 has a varying voltage range of 0.95 V to 1.05 V, and an operational frequency range of 60 MHz to 400 MHz [*Klein*, 2009]. In order to assess the efficacy of the DVFS technique in saving energy consumption in wireless communication, both MIMO detection algorithms - FSD and V-BLAST/ZF, are operated at low power mode (0.95 V, 60 MHz) and high performance mode (1.05 V, 400 MHz) to get the minimum and maximum thresholds of operation. This information is determined using the Xilinx®Design Suite software for the Xilinx®Virtex-5. The Xilinx®Design Suite software comprises a co-design software/hardware setup performed in Matlab™ and Xilinx®System Generator, which is a part of the Xilinx®ISE. In addition, the power profile is analyzed using the Xilinx®Power Estimator (XPE) tool. The summary of the total number of the FPGA resources used are given in Table 3. The percentage of slices used can be seen as an indicator of the amount of control logic and intermediate buffers required in the adaptive algorithm. This factor affects hardware mapping and the resulting throughput. The average throughput of the system is a parameter of importance when considering the performance of the algorithm. The throughput in megabits per second (Mbps) is calculated according to

$$Q_{avg} = M \cdot \log_2 P \cdot f / C_{avg} \tag{13}$$

where C_{avg} is the average number of clock cycles required to detect a MIMO symbol.

For low power mode, where $f = 60$ MHz and the minimum number of cycles is $C_{min} = 4$, the maximum throughput is $Q_{min} = 240$ Mbps while the high performance mode gives a throughput of $Q_{max} = 1200$ Mbps. Increasing the clock frequency would result in a significant increase in the throughput, therefore, the $f = C_{avg}$ could be seen as an indicator of the level of optimization of the hardware design.The hardware setup parameters are included in Table 4.

Similar to details reported in [*Mirsad et al.*, 2009] [*Andrei et al.*, 2009] [*Salehi et al.*, 2009] [*Larson and Gustafsson*, 2011], there are significant dynamic power savings in the circuit, portrayed in Figure 6, where low power mode uses 9% of the overall power in comparison to 29% when the circuit is run at full power i.e. the high performance mode. However, these savings would be minimal in comparison due to the much larger static power, which dominates the overall chip power. Figure 7 shows the low power results for FSD (a) and V-BLAST/ZF (c) as well as the high performance statistics, (b) and (d), for FSD and V-BLAST/ZF respectively. It is shown that some savings are gained when the adaptive algorithm switches from the high complexity FSD to the simpler V-BLAST/ZF detection. The power saved during the swap is equivalent to 20% for high performance and 8% for low power mode. The energy savings when changing from high performance to low power are also illustrated here. The total time computed is obtained by transmitting one packet of 1024 bit frame using a 16-QAM modulation symbol over the 4×4 MIMO channel when operating at the lowest frequency of 60 MHz. When operating at 400 MHz, the task completion time takes approximately 7 times less than when operating at lower frequency. By finishing quickly, the hardware can be put into sleep mode, reducing the total energy, since the idle power is negligible ≈ 0.08 mW. By calculation, at the same total rate of completion, the energy required to complete one task is lower by 42% when the circuit operates quickly and switches into idle state (high performance) than to run slowly and finishes just in time, at lower frequency (low power) when deploying FSD, and 52% for the V-BLAST/ZF algorithm. These are the savings which can be gained when putting the chip into sleep mode for more than 15 μs. Even though in theory, verified in (11), the longer the task runs, the lower the dynamic energy consumption, this is not the case here because when evaluating the total energy consumption of the circuit, the E_{static} required in powering up the Xilinx®Virtex-5 hardware is too large, occupying most of the power demand of the chip, resulting in 84% and 65% of the total power for low power and high performance mode respectively, as shown in Figure 6. These findings coincide with the work reported in [*Hasan and Bird*, 2011];

stating that, as manufacturing process get smaller, the E_{static} seems to dominate the overall chip power. Therefore, it can be concluded that running the circuit at a lower speed is not the answer to overall power savings in this technology. E_{static} could no longer be neglected when designing a circuit, and it is now more essential to take temperature as a parameter in saving overall energy consumption, since E_{static} strongly depends on the heat generated by the circuit.

Figure 8 shows the overview of the algorithm flow within the chip. Only one detector is switched on at any given time according to the calculation from the threshold control block. This is particularly useful for FPGA implementation since the hardware resources are switched on and off as required. The implementation of the adaptive algorithm is illustrated in terms of the FPGA hardware given in Figure 9. The configurable logic utilised for each detector is shown in (a) for FSD, (b) for V-BLAST/ZF and (c) when 'No Decoding' is taken place. It can be seen that only certain parts of the overall chip hardware are turned on at any given time. Seeing that most power consumption is due to powering the up the chip itself, i.e. static power, the adaptive algorithm takes advantage of this fact and therefore shuts down parts of the chip which are not in use. To show how the adaptive algorithm behaves, consider four extreme scenarios of three frames of 1024 data bits per frame size being transmitted over different environments, where T_1 is when the MI is at a high value, T_2 is for when MI is acceptable and T_3 is for MI being low and not suitable for further decoding. From Figure 5, it is shown that, the complexity of an FSD is four times larger than that of the V-BLAST/ZF. Therefore, if the complexity of the V-

BLAST/ZF is set to 1, the FSD will have the equivalent complexity of 4. The overall chip area usage is given in Figure 10. Using the same complexity ratio, consider a transmission of 100000 frames of 1024 bits per frame on random fast fading channel realisations over various ranges of SNR values from -4 dB to 20 dB. The adaptive algorithm saves approximately 30% of the overall resource in comparison to the FSD detector whilst maintaining the BER performance at a satisfactory region.

Shutting down parts of the chips i.e. sleep modes, are the key enablers in saving further energy in the design on Virtex-5 hardware. By running the circuit at high frequency, the sleep modes can help prevent the circuit from running and powering up the entire logic gates all the time, consequently preventing the circuitry from overheating that leads to high E_{static} consumption.

For greater insight of the total energy savings that can be achieved in a realistic setting, Figure 11 considers the adaptive algorithm in a Rayleigh fading channel. The SNR chosen are based on the operating SNR regions of the Long Term Evolution (LTE). In small cells, the transmit power is to be in the range of 23 dB to 46 dB, averaging at 26.5 dB [*Nakamura, et al.*, 2013]. The savings can found by integrating the power, P, with respect to the probability density function, f, of the fading environment, ρ, as shown in (14).

$$\int_a^b P(\rho) f(\rho) \, \mathrm{d}\rho \tag{14}$$

where a is the lower SNR value of -4 dB and b is the upper limit of the SNR, which is 40 dB in this case. Using a discrete approximation to this gives a measure of the savings that can be achieved in practice. For example, taking the FSD as a benchmark would use 8 J (in high performance) of energy to decode the 1024 bits data packet size. Utilizing the adaptive algorithm would use 70% less resources since the FSD does not take into account the transmit power nor the SNR values, which results in unnecessary power wastage. In addition, the behavior of the adaptive algorithm follows that of the Rayleigh fading channel for a 4×4 MIMO system, operating on 74% of the fading channel environment, gaining energy savings due to sleep implemented in the appropriate regions i.e. FSD is on sleep mode at SNR of 20 dB, and only V-BLAST is active.

The energy saving results obtained can be optimized further by combining the common circuitry of the FSD and V-BLAST since they share some common functionality. By sharing the circuitry resources between the two algorithms can gain additional energy savings. Detailed evaluation of the issues is the next major step of the project.

5. Future Direction

Research is still ongoing in the field of both hardware and software design. This section describes some of the planned future work.

5.1. SOFTWARE - Algorithm Switching Selection

The SNR values at which the adaptive algorithm switches between the different thresholds is illustrated in Figure 4. The selection of adaptive algorithm can be optimized. At a particular SNR, the MI varies, and must be calculated by the receiver. The effect is that the detector switches between approaches in regions corresponding to the MI thresholds. The transitions across the MI thresholds result in switching from one to the other rapidly. This switching could have an impact on the power consumption. One possible improvement is to enforce use of FSD during these situations when V-BLAST/ZF fails to decode a packet, or when there would be rapid switching between FSD and 'No Decoding'. However, even though this would increase the likelihood of decoding, it would be at a cost of higher energy consumption.

5.2. HARDWARE - New Xilinx®Virtex 7

Newer technology chips such as the Xilinx®Virtex-7, based on a different manufacturing process, have an improved solution to the high E_{static} consumption of previous circuit technologies [*Hussein et al.*, 2013]. It may therefore be that DVFS can be applied to minimize power consumption in this type of hardware, due to E_{static} no longer dominating the total chip power.

6. Conclusion

Having 'intelligence' in the algorithm design and the hardware offers both adequate performance and reduced complexity in future iterative-MIMO systems. The adaptive algorithm within the MIMO receiver demonstrates significant energy savings in both software and hardware implementation. It has the potential to save up to 30% energy in the software design and in the Xilinx®Virtex-5 hardware. This can be improved further when incorporating sleep modes to reduce the E_{static} in the hardware apparatus.

Acknowledgments. This work is funded by the University of Tun Hussein Onn Malaysia as a part of the main author's PhD program.

References

Abusaidi, B. P., M. Klein, and B. Philofsky, Virtex-5 FPGA System Power Design Considerations (2008), *White Paper*, vol. 285, pp. 1-23.

Agrell, E., T. Eriksson, A. Vardy, and K. Zeger, Closest Point Search in Lattices (2002), *IEEE Transactions on Information Theory*, vol. 48, no. 8, pp. 2201-2214.

Andrei, A., P. Eles, Z. Peng, S. Link, M. Schmitz, and B. M. Al-Hashimi, Energy Optimization of Multiprocessor Systems on Chip by Voltage Selection (2009), *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 15, no. 3, pp. 262-275.

Ariyavisitakul, S.L, Turbo Space-Time Processing to Improve Wireless Channel Capacity (2000), *IEEE Transactions on Communications*, vol. 48, no. 8, pp. 1347–1359.

ARM Industry, The ARM Cortex-A9 Processors (2009), *White Paper*, pp. 1-11.

Barbero, L.G., T. Ratnarajah, and C. Cowan, A low-complexity soft-MIMO detector based on the fixed-complexity sphere decoder (2008), *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 2669-2672.

Barbero, L.G., J. S. Thompson, Extending a Fixed-Complexity Sphere Decoder to Obtain Likelihood Information for Turbo-MIMO Systems (2008), *IEEE Transactions on Vehicular Technology*, vol. 57, no. 5, pp. 2804-2810

Barbero, L.G., J. S. Thompson, Fixing the Complexity of the Sphere Decoder for MIMO Detection (2008), *IEEE Transactions on Wireless Communications*, vol. 7, no. 6, pp. 2131-2142.

Berrou, C., A. Glavieux, and P. Thitimajshima, Near Shannon Limit Error - Correcting Coding and Decoding: Turbo Codes (1993), *IEEE International Conference on Communications*, no. 1, pp. 1064-1070.

Chen, Y.K, C. Chakrabarti, and B. Bougard, Signal Processing on Platforms with Multiple Cores: Part 2 – Applications and Design (2010), *IEEE Signal Processing Magazine*, vol. 2, no. 1, pp. 20-21.

Esmaeilzadeh, H., E. Blem, R. St. Amant, K. Sankaralingam, and D. Burger, Dark Silicon and the end of Multicore Scaling (2011), *Proceeding of the 38th Annual International Symposium on Computer Architecture*, pp. 365.

Fincke, B. U., M. Pohst, Improved Methods for Calculating Vectors of Short Length in a Lattice, Including a Complexity Analysis (1985), *Mathematics of Computations*, vol. 44, no. 170, pp. 463-471.

Golden, G.D, C. J. Foschini, R. A. Valenzuela, and P. W. Wolniansky, Detection Algorithm and Initial Laboratory results using V-BLAST space time Communication Architecture (1999), *IEEE Electronic Letters*, vol. 35, no. 1, pp. 14-16

Goldsmith, A., E. Biglieri, R. Calderbank, A. Constantinides, A. Paulraj, and H. V. Poor (2007), *MIMO Wireless Communications*, Cambridge, Cambridge University Press, pp. 1–559.

Golub, G. H., C. F. Van Loan, *Matrix Computations* (1983), Third. Baltimore and London: The Johns Hopkins University Press, pp. 292-300.

Hagenauer, J, E. Offer, and L. Papke, Iterative decoding of binary block and convolutional codes (1996), *IEEE Transactions on Information Theory*, vol. 42, no. 2, pp. 429-445.

Hasan, M. Z., M. Bird, Energy Reductions for Embedded Processors in Reconfigurable Hardware (2011), *IEEE International Conference on Electro/Information Techonology*, pp. 1-8.

Hochwald, B. M., and S. T. Brink (2003), Achieving Near-Capacity on a Multiple-Antenna Channel *IEEE Transactions on Communications*, vol. 51, no. 3, pp. 389–399.

Hussein, B. J., M. Klein, and M. Hart, Lowering Power at 28 nm with Xilinx®7 Series Devices (2013), vol. 389, pp. 1-25

Jaldén, J., L. G. Barbero, B. Ottersten, and J. S. Thompson, The Error Probability of the Fixed-Complexity Sphere Decoder (2009), *IEEE Transaction in Signal Processing*, vol. 57, no. 7, pp. 2711-2720.

Kim, W., M. S. Gupta, G. Wei, and D. Brooks, System Level Analysis of Fast, Per-Core DVFS using On-Chip Switching Regulators (2008), *High Performance Computer Architecture, IEEE 14th International Symposium*, no. 1620, pp. 123-

134.

Klein, M., Power Consumption at 40 and 45 nm (2009), *White Paper*, vol. 298, pp. 1-21.

Kumar, R., K. I. Farkas, N. P. Jouppi, P. Ranganathan, and D. M. Tullsen, Single-ISA Heterogeneous Multi-Core Architectures: The Potential for Processor Power Reduction (2003), *Proceedings of the 36th International Symposium on Microarchitecture*.

Larson, E. G., O. Gustafsson, The Impact of Dynamic Voltage and Frequency Scaling on Multicore DSP Algorithm Design The Impact of Dynamic Voltage and Frequency Scaling on Multicore DSP Algorithm Design (2011), *IEEE Signal Processing Magazine*, no. 28.

Lei, S., Q. Tu, D. Yang, and J. Chen, Probabilistic tree pruning for fixed-complexity sphere decoder in MIMO systems (2010), *International Conference on Wireless Communications and Signal Processing (WCSP)*, pp. 1-6.

Li, G., X. Zhang, S. Lei, C. Xiong, and D. Yang, An Early Termination-based Improved Algorithm for Fixed-complexity Sphere Decoder (2012), *IEEE Wireless Communications and Networking Conference: PHY and Fundamentals*, no. 1, pp. 629-634.

Li, P., D. Paul, R. Narasimhan, and J. Cioffi (2006), On the Distribution of SINR for the MMSE MIMO Receiver and Performance Analysis, *IEEE Transactions on Information Theory*, vol. 52, no. 1, pp. 271-286.

Li, Liang, Maunder, Robert G., Al-Hashimi, Bashir, Hanzo, Lajos, A Low-Complexity Turbo Decoder Architecture for Energy-Efficient Wireless Sensor Networks (2013), *IEEE Transactions on Very Large Scale Integration*, vol. 21, no. 1, 14-22.

Liu, L., J. Lofgren, and P. Nilsson, Low Complexity Soft-output Signal Detector for Spatial-multiplexing MIMO system (2011), *IEEE International Wireless Communications and Mobile Computing Conference*, pp. 988–993.

Matthaiou, M., Laurenson, D.I., Wang, C.X, Reduced Complexity Detection for Ricean MIMO Channels Based on Channel Number Thresholding (2008), *IEEE 22nd International Symposium on Personal, Indoor and Mobile Radio Communications*, pp. 1718–1722.

Mathana, J. M., P. Rangarajan, J. Raja Paul Perinbam, Low Complexity Reconfigurable Turbo Decoder for Wireless Communication Systems (2013), *Arabian Journal for Science and Engineering*, vol. 38, no. 10, pp. 2649–2662.

Mirsad, C., D. Persson, and E. G. Larson, Allocation of Computational Resources for Soft MIMO Detection (2011), vol. 5, no. 8, pp. 1451-1461.

Nakamura, T., Trends in Small Cell Enhancements in LTE Advanced (2013), *IEEE Communication Magazine*, vol. 51, no. 2, pp. 98–105.

Pohst, M., On the Computation of Lattice Vectors of Minimal Length, Successive Minima and Reduced Bases with Applications (1981), *Newsletter ACM SIGSAM Bulletin*, vol. 15, no. 1, pp. 37-44.

Rabaey, J., *Low Power Design Essentials* (2009), Department. New York: Springer 2009, pp. 289-316.

Salehi, M. E., M. Samadi, M. Najibi, A. Afzali-Kusha, M. Pedram, and S. M. Fakhraie, Dynamic Voltage and Frequency Scheduling for Embedded Processors Considering Power/Performance Tradeoffs (2011), *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 19, no. 10, pp. 1931-1935.

Sellathurai, M., S. Haykin, TURBO-BLAST for Wireless Communications: Theory and Experiments (2002), *IEEE Transactions on Signal Processing*, vol. 50, no. 10, pp. 2538–2546.

Telikepalli, A., Power vs . Performance: The 90 nm Inflection Point Reducing Power in FPGAs The Triple Challenge (2006), *White Paper*, vol. 223, pp. 1-18

Viterbo, E., J. Boutros, A Universal Lattice Code Decoder for Fading Channels (1999), *IEEE Transactions on Information Theory*, vol. 45, no. 5, pp. 1639-1642.

Winters, J. H., S. Member, J. Salz, R. D. Gitlin, The Impact of Antenna Diversity on the Capacity of Wireless Communication Systems (1994), *IEEE Transactions on Communications*, vol. 42, no. 2, pp. 1740-1751.

Wu, P. H. Y., On the complexity of Turbo Decoding Algorithms (2001), *IEEE Vehicular Technology Conference*, vol. 53, no. 2

Wu, X., J. S. Thompson, A Fixed-Complexity Soft-MIMO Detector via Parallel Candidate Adding Scheme and its FPGA Implementation (2011), *IEEE Communications Letters*, vol. 15, no. 2, pp. 241-243.

Zhang, J., M. A. Armand, P. Y. Kam, and A. T. Mi, A Mutual Information Approach for Comparing LLR Metrics for Iterative Decoders (2009), *IEEE Communication Society*, vol. 1, no. 4, pp. 1-4.

Zhang, J., M. A. Armand, P. Y. Kam, A. T. Mi, Low hardware complexity parallel turbo decoder architecture (2009), *International Symposium on Proceedings of the Circuits and Systems*, vol. 1, no. 4, pp. 1-4.

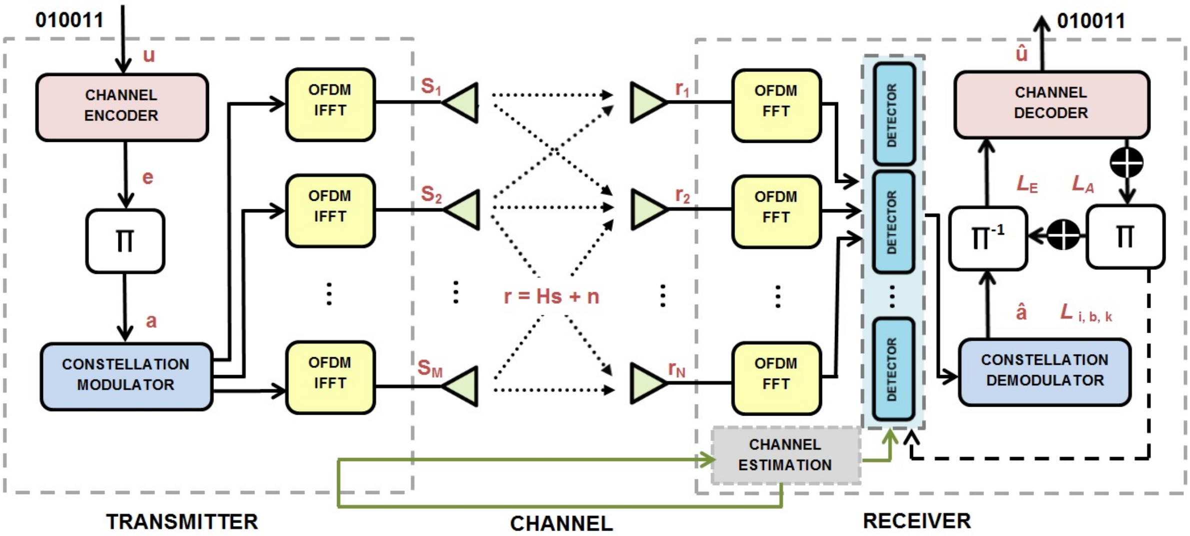

Figure 1. Iterative-MIMO (BICM) System

Table 1. V-BLAST/ZF Algorithm

Pseudo-Code
Channel realization:
$\mathbf{G}_1 = \mathbf{H}^{+}$
$i = 1$
Recursion:
$k_i = \arg\min_{j \notin \{k_1, \cdots, k_{i_1}\}} \parallel (\mathbf{G}_i)_j \parallel^2$
$y_{k_i} = (\mathbf{G}_i)_{k_i} \mathbf{r}_i$
$\hat{\mathbf{s}}_{k_i} = Q(y_{k_i})$
$\mathbf{r}_{i+1} = \mathbf{r}_i - \hat{\mathbf{s}}_{k_i}(\mathbf{H}_{k_i})$
$\mathbf{G}_{i+1} = \mathbf{H}^{+}_{\bar{k}_i}$
$i = i + 1$

[1] Algorithm consists of channel ordering given by *Line 3*; *Line 4* performs nulling and computes the decision statistic; *Line 5* quantizes the computed decision statistic to yield the decision; *Line 6* performs cancellation by decision feedback, and *Line 7* computes the new pseudoinverse for the next iteration.

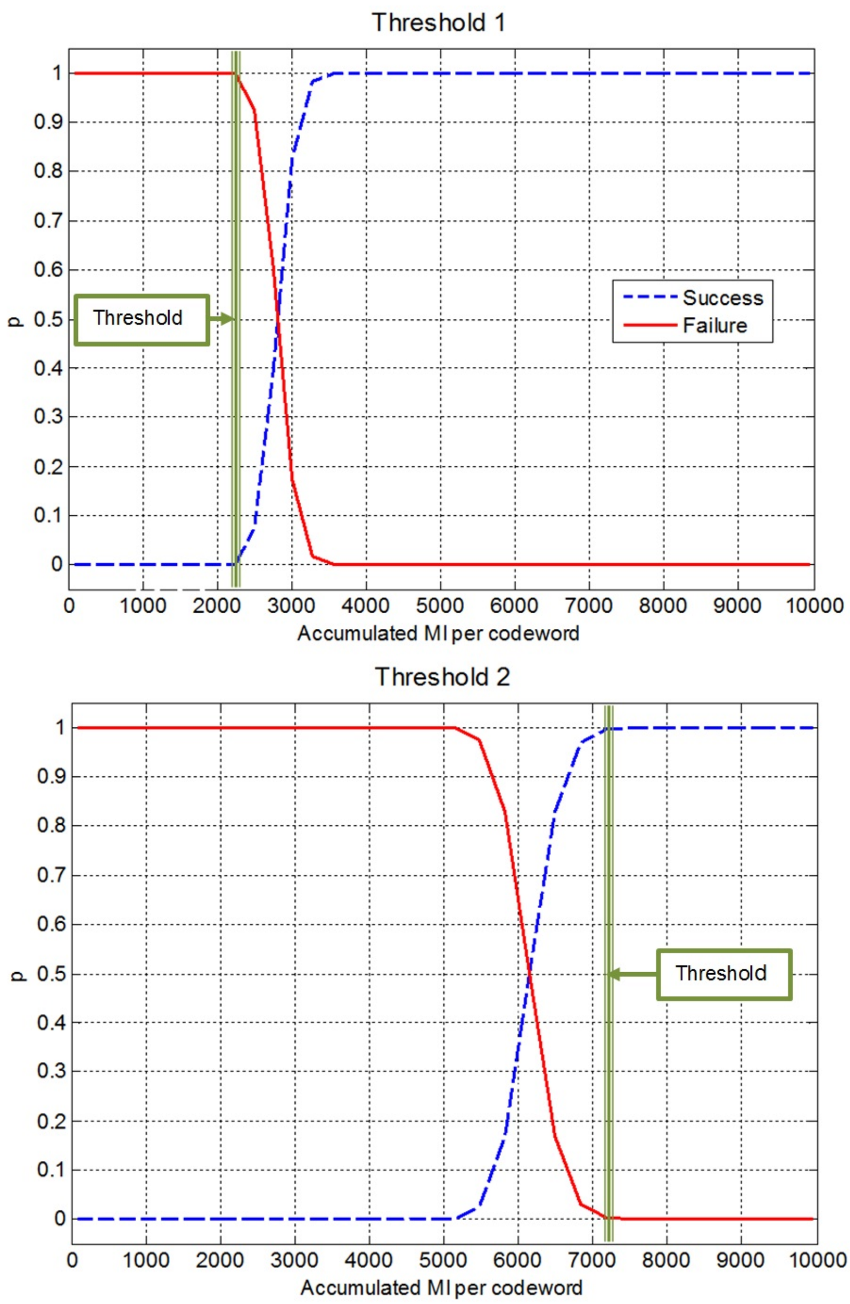

Figure 2. Probability of Receiver Successes and Failures on 4×4 MIMO where threshold 1 (a) is for FSD method and (b) is threshold 2, for V-BLAST/ZF method

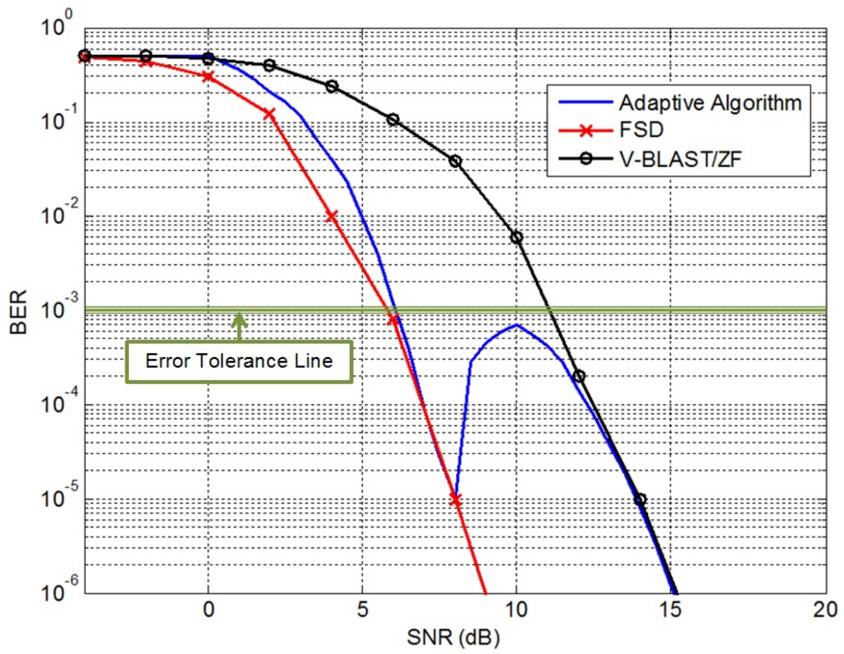

Figure 3. Performance Measurement of BER on Complex 4×4 MIMO System

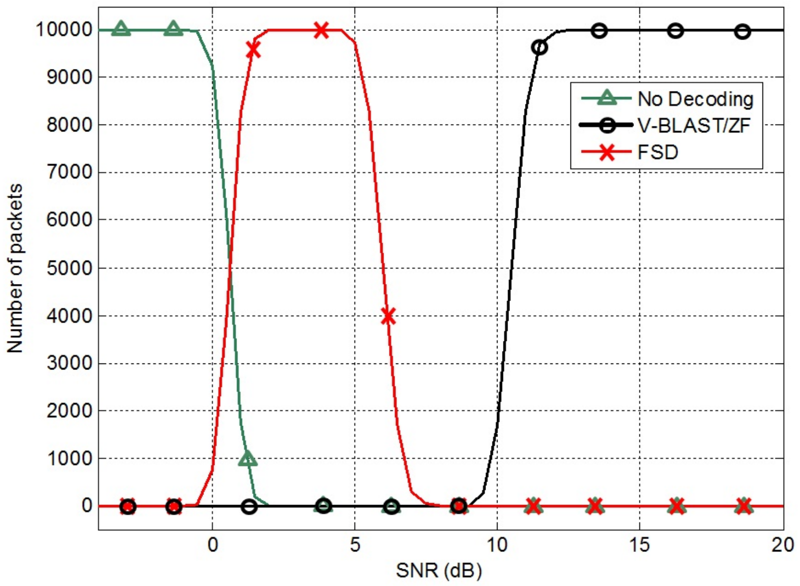

Figure 4. Algorithm Switching Selection in Receiver

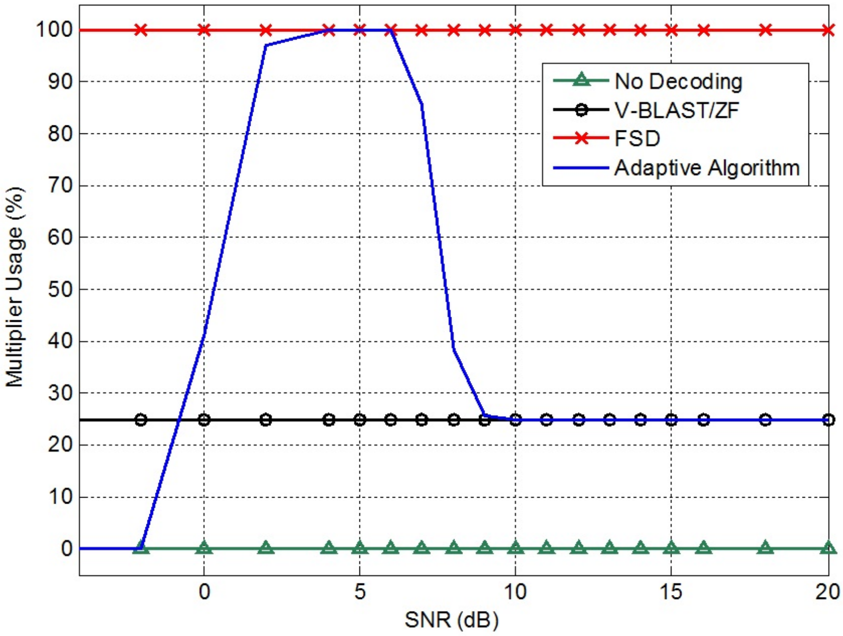

Figure 5. Complexity Measurements of Multiplier Counts between Different MIMO Detection Schemes

Table 2. Adaptive Algorithm

Pseudo-Code

Channel realization: $\{\mathbf{H_1}, \mathbf{H_2}, \cdots, \mathbf{H_k}\}$

for $\mathbf{r}_i \leq \mathbf{r}_k$

$\bar{I}(\mathbf{H}_k) \triangleq \log_2 \det(I + \frac{\mathbf{H}_k \mathbf{H}_k^T}{N_0})$

if $\bar{I}_i \leq$ Threshold 1
$\mathbf{r}_i$ error, request retransmission

elseif Threshold 1 $\leq \bar{I}_i \leq$ Threshold 2
$\mathbf{r}_i$ with low MI : FSD

else $\bar{I}_i \geq$ Threshold 2
$\mathbf{r}_i$ with high MI : V-BLAST/ZF

endif

endfor

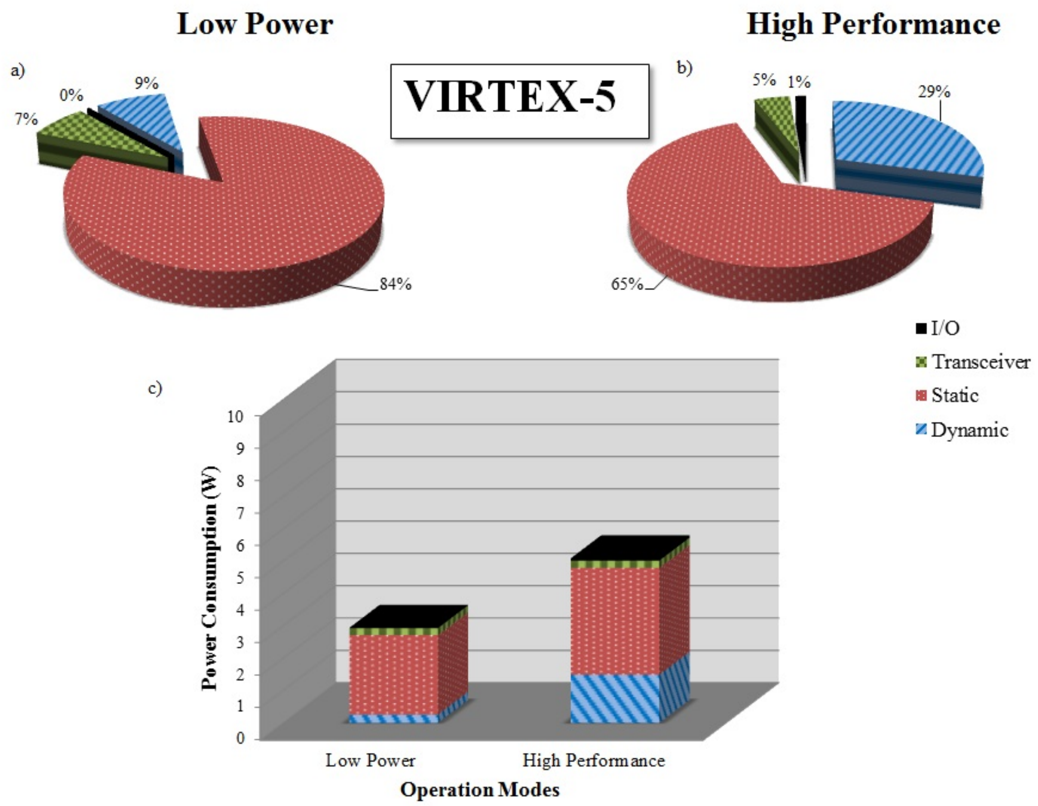

Figure 6. Total Power Usage in Xilinx®Virtex-5 Hardware Apparatus

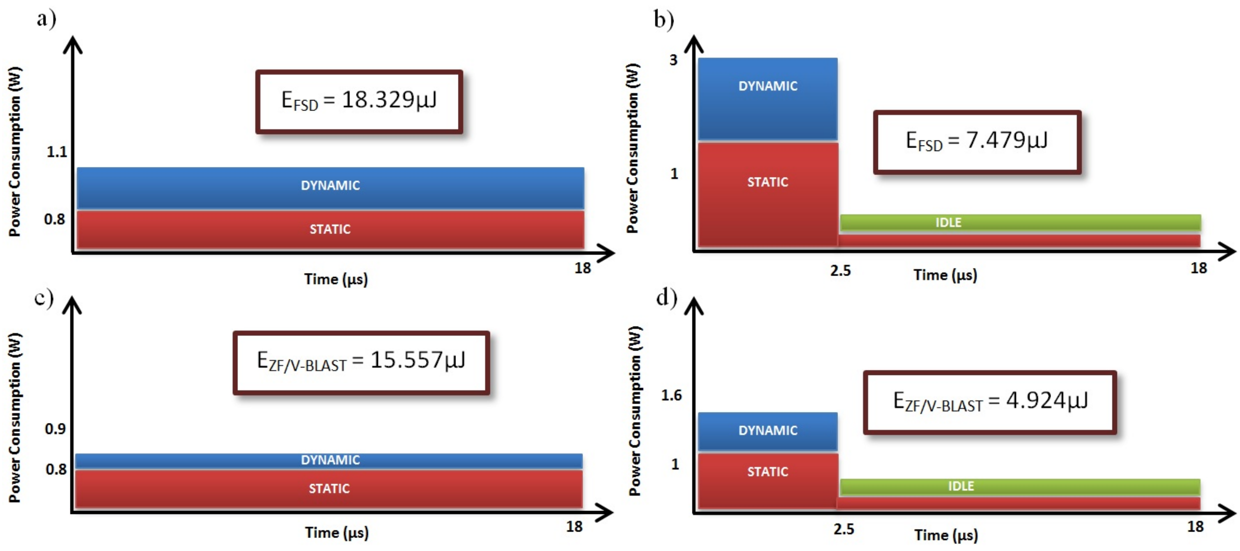

Figure 7. MIMO Detection FSD (a) and (b) in comparison with V-BLAST/ZF (c) and (d) for Low Power Mode and High Performance Mode respectively

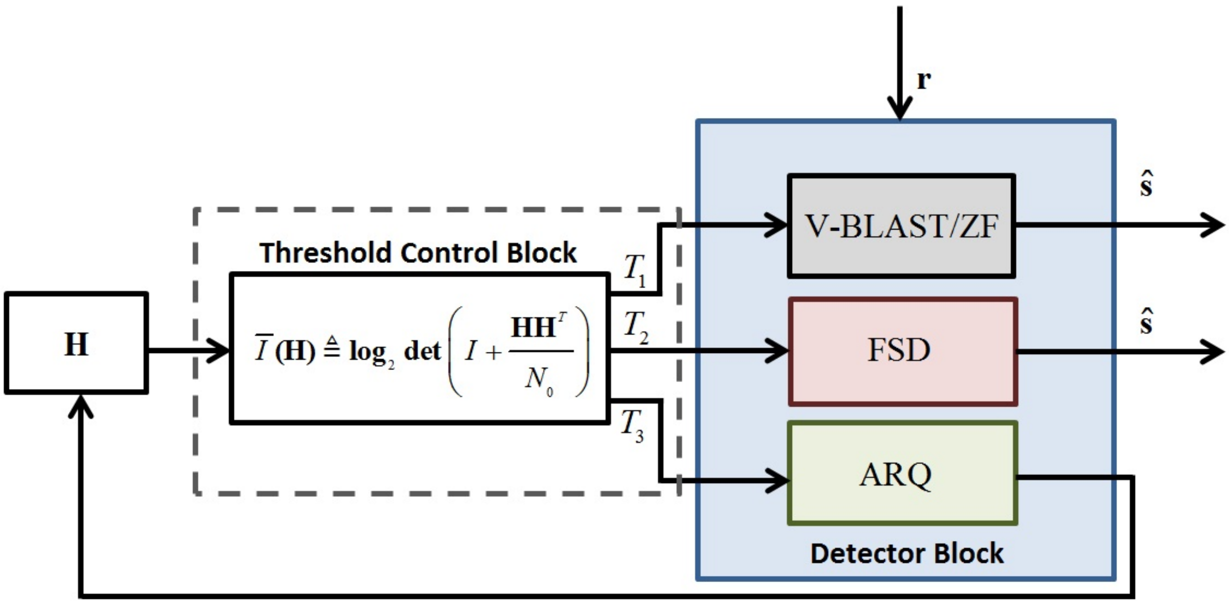

Figure 8. Simple Adaptive Algorithm Implementation Model

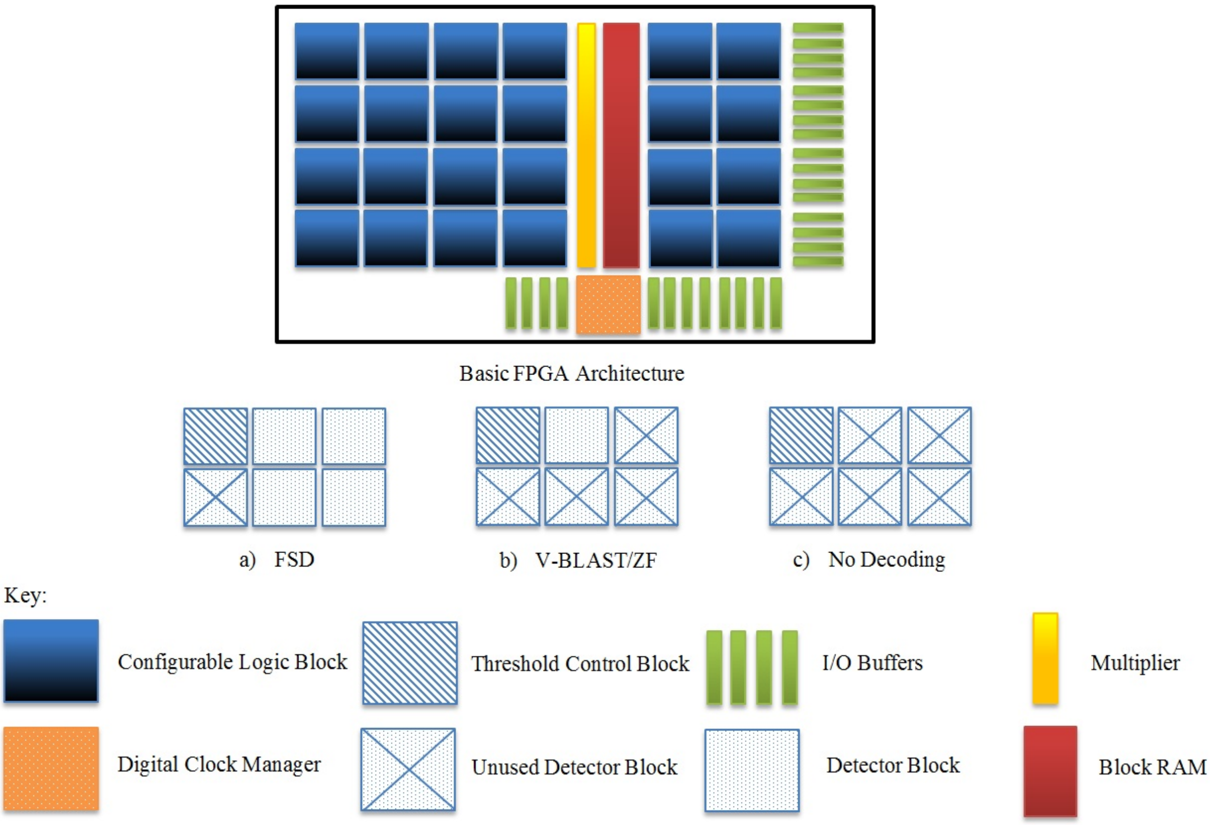

Figure 9. Total Resource Allocation of Adaptive Algorithm on a Basic FPGA Architecture

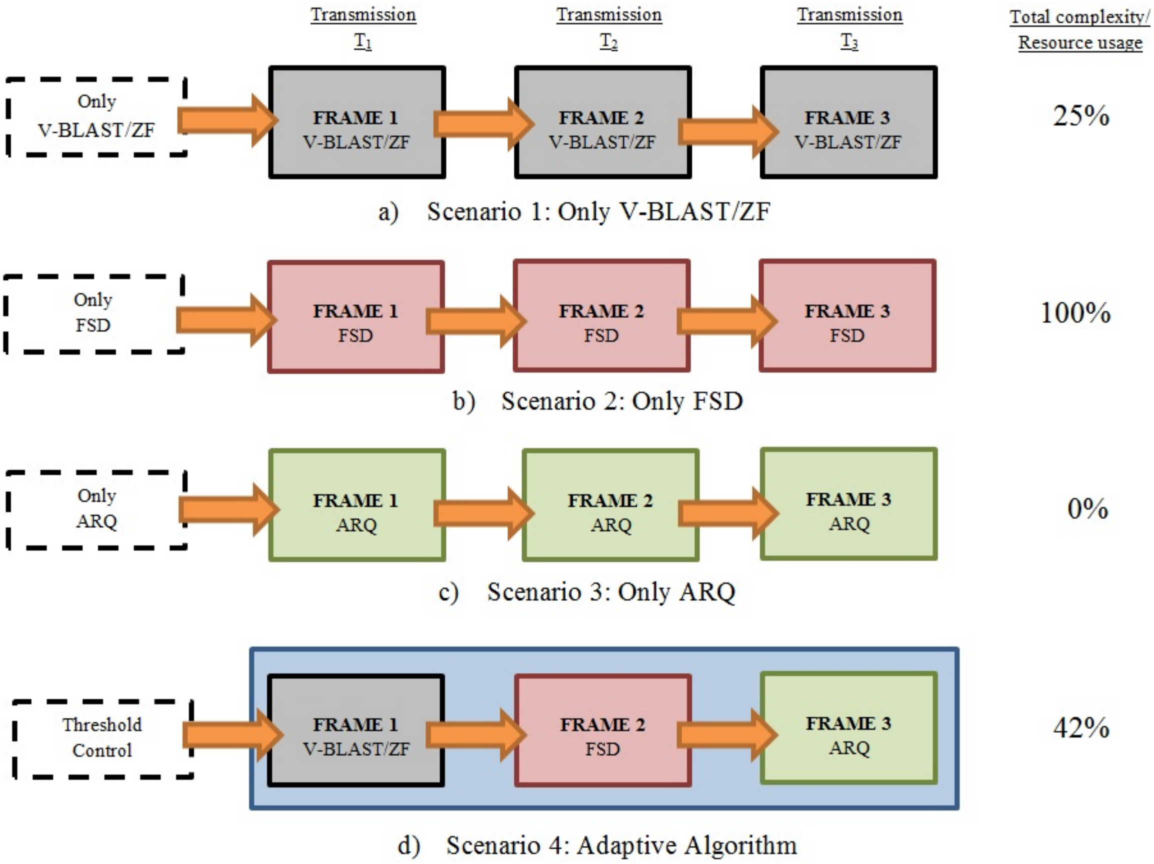

Figure 10. Basic Overview of the Inner Workings of the Adaptive Algorithm

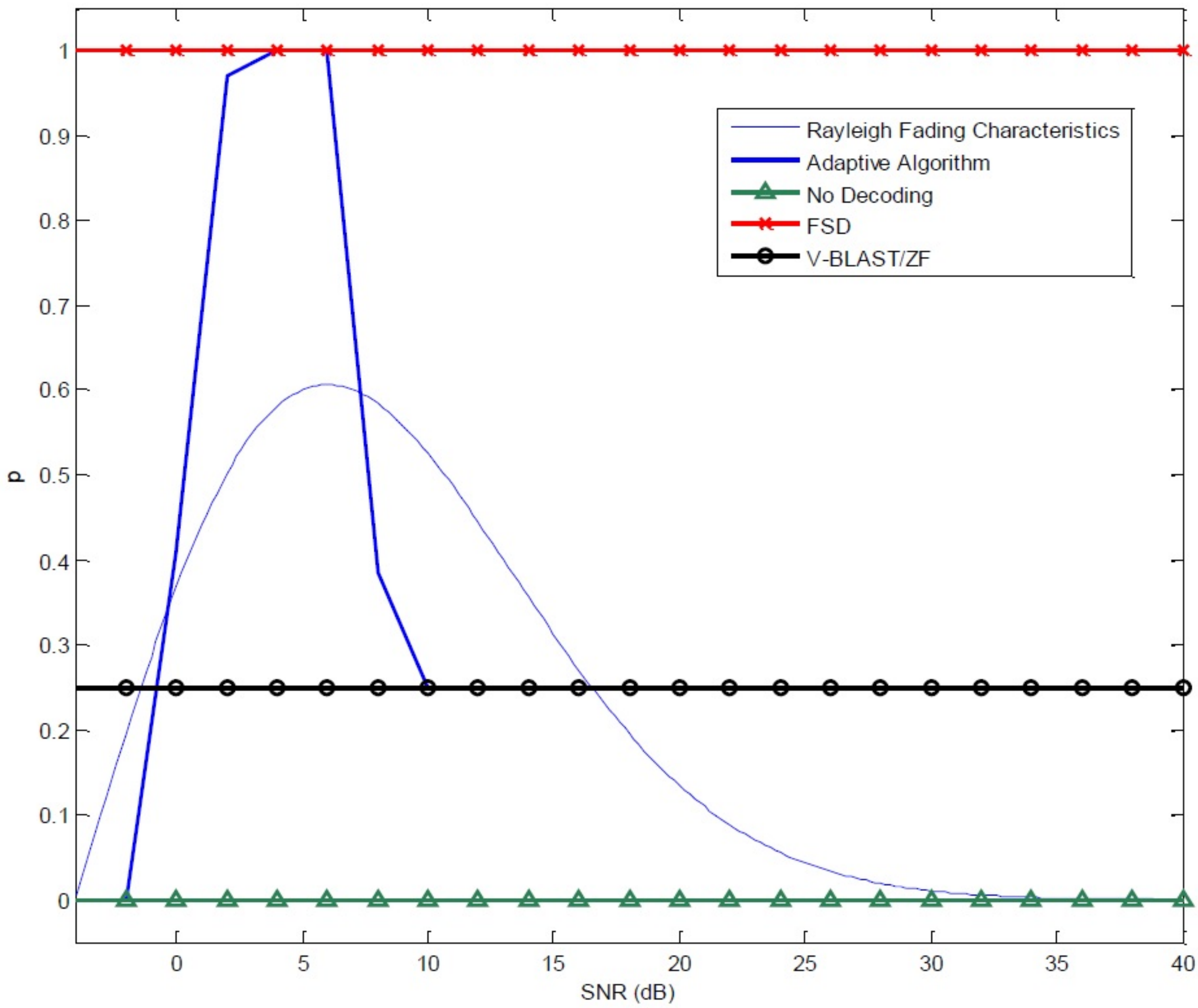

Figure 11. Behaviors of Different Detection Algorithms in a Rayleigh Fading Channel

Table 3. Virtex-5 Resource Utilization of Adaptive Algorithm

Logic Resource Utilization	Used	Available	Utilization
Slice Registers	13,683	149,760	9%
Flip Flops	4688	37,440	12%
4-Input LUTs	12,161	149,760	8%
DSP48E	132	1,056	12%
Memory (RAM)	28	516	5%

Virtex 5: XC5VLX330TFF1738

MIMO setup 4x4 *Modulation Scheme 16-QAM* *Bit Frame Size 1024 bits*

Operation Mode Parameters	Low Power	High Performance
Core Voltage	0.95V	1.05V
Clock Frequency	60MHz	400MHz
Max Throughput	240Mbps	1200Mbps

Table 4. Experiment Parameters of Adaptive Algorithm