LOG-DOMAIN ITERATIVE SPHERE DECODER WITH SYMBOL SORTING

P.R. Botha* **and B.T. Maharaj†**

* *Dept. of Electrical, Electronic & Computer Engineering, Corner of University Road and Lynnwood Road, University of Pretoria, Pretoria 0002, South Africa E-mail: prbotha@ieee.org*

† *Faculty of Engineering, Built environment and IT, Corner of University Road and Lynnwood Road, University of Pretoria, Pretoria 0002, South Africa E-mail: ssinha@saiee.org.za E-mail: sunil.maharaj@up.ac.za*

Abstract: In this paper the authors propose modified branch and pruning metrics for the sphere decoder to facilitate the use of apriori information in the sphere decoder. The proposed sphere decoder operates completely in the log-domain. Additionally the effect of order in which the symbols are decoded on the computational requirements of the decoder are investigated with the authors proposing a modification of the sorted QR decomposition (SQRD) algorithm to incorporate apriori information. The modified SQRD algorithm is shown to reduce the average number of computations by up to 95%. The apriori sphere decoder is tested in an iterative multiple input multiple output (MIMO) decoder and shown to reduce the bit error rate (BER) by an order of magnitude or provide approximately a one decibel improvement.

Key words: Sphere decoder, iterative decoding, MIMO, TAST, QR.

1. INTRODUCTION

The ever increasing demand for affordable, high speed and reliable wireless communication has led to the development of various technologies such as multiple input multiple output (MIMO) systems. The usage of multiple transmit and receive antennas of a MIMO system can potentially result in a significant increase in the capacity of the communication channel. Using MIMO in tandem with other communication techniques, such as orthogonal frequency division multiplexing (OFDM), enables the transmission of information over time, space and frequency.

The performance of a wireless communication system can also be greatly increased by exploiting some or all of the diversity, independently faded signal paths, in the wireless channel. Various coding schemes that aide in the utilisation of all of the available diversity and/or capacity in a MIMO channel have been proposed [1–4]. These codes generally require *joint* decoding using methods such as the Zero-Forcing (ZF), Minimum Mean Squared Error (MMSE) and Sphere decoder (SD) [5, 6]. Forward Error Correction FEC is also typically used to improve the performance of the system. Accurate soft outputs from the decoder can typically provide a 2dB gain [7] and is therefore desired. In this paper Max-log-map Hard-to-Soft decoding is used [8].

Drawing on the concept of turbo-codes [9], iteratively decoding can also provide an increase in performance. This entails a MIMO decoder that can use soft-inputs generally as apriori information. In [10] the authors provide a means to incorporate apriori information in the ZF and MMSE decoder. An alternative method that can be used to implement a soft-input sphere decoder [8] is also described and shown to be limited to BPSK and QPSK signal constellations. A similar approach is used by the authors of [11] without addressing nor identifying the limitations of their approach.

In this paper the authors will describe a modification to the branch and pruning metrics of the sphere decoder of [6] that incorporates apriori information. Studer et al in [12] propose what seems to be a similar approach to the approach used in this paper. They, however, focus on optimizing the MAP values whereas in this paper the focus is on the complexity introduced by the delayed pruning due to apriori information. It will be shown empirically that for iterative decoding the modification typically results, contrary to what is expected, in a reduction in the computations required by the sphere decoder is achieved.

Additionally, the effects of various symbol sorting strategies on the computational complexity of the apriori sphere decoder will be shown. It will be shown that a modified instance of the sorted QR decomposition (SQRD) [13] yields the greatest reduction in computations.

Notation: In this paper we use the following notation. Vectors are denoted by boldface lowercase letters. Matrices are denoted by boldface uppercase letters. Superscripts $\mathcal{T}$ and $\mathcal{H}$ denote the transpose and Hermitian transpose operations, respectively; $diag(d_1 \dots d_N)$ denotes a $N \times N$ diagonal matrix with diagonal entries $d_1 \dots d_N$. $\mathbf{F_N}$ is the $N \times N$ discrete Fourier transform (DFT) matrix and $(\cdot)$ denotes the dot product of two vectors.

2. MAP MIMO DECODING

The received signal in a MIMO system may be described as:

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n}, \tag{1}$$

where $\mathbf{H}$ is the equivalent channel matrix, $\mathbf{x}$ is the transmitted data vector and $\mathbf{n}$ is the noise vector. In general $\mathbf{H}$, $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{n}$ are complex valued. The optimal decoding of the receive signal involves the MAP calculation of the value of a bit. This probability is generally expressed in terms of log likelihood ratios (LLR) defined as:

$$\lambda_i = \ln\left(\frac{P(b_i=+1)}{P(b_i=-1)}\right), \tag{2}$$

where λ is the LLR and $P(b_i = x)$ is the probability that $b_i = x$. BPSK signalling has been assumed for the individual bit values instead of 0 and 1.

Using equation 2 and Bayes' rule, the MAP solution can be given as:

$$\tilde{\mathbf{x}} = \arg\min_{\mathbf{x}\in\mathbb{X}} \ln\{P(\mathbf{y}|\mathbf{x},\mathbf{H})\mathbf{P}(\mathbf{x})\}, \tag{3}$$

where the conditional probability $P(\mathbf{y}|\mathbf{x},\mathbf{H})$ is given by:

$$P(\mathbf{y}|\mathbf{x},\mathbf{H}) = \frac{1}{\pi^{N_R}\det(\Sigma)}\cdot \exp\left[-(\mathbf{y}-\mathbf{H}\mathbf{x})^{\mathcal{H}}\Sigma^{-1}(\mathbf{y}-\mathbf{H}\mathbf{x})\right], \tag{4}$$

where N_R is the number of receive antennas and Σ denotes the covariance matrix of $(\mathbf{y}-\mathbf{H}\mathbf{x})$ given by:

$$\Sigma = E\left[(\mathbf{y}-\mathbf{H}\mathbf{x})(\mathbf{y}-\mathbf{H}\mathbf{x})^{\mathcal{H}}\right]. \tag{5}$$

Since the noise $\mathbf{n}$ is assumed to be AWGN the following simplification may be made:

$$\Sigma = \sigma_n^2\mathbf{I}_{\mathbf{N_R}}, \tag{6}$$

$$\det(\Sigma) = \sigma_n^{2N_R}, \tag{7}$$

$$\Sigma^{-1} = \frac{1}{\sigma_n^2}, \tag{8}$$

where σ_n^2 is the noise power and I_k is the $k\times k$ identity matrix. Using equations 2 to 8 the MAP solution can be expressed as:

$$\lambda_i^p \approx \min_{\mathbf{x}\in\mathbb{X}_{\exists}^{\exists-\nu}}\left\{\frac{||\mathbf{y}-\mathbf{H}\mathbf{x}||^2}{\sigma_n^2} - \frac{1}{2}\mathbf{b}\cdot\lambda^{\mathbf{a}}\right\} - \min_{\mathbf{x}\in\mathbb{X}_{\exists}^{\exists+\nu}}\left\{\frac{||\mathbf{y}-\mathbf{H}\mathbf{x}||^2}{\sigma_n^2} - \frac{1}{2}\mathbf{b}\cdot\lambda^{\mathbf{a}}\right\}, \tag{9}$$

where λ_i^p is the LLR of the MAP of the i^{th} bit, b_i, with $\mathbb{X}_{\exists}^{\exists\curvearrowleft}$ denoting the set of all transmit vectors having $b_i = x$. $\mathbf{b}$ denotes the vector of bits associated with $\mathbf{x}$ and λ^a is the vector of apriori LLRs. Additional use of the max-log approximation:

$$\ln\left(\sum_i e^{x_i}\right) \approx \max_i x_i, \tag{10}$$

was made in the derivation of equation 9.

Thus the decoding problem involves the calculation of the *most* likely metric for both possible values of the bit in question. Non-iterative decoders typically assume that there is no apriori information.

3. CLASSICAL SPHERE

The sphere decoder in [6] is essentially a tree search algorithm. It is a variation of the A* tree search algorithm [14, 15]. In the classical sphere decoder the A* algorithm is used to solve the following system of linear equations:

$$\tilde{\mathbf{x}} = \arg\min_{\mathbf{x}\in\mathbb{X}}\left\{\frac{\|\mathbf{y}-\mathbf{H}\mathbf{x}\|^{\mathbf{2}})}{\sigma_n^2}\right\}, \tag{11}$$

in order to obtain the MAP transmitted channel symbols and, thus, transmitted data. This is essentially the same as equation 9 without the apriori information. For use with the sphere decoder, equation 11 is transformed into a triangular system. In [6], as with most other formulations, the Cholesky decomposition:

$$\mathbf{R}^{\mathcal{H}}\mathbf{R} = \mathbf{A}, \tag{12}$$

where $\mathbf{R}$ is upper triangular, is used. Equation 11 is then transformed into the triangular equation:

$$\tilde{\mathbf{x}} = \arg\min_{\mathbf{x}\in\mathbb{X}}\left\{(\mathbf{x}-\hat{\mathbf{x}})^{\mathcal{H}}\mathbf{R}^{\mathcal{H}}\mathbf{R}(\mathbf{x}-\hat{\mathbf{x}})\right\}, \tag{13}$$

where $\hat{\mathbf{x}}$ is the zero-forcing solution, and $\mathbf{R}^{\mathcal{H}}\mathbf{R} = \mathbf{H}^{\mathcal{H}}\mathbf{H}$ and σ_n^2 has been omitted as it is a constant and does not affect the result. The zero-forcing solution, $\hat{\mathbf{x}}$ can be obtained using, for example, the Moore-Penrose Pseudo inverse:

$$\mathbf{A}^{\dagger} = (\mathbf{A}^{\mathcal{H}}\mathbf{A})^{-\mathbf{1}}\mathbf{A}^{\mathcal{H}}, \tag{14}$$

as:

$$\hat{\mathbf{x}} = \mathbf{H}^{\dagger}\mathbf{y}. \tag{15}$$

Alternatively the QR decomposition, SV decomposition, LU decomposition or even the Cholesky decomposition can be used to compute the zero-forcing solution. Equation 11 can also be transformed into a triangular system using the QR decomposition:

$$\tilde{\mathbf{x}} = \arg\min_{\mathbf{x}\in\mathbb{X}}\left\{\|\mathbf{Q}^{\mathcal{H}}\mathbf{y}-\mathbf{R}\mathbf{x}\|^{\mathbf{2}}\right\}, \tag{16}$$

where $\mathbf{QR} = \mathbf{H}$. This formulation will be used henceforth.

The tree structure arrises from the observation that $x \in \mathbf{X}$, i.e. the channel symbol x is part of a finite set $\mathbf{X}$ and that when performing the back substitution, the value of the current symbol is dependent on the previous symbols.

In equations 13 & 16 it is important to note that when the equations are being evaluated, the evaluations can be calculated by traversing the tree on a *symbol-by-symbol* basis and are *monotonously* increasing:

$$T_{n-1} \geq T_n, \tag{17}$$

where it should be noted that the tree is traversed from node N upwards in an upper triangular system. Thus, as soon as the metric for a branch, T_n in the tree exceeds the current lowest metric, $T_{\min}$, it will *always* exceed that metric and therefore the branch can be pruned from the solution space. In such cases, the A* algorithm is the optimal tree-traversal algorithm [14]. In the next section it will be shown that should apriori information be incorporated into the MAP solution, then the branch metric is no longer monotonously increasing. Thus, a branch can no longer be pruned from the tree once its metric exceeds the current minimum.

4. APRIORI SPHERE DECODER

The MAP solution including apriori information is given by:

$$\tilde{\mathbf{x}} = \arg\min_{\mathbf{x}\in\mathbb{X}} \left\{ \frac{\|\mathbf{y}-\mathbf{H}\mathbf{x}\|^2}{\sigma_n^2} - \frac{\boldsymbol{\lambda}\cdot\mathbf{b}}{2} \right\}, \tag{18}$$

where $\boldsymbol{\lambda}$ is the Log-Likelihood Ratio of the apriori information and $\mathbf{b}$ is the bitwise data vector, in BPSK, representation of $\mathbf{x}$.

Equation 18 can be partly transformed into a triangular system as in section 3.:

$$\tilde{\mathbf{x}} = \arg\min_{\mathbf{x}\in\mathbb{X}} \left\{ \frac{\|\mathbf{Q}^{\mathcal{H}}\mathbf{y}-\mathbf{R}\mathbf{x}\|^2}{\sigma_n^2} - \frac{\boldsymbol{\lambda}\cdot\mathbf{b}}{2} \right\}, \tag{19}$$

where it should be noted that σ_n^2 can no longer be omitted from the calculations. Whilst eq. 19 can still be represented as a tree structure, it is evident that the equation is no longer monotonously increasing. In the event that the current branch metric, T_n exceeds the minimum metric, $T_{\min}$ it is still possible that the remaining apriori information can result in the branch metric being lower than the current minimum.

Therefore it is no longer possible to prune the branch the moment the branch metric exceeds the current minimum metric as was the case with the classical sphere decoder. It should be mentioned that for BPSK and QPSK channel symbols it is possible to incorporate the apriori information vector, $\boldsymbol{\lambda}$, into the received vector, $\mathbf{y}$ [8]. This approach does not work for other modulation constellations [10].

It is therefore necessary to modify the pruning metric to incorporate the apriori information in $\boldsymbol{\lambda}$. The pruning metric now only prunes a branch, if the current branch metric exceeds the current minimum *and* the remaining apriori information is insufficient to change this. Thus the branch is pruned when:

$$T_n - \frac{1}{2}\sum_{i=n}^{N} \lambda_i > T_{\min}, \tag{20}$$

where λ_i is the apriori information associated with the channel symbol x_i. Whilst the new pruning metric allows the sphere decoder to incorporate the apriori information, it potentially delays the pruning of branches potentially increasing the average number of computations. However, it is possible to re-arrange the tree such that the branches are pruned earlier as is discussed in the next section.

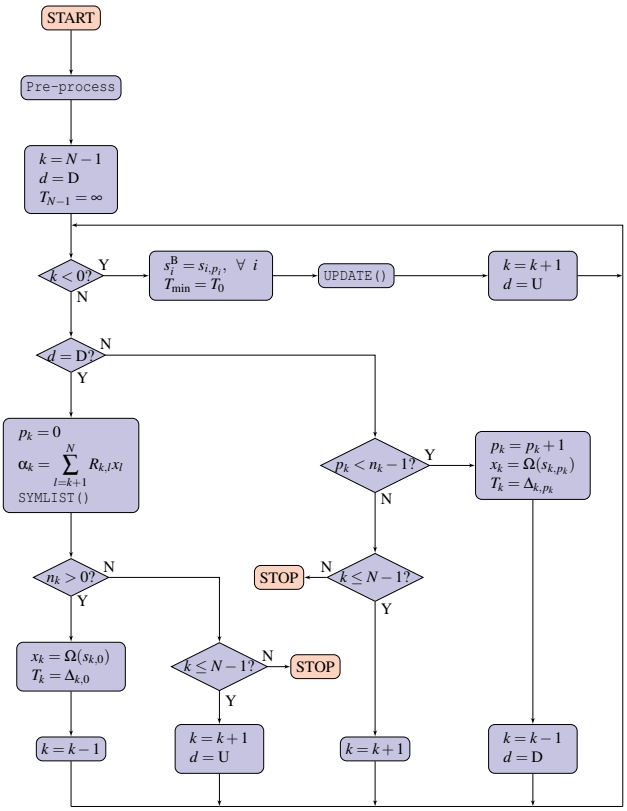

Figure 1: Flow diagram of the tree search algorithm.

5. RE-ORDERING OF SYMBOLS

In order to prune branches on the tree, it is known that having the diagonal of $\mathbf{R}$ be ordered *ascendantly*, i.e. $|r_{n+1,n+1}| \geq |r_{n,n}|$ can speed up the sphere decoder [16]. Similarly the apriori sphere decoder can be sped up by ordering the symbols. Two approaches have been investigated: one based solely on the apriori information $\boldsymbol{\lambda}$, the other based on apriori information *and* the channel $\mathbf{H}$.

5.1 *Re-ordering by* $\boldsymbol{\lambda}$

The reasoning behind ordering the tree based on the apriori information is the idea of reducing the impact of the remaining apriori information on the classical pruning metric. Thus, ideally the tree should be traversed in the *descending* order of λ_n, where λ_n is:

$$\lambda_n = \sum_{i=0}^{N_b-1} |\lambda_i|, \tag{21}$$

i.e. the sum of the absolute values of the LLRs of the bits of symbol n. In this manner the sum of absolute values of the LLRs diminishes rapidly as the tree is traversed by the decoder. Since the system being decoded has been transformed into an upper triangular system, this means

that the channel symbols and their respective columns in $\mathbf{H}$ must be sorted based on the *ascending* values of λ_n. λ_n is the total apriori information for symbol x_n given by the summation of the absolute values of the LLRs of each bit in symbol x_n.

5.2 Re-ordering by λ and $\mathbf{H}$

It is known that performing the triangular transformation in such a way that the diagonal elements, of the upper triangular system, are increasing results in a speedup of the sphere decoder [16]. A sub-optimal approximation is to sort the column of $\mathbf{H}$ according to their Frobenius norms.

In an attempt to combine the ordering of the symbols with the norms of the columns of $\mathbf{H}$ with their ordering with respect to their apriori information, the following metric, inspired by eq. 19, is proposed:

$$C_n = \frac{1}{\sigma_n^2} \sum_{i=0}^{N_{RX}-1} |H_{i,n}|^2 + \frac{1}{2}\lambda_n. \tag{22}$$

The symbols are then ordered ascendantly according C_n. Equation 22 scales the contribution of the channel by the noise power. The first term is essentially an indicator of the SNR of symbol n.

5.3 Sorted QR Decompostion

The sorted QR decompostion proposed in [13] attempts to order the diagonal elements of $\mathbf{R}$ in increasing orders of magnitude. The algorithm is:

$\mathbf{R} = \mathbf{0}$, $\mathbf{Q} = \mathbf{H}$ & $\mathbf{P} = \mathbf{I_{N_T}}$
for $n = 0$ to $N_T - 1$ **do**
  $k_n = \underset{j=n,\cdots,N_T}{\arg\min} |\mathbf{q_j}|^2$
  exchange columns n and k_n in $\mathbf{Q}$, $\mathbf{R}$ and $\mathbf{P}$
  $R_{n,n} = |\mathbf{q_n}|$
  $\mathbf{q_n} = \mathbf{q_n}/\mathbf{R_{n,n}}$
  for $j = n+1$, to $N_T - 1$ **do**
    $R_{n,j} = \mathbf{q_n^{\mathcal{H}}}\mathbf{q_j}$
    $\mathbf{q_j} = \mathbf{q_j} - \mathbf{R_{n,j}}\mathbf{q_n}$
  end for
end for

where $\mathbf{q_n}$ is the n^{th} column of matrix $\mathbf{Q}$. $\mathbf{P}$ is the permutation matrix by which the columns of $\mathbf{H}$ and the rows of $\mathbf{x}$ have been permuted.

5.4 Modified Sorted QR Decomposition

The standard SQRD does not make use of information other than the channel matrix $\mathbf{H}$. The authors therefore propose the following modified SQRD (mSQRD)algorithm:

$\mathbf{R} = \mathbf{0}$, $\mathbf{Q} = \mathbf{H}$ & $\mathbf{P} = \mathbf{I_{N_T}}$
for $n = 0$ to $N_T - 1$ **do**
  $\lambda_k^s = \sum_{k=0}^{N_b} |\lambda_{nN_b+k}|$
end for
for $n = 0$ to $N_T - 1$ **do**
  $k_n = \underset{j=n,\cdots,N_T}{\arg\min} \left\{ \frac{\|\mathbf{q_j}\|^2}{\sigma_n^2} + \frac{1}{2}\lambda_j^s \right\}$
  exchange columns n and k_n in $\mathbf{Q}$, $\mathbf{R}$, $\mathbf{P}$ and λ^s
  $R_{n,n} = |\mathbf{q_n}|$
  $\mathbf{q_n} = \mathbf{q_n}/\mathbf{R_{n,n}}$
  for $j = n+1$, to $N_T - 1$ **do**
    $R_{n,j} = \mathbf{q_n^{\mathcal{H}}}\mathbf{q_j}$
    $\mathbf{q_j} = \mathbf{q_j} - \mathbf{R_{n,j}}\mathbf{q_n}$
  end for
end for

where σ_n^2 is the noise variance and N_b is the number of bits per symbol.

6. ALGORITHM DETAILS

The apriori sphere decoder is based on the implementation by [6]. It has been modified to use the QR decomposition instead of the Cholesky decomposition. The flow diagram for the apriori sphere decoder is shown in figure 1.

6.1 Pre-processing

The `Pre-process` block is responsible for calculating the best symbol order to obtain the permutation matrix $\mathbf{P}$, the QR decomposition of $\mathbf{H}$ as well as the calculation of the cumulative LLRs λ^c:

for $i = 0$, to $N - 1$ **do**
  $\lambda_i^c = \lambda_{i-1}^c + \sum_{j=0}^{N_b-1} |\lambda_{i \cdot N_b + j}|$
end for
$\mathbf{P} \leftarrow$ Optimal symbol order for decoding
$\mathbf{Q}, \mathbf{R} \leftarrow \mathrm{qr}(\mathbf{HP^{-1}})$
$\mathbf{y} \leftarrow \mathbf{Q^{\mathcal{H}}}\mathbf{y}$

where $\lambda_{-1} = 0$.

6.2 Symlist

The `SYMLIST()` function calculates the possible symbols for the level and calculates their metrics. The symbols are then sorted ascendantly according to the metric. The variable n_k is then assigned the number of symbols that are smaller than the pruning metric:

for $i = 0$, to $Ns - 1$ **do**
  $\beta_i = \sum_{j=0}^{N_b-1} \lambda_{j+N_b k} \hat{b}_j^i$
  $\Delta_i^k = \frac{\|y_k - \alpha_k - R_{k,k}\Omega(i)\|^2}{\sigma_n^2} - \frac{1}{2}\beta_i + \Delta_{p_{k+1}}^{k+1}$
end for
`sort(`Δ^k, s^k`)`
$n_k = \underset{i}{\arg\max} \left\{ \Delta_i^k < T_{\min} + \frac{1}{2}\lambda_k^c \right\}$
$T_k = \Delta_0^k$

where $\hat{b}_j^i$ denotes the j^{th} bit represented by symbol i. The `sort()` function sort the symbols ascendantly according to their Δ_i^k metrics and stores the symbol order in s_i^k. The

number of symbols that are less than the pruning metric is stored in n_k.

Performance metric: The `SYMLIST()` function is the function where the vast majority of the calculations are performed. It is also visited for each node in the tree. Thus, the number of times the `SYMLIST()` function is called is used as a performance metric in evaluating the relative performance of the sphere decoder with regard to various symbol ordering strategies. This allows the metric to be independent of the specific implementation architecture and programming.

6.3 Update

The `UPDATE()` function uses the newly found minimum metric T_{min} to prune the tree. ItiIterates through the levels and updates the n^k value for each level by only keeping the symbols whose metrics are less than the new pruning metric:

for $k = 0$, to $N-1$ **do**

$$n^k = \arg\max_j \left\{ \Delta_j^k < T_{\min} + \frac{1}{2}\lambda_k^c \right\}$$

end for

7. SYSTEM DESCRIPTION

The iterative decoding of linear pre-coded (LP) MIMO will be used to test the performance of the apriori sphere decoder. A short LDPC code is used to provide the apriori information to the MIMO decoder from the second iteration onwards.

7.1 Linear Pre-coding

Linear pre-coding of a MIMO system using threaded algebraic space time (TAST) codes enables the exploitation of all of the diversity in a MIMO channel without sacrificing transmission rate [2]. This is achieved by linearly mapping the channel symbol vector to a new encoded channel symbol vector. The linearity of the mapping enables the mapping to be expressed as a matrix multiply. Let $\mathbf{x} = [\mathbf{x_1}, \cdots \mathbf{x_m}]^{\mathcal{T}}$ be a data vector of length N complex channel symbols taken from a modulation alphabet $\mathbf{X}$ such as QPSK or M-QAM. Let Θ be a unitary matrix of dimensions $N \times N$ defined as [2]:

$$\Theta = \mathbf{F_N^{\mathcal{H}}} \mathrm{diag}(\mathbf{1}, \varphi, \ldots, \varphi^{\mathbf{N-1}}), \tag{23}$$

where $\varphi = \exp(\mathbf{i}2\pi/4N)$ and $\mathbf{F_N}$ is the $N \times N$ discrete Fourier transform matrix. The mapping operation can thus be expressed as:

$$\mathbf{s} = \Theta \mathbf{x}, \tag{24}$$

where $\mathbf{s}$ is the newly encoded channel symbol vector. In a noiseless environment the correct decoding of the entire $\mathbf{x}$ channel symbol vector only requires the correct reception of a single encoded symbol of $\mathbf{s}$ achieving diversity equal to the rank of Θ, R_Θ. Two or more streams of LP encoded vectors can be layered together in order to exploit the full rate of the channel. Diophantine numbers aide the decoder in separating the various streams from each other [2]. The Diophantine number for each layer is obtained by : $\phi_n = \varphi^n$, $n = 0, \ldots, N_L - 1$, where N_L is the number of layers.

The disadvantage of TAST codes is their decoding complexity as $N_L \times N$ symbols need to be *jointly* decoded.

7.2 MIMO System

In this paper the MIMO channel is modelled as an $N_R \times N_T$ matrix, $\mathbf{H}$. The elements of $\mathbf{H}$ are each i.i.d. complex Gaussian with zero mean and unit variance. This corresponds to an ideal Rayleigh fading channel with no correlation. Time and frequency diversity can be expressed as a matrix with MIMO channel blocks on the diagonal. In this paper it is also assumed that time and frequency are independently faded. The full system equation then becomes:

$$\mathbf{y} = \mathbf{H}'\Theta'\mathbf{P}\mathbf{x}' + \mathbf{n}, \tag{25}$$

with $\mathbf{P}$ a permutation matrix that determines the manner in which the two streams are threaded together.

With layering, the matrices $\mathbf{H}', \Theta'$ and $\mathbf{x}'$ are given as:

$$\mathbf{H}' = \begin{bmatrix} \mathbf{H_0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{H_1} & \mathbf{0} & \mathbf{0} \\ & & \ddots & \\ \mathbf{0} & \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{H_{N_H-1}} \end{bmatrix}, \tag{26}$$

$$\Theta' = \begin{bmatrix} \Theta\varphi^0 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \Theta\varphi^1 & \mathbf{0} & \mathbf{0} \\ & & \ddots & \\ \mathbf{0} & \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \Theta\varphi^{N_L-1} \end{bmatrix}, \tag{27}$$

$$\mathbf{x}' = \begin{bmatrix} \mathbf{x_0^{\mathcal{T}}} & \mathbf{x_1^{\mathcal{T}}} & \cdots & \mathbf{x_{N_L-1}^{\mathcal{T}}} \end{bmatrix}^{\mathcal{T}}, \tag{28}$$

where N_H is the number of individual MIMO transmissions that are made, either in frequency or time, and is given by $N_L \times R_\Theta / N_T$. The simulations in this paper are made with $R_\Theta = 4$, $N_T = N_R = N_L = 2$ and with a resultant $H_N = 4$.

7.3 Turbo Structure

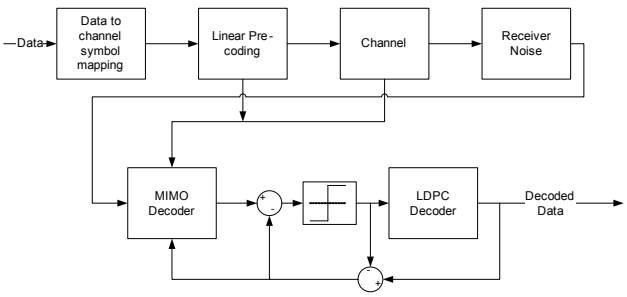

Figure 2: Block diagram of the iterative decoder system.

Figure 2 shows the block diagram of the iterative turbo MIMO decoder. The binary data is mapped to uncoded channel symbols which are then mapped to and layered to LP encoded channel symbols. These encoded channel symbols are transmitted over the Rayleigh faded MIMO channel. At the receiver the channel symbols are received and AWGN is added. The MIMO decoder uses the information about the LP code, Θ', the channel information $\mathbf{H}'$ and the signal constellation to soft decode the data into MAP LLR values.

In the initial iteration there is no extrinsic information provided by the FEC code. Thus the initial input to the FEC decoder is the aposteriori output of the MIMO decoder. On subsequent iterations the intrinsic information is subtracted from the aposteriori information from the FEC decoder to yield the extrinsic information from the FEC decoder which is used by the MIMO decoder as apriori information. Similarly the input to the FEC decoder, on subsequent iterations, is obtained by subtracting the intrinsic information from the aposteriori information of the MIMO decoder.

Hard Limiting of the LLRs: It was found that it was necessary to limit the magnitude of the LLRs provided to the LDPC decoder especially specifically at high SNR values, for performance and stability reasons. In the event that the LLRs were too large, the specific LDPC used decoder would breakdown. Hard limiting of the magnitude of the LLRs was done on the input to the LDPC decoder. The effect of the value of the hard limit threshold on the BER of the system is shown in figures 3 & 4. It is shown that at the specific E_b/N_0 values in the figure that a low threshold of around five (5) is optimal. The two figures also show the diminishing returns that each additional iteration provides.

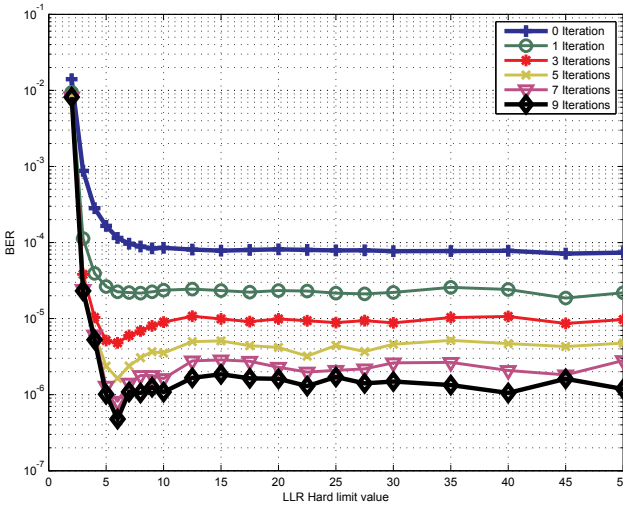

Figure 3: Effect of LLR hard limiting threshold on the BER for QPSK at 11dB E_b/N_0.

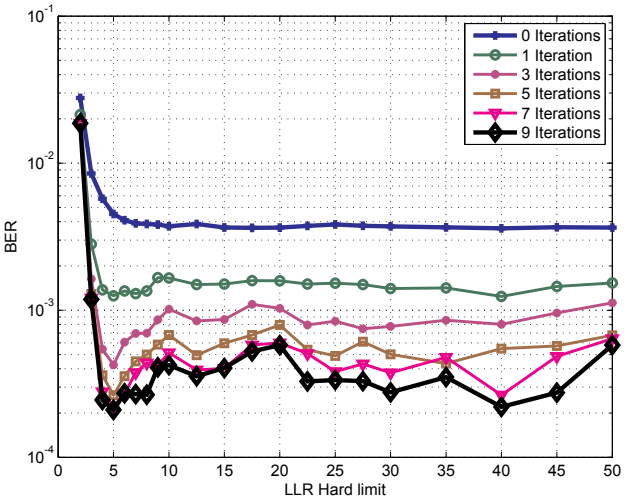

Figure 4: Effect of LLR hard limiting threshold on the BER for 16QAM at 15dB E_b/N_0.

8. RESULTS

Figure 5 shows the effects iterative decoding has on the BER performance of the MIMO system for case of a QPSK signal constellation and a 16QAM signal constellation. It is evident that the BER is reduced by an order of magnitude for both constellations. Due to the steep slope of the BER curve, this only corresponds to a 1 dB improvement as measured at 10^{-4}.

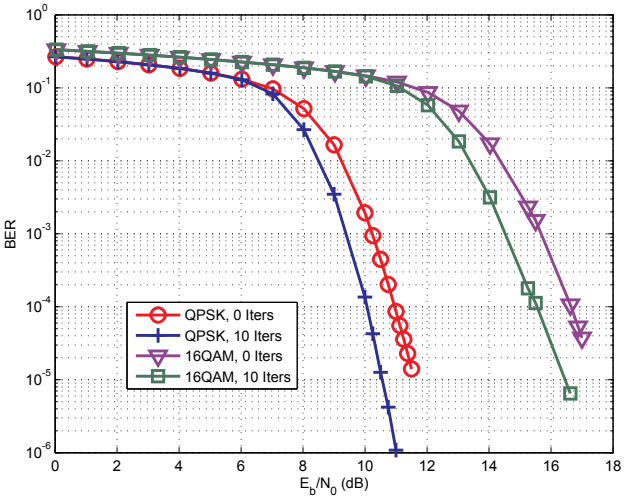

Figure 5: Plot of the BER for QPSK and 16QAM showing the effects of ten decoding iterations on the BER.

The decrease in the number of `SYMLIST()` calls made by the decoder for QPSK and 16QAM signal constellations are shown in figures 6 & 7 respectively. The decrease in the number of function calls is with respect to the unsorted decoder. It is seen in figure 6 that the mSQRD and SQRD perform similarly in the low E_b/N_0 regions, followed by the sub-optimal $\mathbf{H}+\lambda$ ordering method. At higher E_b/N_0 values the mSQRD and $\mathbf{H}+\lambda$ start outperforming the SQRD algorithm. With the mSQRD algorithm a maximum

speed up of 42.5% is achieved for QPSK.

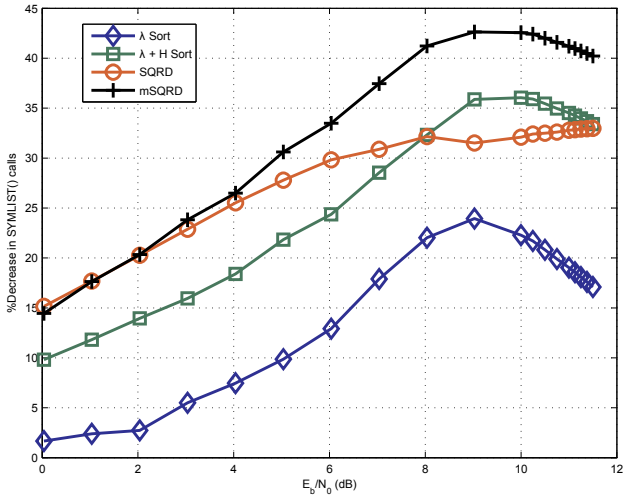

Figure 6: Plot in of the decrease in SYMLIST() calls for each of the sorting methods as compared to the unsorted case. QPSK.

Figure 7 shows that a significant performance increase is obtained from the SQRD based algorithms of approximately 35% at 0dB E_b/N_0 increasing to a maximum of approximately 70%. All of the methods incorporating channel information appear to perform similarly at higher E_b/N_0 values.

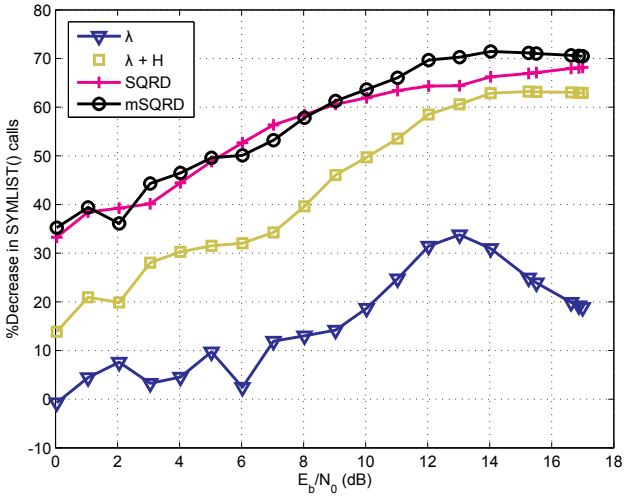

Figure 7: Plot in of the decrease in SYMLIST() calls for each of the sorting methods as compared to the unsorted case. 16QAM.

From figure 8 it can be noted that there is a distinct region where the addition of the apriori information adds to the decoding complexity; however, sorting based on λ does not appear to increase the complexity of subsequent iterations. The added complexity at low E_b/N_0 values is negligeable. At high E_b/N_0 the addition of apriori information substantially reduces the decoding complexity by up to 95%.

In figure 9 the average number of SYMLIST() calls per bit needed to perform hard output decoding is plotted for the

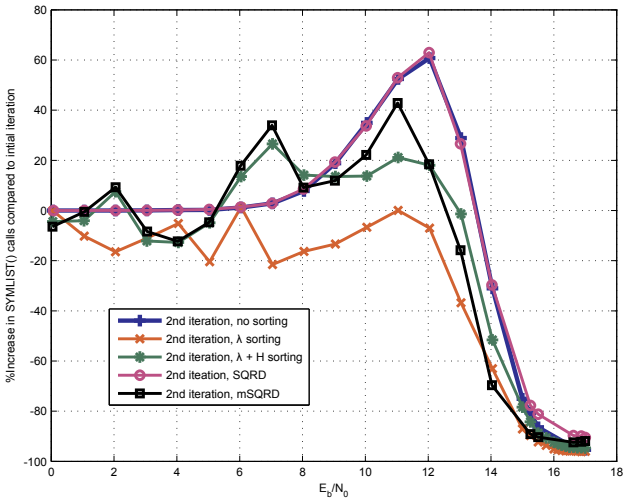

Figure 8: Plot of the increase in SYMLIST() calls due to the apriori pruning metric used in subsequent iterations vs. the normal sphere metric of the first iteration for the various sorting strategies. 16QAM.

unsorted sphere using the classical pruning metric. The figure show a decrease in the number of calls required as E_b/N_0 increases.

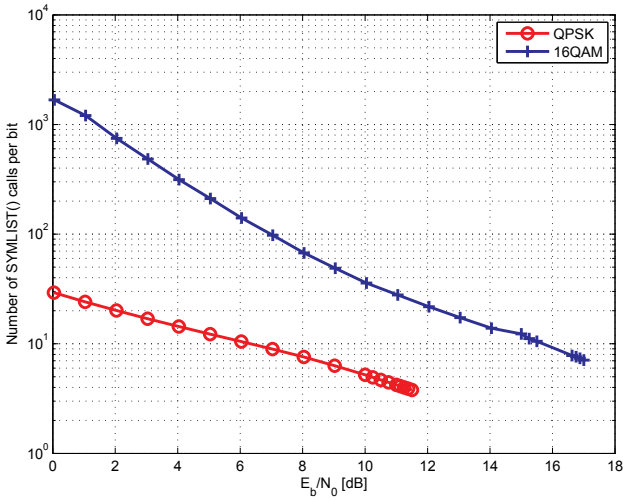

Figure 9: Plot of the number of SYMLIST() calls for the unsorted sphere decoder using the classic pruning metric for QPSK and 16QAM. Hard output decoding.

9. DISCUSSION

Whilst figures 6 & 7 show a definitive improvement at high E_b/N_0 values, it should also be kept in mind that the overall number of SYMLIST() calls at high E_b/N_0 is much less than at lower E_b/N_0 values. As such, it is ultimately expected that at *very* high E_b/N_0 values, the speed difference between the algorithms would converge to almost nothing. The sorting metrics incorporating channel information $\mathbf{H}$ yield an improvement of between 10% to 15% in the low E_b/N_0 regions for QPSK and between 15%

to 35% for 16QAM.

Considering that incorporating apriori information adds approximately, worst case, 60% additional calls, as compared to not using apriori information, but with the mSQRD also resulting in a 60% overall reduction in the number of calls the overall effect is that the apriori sphere decoder is not significantly more complex when used for iterative decoding. Deciding the decoding order based on λ does not appear to increase the decoding complexity at all. This is most likely due to the fact that the MAP outputs of both the MIMO decoder and the FEC decoder are supposed to, ideally, be highly correlated. Should that not be the case then the apriori decoder would likely have a significantly increased complexity.

Figure 9 shows that whilst the sorted algorithms yield a significant decrease in computations, with respect to the unsorted decoder, at high E_b/N_0 values, the decoding complexity is already an order of magnitude less for QPSK and two orders of magnitude for 16QAM.

10. CONCLUSION

In this paper an algorithm sphere decoder that accepts apriori information in LLR form was proposed. A simple iterative MIMO decoder was used to evaluate the algorithm practically. It was shown that sorting the order in which the symbols are decoded can have a great impact on the computational requirements of the algorithm. Based on the good performance of the SQRD algorithm in reducing the complexity of the decoder, a modification of the algorithm to incorporate apriori information was proposed. The mSQRD algorithm was shown to yield the greatest reduction in computational requirements of the decoder. It was also shown that in the case of iterative decoding the addition of apriori information does not significantly increase the computational requirements. The addition of apriori information actually reduced the computational requirements. This is contrary to what was expected since the pruning of the tree is delayed with the addition of apriori information.

REFERENCES

[1] S. Alamouti, "A simple transmit diversity technique for wireless communications," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 8, pp. 1451–1458, October 1998.

[2] H. El Gamal and M. Damen, "Universal space-time coding," *Information Theory, IEEE Transactions on*, vol. 49, no. 5, pp. 1097 – 1119, May 2003.

[3] V. Tarokh, H. Jafarkhani, and A. Calderbank, "Space-time block codes from orthogonal designs," *Information Theory, IEEE Transactions on*, vol. 45, no. 5, pp. 1456 –1467, July 1999.

[4] W. Zhang, X. G. Xia, and P. C. Ching, "High-Rate Full-Diversity Space-Time-Frequency Codes for Broadband MIMO Block-Fading Channels," *IEEE Transactions on Communications*, vol. 55, no. 1, pp. 25–34, 2007.

[5] J. Yoo, J. Lee, and P. Sin-Chong, "Performance evaluation of various MIMO decoders for IEEE 802.11n WLAN system," in *Communication Technology, 2006. ICCT '06. International Conference on*, November 2006, pp. 1 –3.

[6] Z. Safar, W. Su, and K. J. R. Liu, "A fast sphere decoding algorithm for space-frequency block codes," *EURASIP J. Appl. Signal Process.*, vol. 2006, pp. 148–148, January 2006. [Online]. Available: http://dx.doi.org/10.1155/ASP/2006/97676

[7] J. Proakis, *Digital communications*. Boston: McGraw-Hill, 2001.

[8] R. Wang and G. Giannakis, "Approaching MIMO channel capacity with soft detection based on hard sphere decoding," *IEEE Transactions on Communications*, vol. 54, no. 4, pp. 587–590, April 2006.

[9] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: turbo-codes," *Communications, IEEE Transactions on*, vol. 44, no. 10, pp. 1261 –1271, October 1996.

[10] P. Botha and B. Maharaj, "Turbo STFC decoding with the Zero Forcing decoder," in *AFRICON, 2011*, September 2011, pp. 1–5.

[11] T. Seifert, E. P. Adeva, and G. Fettweis, "Towards complexity-reduced soft-input soft-output sphere detection," in *Systems, Communication and Coding (SCC), Proceedings of 2013 9th International ITG Conference on*, January 2013, pp. 1–6.

[12] C. Studer and H. Bolcskei, "Soft-input soft-output single tree-search sphere decoding," *Information Theory, IEEE Transactions on*, vol. 56, no. 10, pp. 4827–4842, October 2010.

[13] D. Wubben, R. Bohnke, J. Rinas, V. Kuhn, and K. Kammeyer, "Efficient algorithm for decoding layered space-time codes," *Electronics Letters*, vol. 37, no. 22, pp. 1348–1350, October 2001.

[14] P. Hart, N. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *Systems Science and Cybernetics, IEEE Transactions on*, vol. 4, no. 2, pp. 100–107, July 1968.

[15] R. Dechter and J. Pearl, "Generalized best-first search strategies and the optimality of a*," *J. ACM*, vol. 32, no. 3, pp. 505–536, July 1985. [Online]. Available: http://doi.acm.org/10.1145/3828.3830

[16] C. Studer, A. Burg, and H. Bolcskei, "Soft-output sphere decoding: algorithms and vlsi implementation," *Selected Areas in Communications, IEEE Journal on*, vol. 26, no. 2, pp. 290 –300, February 2008.