Fast Sphere Decoding of Space-Frequency Block Codes via Nearest Neighbor Signal Point Search

Zoltan Safar[1]**, Weifeng Su and K. J. Ray Liu**

Department of Electrical and Computer Engineering
and Institute for Systems Research
University of Maryland, College Park, MD 20742, USA
e-mail: safar@itu.dk, and {weifeng, kjrliu}@eng.umd.edu

***Abstract*: The recently proposed space-frequency coded MIMO-OFDM systems have promised considerable performance improvement over single-antenna systems, but the complexity of the ML decoding algorithm may hamper the widespread use of such systems. In this paper, we propose a computationally efficient decoding algorithm for space-frequency block codes. The algorithm is based on a general sphere decoding framework formulated in the complex domain. We develop two decoding approaches: a modulation-independent approach, applicable to any memoryless modulation method, and a QAM-specific, fast decoding algorithm performing nearest neighbor signal point search. The simulation results demonstrate that the proposed algorithm can significantly reduce the decoding complexity. For 64 QAM modulation, we observe about 57% reduction in the required FLOP count per code block compared to previously proposed methods without noticeable performance degradation.**

1. Introduction

Multiple-input-multiple-output (MIMO) systems employing multiple transmit and receive antennas have the potential to play a significant role in the development of future wireless communication systems. By taking advantage of the larger number of propagation paths between the transmitter and the receiver, the adverse effects of the channel fading can be significantly reduced. To take advantage of both the MIMO systems and the OFDM modulation, MIMO-OFDM systems have been proposed recently, where space-frequency (SF) coding is applied as the channel code. However, before such a system becomes an attractive choice for practical applications, implementation issues such as decoding complexity must be addressed successfully.

Computationally efficient decoding algorithms have only been proposed for decoding space-time (ST) block codes in quasi-static, flat fading environment [1], [2]. For ST block codes transmitted over temporally evolving channels and for SF block-coded MIMO-OFDM systems, where the channel changes along the frequency axis, low complexity decoding algorithms still do not exist in the literature.

The sphere decoding algorithm was introduced in [3] assuming a single-antenna, real-valued fading channel model. Later results [4] [5], [6] generalized the algorithm to complex valued MIMO systems. In [7], the sphere decoding algorithm was applied to equalize frequency selective MIMO channels. All of these works considered uncoded MIMO systems and assumed quasi-static, flat fading channels. Moreover, they formulated the sphere decoding problem in the real domain, so the algorithms can only be used with modulation methods that can be decomposed into the product of two real constellations (for example, square QAM).

A complex-domain sphere decoding algorithm was described in [8]. This work considered iterative (turbo) decoding in a MIMO system where linear ST mapping was combined with an outer channel code, and a sphere detector was used to approximate the log-likelihood ratio in a computationally efficient way. This approach was specific to modulation methods that can be decomposed into PSK constellations, and its objective was to identify a set of candidate solutions, as opposed to finding the best candidate solution with the maximum possible efficiency.

In this paper, we propose an algorithm for decoding SF block codes using the idea of sphere decoding. The algorithm is based on a general framework formulated in the complex domain, which allows us to fully exploit the distance structure of complex signal constellations. We describe two decoding approaches. The first approach does not assume any apriory information about the used constellation, and it can decode SF block codes with any memoryless modulation. The second approach is QAM-specific, and it takes advantage of the special structure of the QAM constellation by performing nearest neighbor signal point search.

2. System Model and Notation

Consider a SF-coded MIMO-OFDM system having K transmit antennas, L receive antennas and M subcarriers, with M being a multiple of K. Suppose that the frequency selective fading channels between each pair of transmit and receive antennas have P independent delay paths and the same power delay profile. The MIMO channel is assumed to be constant over each OFDM block period. The channel impulse response from transmit antenna k to receive antenna l at time τ is modeled as $h_{k,l}(\tau) = \sum_{p=0}^{P-1} \beta_{k,l}(p)\delta(\tau - \tau_p)$, where τ_p is the delay and $\beta_{k,l}(p)$ is the complex amplitude of the p-th path between transmit antenna k and receive antenna l. The $\beta_{k,l}(p)$'s are zero-mean, complex Gaussian random variables with variances $E[|\beta_{k,l}(p)|^2] = \delta_p^2$. The powers of the P paths are normalized such that $\sum_{p=0}^{P-1} \delta_p^2 = 1$. The frequency response of the channel is given by $H_{k,l}(f) = \sum_{p=0}^{P-1} \beta_{k,l}(p) e^{-j2\pi f \tau_p}$. We assume that the MIMO channel is spatially uncorrelated, i.e. the

[1] Z. Safar is with the Department of Innovation, IT University of Copenhagen, Copenhagen, Denmark.

$\beta_{k,l}(p)$'s are independent for different indices (k,l).

The input bit stream is divided into b bit long segments, creating B-ary ($B = 2^b$) source symbols. The encoder forms M/K source symbol blocks, each containing N source symbols. Source symbol $s_i \in \{0,1,\ldots,B-1\}, i = 0,1,\ldots,N-1$, is mapped onto a complex channel symbol (or constellation point) x_i according to $x_i = \Omega(s_i)$, where the function $\Omega(.)$ represents the modulation operation. The average energy of the constellation will be denoted by E_{avg}.

Then, the SF encoder forms two-dimensional, square codewords from the channel symbols. The SF codeword corresponding to the t-th ($t = 0,1,\ldots,M/K-1$) source symbol block can be expressed as a K by K matrix $\mathbf{C} = \{c_k[Kt+m]\}$ ($k,m = 0,1,..,K-1$), where $c_k[i]$ denotes the channel symbol transmitted over the i-th sub-carrier by transmit antenna k. We assume that each $c_k[i]$ is either zero, or a channel symbol or a negative and/or complex conjugate of a channel symbol corresponding to a source symbol in the appropriate source symbol block.

At the receiver, after matched filtering, removing the cyclic prefix, and applying FFT, the received signal corresponding to the t-th source symbol block at sub-carrier $Kt+m$ ($m = 0,1,\ldots,K-1$) and receive antenna l is given by

$$y_l[Kt+m] = \sum_{k=0}^{K-1} H_{k,l}[Kt+m]c_k[Kt+m]+n_l[Kt+m], \tag{1}$$

where $H_{k,l}[i] = H_{k,l}(i\Delta f)$ is the channel frequency response at the i-th sub-carrier between transmit antenna k and receive antenna l, $\Delta f = 1/T$ is the sub-carrier separation in the frequency domain, and T is the OFDM symbol period. We assume that the channel state information $H_{k,l}[i]$ is known at the receiver, but not at the transmitter. In (1), $n_l[i]$ denotes the complex, zero-mean, additive white Gaussian noise component at the i-th sub-carrier at receive antenna l. The variance of the noise samples is assumed to be $1/(\rho\gamma)$, where the scaling factor γ is defined as $\gamma = b/(K^2 E_{avg})$, so ρ is the signal to noise ratio per bit at each sub-carrier at each receive antenna. In the sequel, we will focus our attention on decoding a single code block, so a simplified notation will be used by dropping the block index t.

3. Equivalent Representation

In general, SF coding introduces spatial and frequency-domain dependence among the code symbols $c_k[i]$ within a code block $\mathbf{C}$. For example, in case of the 2×2 orthogonal design ($K = 2, N = 2$) [1]

$$\mathbf{C} = \begin{bmatrix} x_0 & x_1 \\ -x_1^* & x_0^* \end{bmatrix}, \tag{2}$$

the channel symbols transmitted from different transmit antennas and through different sub-carriers are clearly related. However, to be able to use a sphere decoder (to be able to make sequential decisions on the sent signal coordinates), it is necessary to transform the received signal to an equivalent signal representation, where the coordinates of the sent signal vector are independent. Due to space limitations, we will only consider SF codes constructed from the 2×2 orthogonal design (2) in this paper, but our approach can be extended to more transmit antennas and quasi-orthogonal designs.

The transformed equivalent received signal vector $\mathbf{y}_l = [\, y_l[0], y_l^*[1] \,]^T$ for receive antenna l can be rewritten in matrix-vector form as

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

where $\mathbf{x} = [\, x_0, x_1 \,]^T$ is the $N \times 1$ transmitted channel symbol vector, $\mathbf{n}_l = [\, n_l[0], n_l^*[1] \,]^T$ is the equivalent noise component, and $\mathbf{H}_l$ is defined as

$$\mathbf{H}_l = \begin{bmatrix} H_{0,l}[0] & H_{1,l}[0] \\ H_{1,l}^*[1] & -H_{0,l}^*[1] \end{bmatrix}.$$

By collecting the received signal and noise components corresponding to different receive antennas in $KL \times 1$ vectors as $\mathbf{y} = [\, \mathbf{y}_0^T, \ldots, \mathbf{y}_{L-1}^T \,]^T$, and $\mathbf{n} = [\, \mathbf{n}_0^T, \ldots, \mathbf{n}_{L-1}^T \,]^T$, the equivalent received signal can be expressed as

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n}, \tag{3}$$

where the $KL \times N$ matrix $\mathbf{H}$ is the equivalent channel matrix, defined as $\mathbf{H} = [\mathbf{H}_0^T, ..., \mathbf{H}_{L-1}^T]^T$. Note that the above described equivalent representation has the following properties that are important from the viewpoint of the sphere decoding algorithm. First, the coordinates of the noise vector $\mathbf{n}$ are independent, zero mean, complex Gaussian random variables with variance $1/(\rho\gamma)$. Second, the coordinates of the $\mathbf{x}$ vector are independent. Third, the matrix $\mathbf{H}$ has at least as many rows as columns, independently of the number of receive antennas. Fourth, since the entries in the matrix $\mathbf{H}$ are complex, zero mean, Gaussian random variables and we have assumed that the MIMO channel is spatially independent, the matrix $\mathbf{H}$ has full (column) rank with high probability.

4. The Sphere Decoding Framework

For systems described by (3), to decode the sent signal vector $\mathbf{x}$ with the maximum likelihood (ML) algorithm, the task is to find a valid signal vector $\mathbf{x}$ that minimizes the metric $||\mathbf{y} - \mathbf{H}\mathbf{x}||^2$. Unfortunately, in some cases this can only be performed by exhaustive search over all valid signal vectors. To alleviate this computational burden, sphere decoding was proposed [3], where the decoder searches over only a subset of $\mathbf{x}$ vectors that lie within a hyper-sphere of radius r centered around the received signal vector, i.e. $||\mathbf{y} - \mathbf{H}\mathbf{x}||^2 \leq r^2$.

This section will give a brief overview of the sphere decoding framework described in [10]. The fast decoding algorithm proposed in this paper will be constructed using that framework. In [10], the decoding task was divided into two parts: the preprocessing stage and the searching stage. In this paper, we only describe the proposed algorithm; more detailed discussions on this topic can be found in [3], [5], [7].

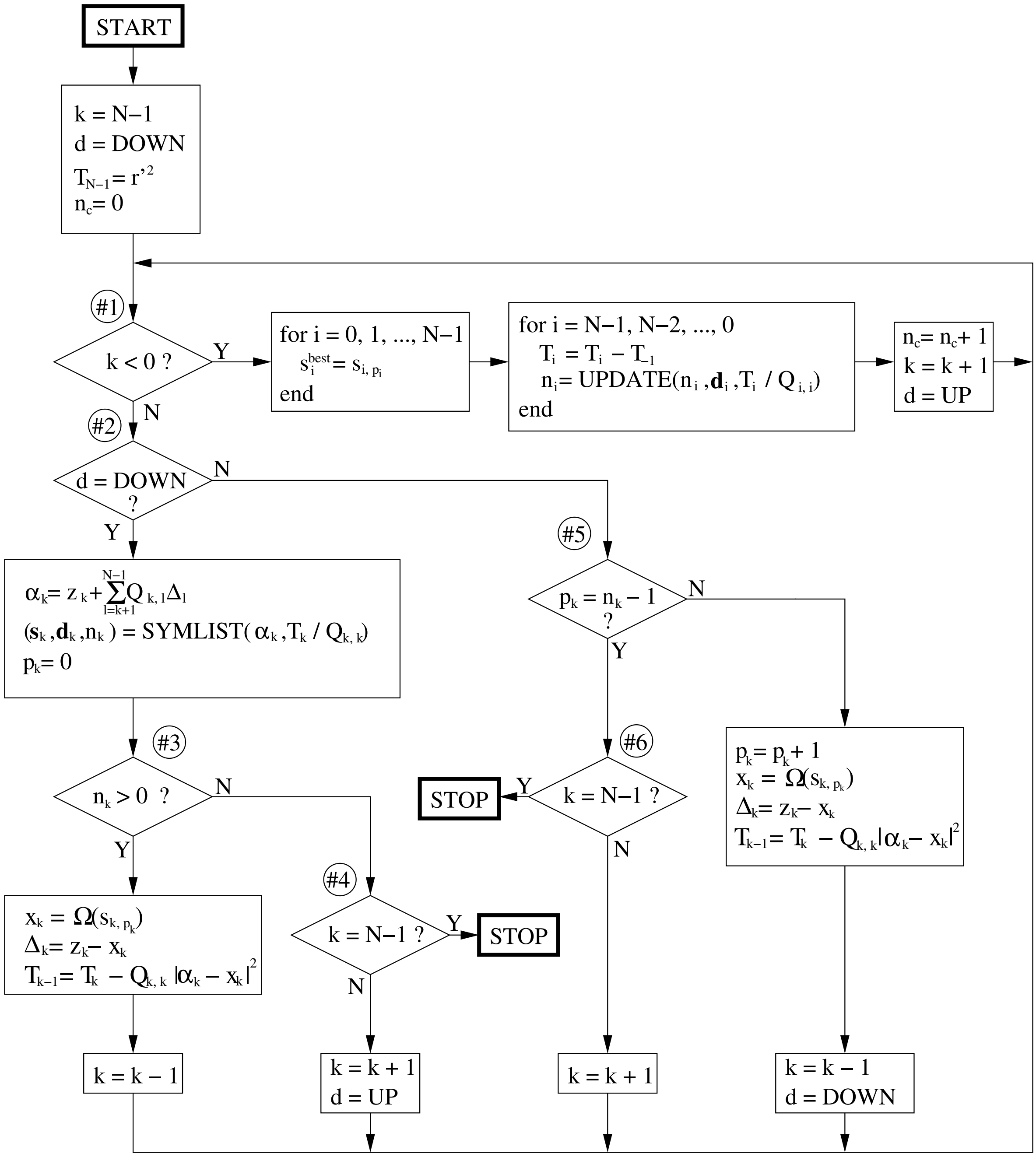

Figure 1: The flowchart of the decoding algorithm

4.1. Preprocessing Stage

The purpose of this stage is to transform the expression $||\mathbf{y} - \mathbf{Hx}||^2$ in such a form that the decisions on the coordinates of **x** can be made sequentially. First, the complex Cholesky factorization of the matrix $\mathbf{H}^{\mathcal{H}}\mathbf{H}$ is calculated, obtaining an $N \times N$ complex, upper triangular matrix $\mathbf{R} = \{R_{k,l}\}$ with positive and real diagonal elements. We used a simple extension of the "Gaxpy" version of the real Cholesky decomposition algorithm [9]. Then, the elements of the $N \times N$ matrix $\mathbf{Q} = \{Q_{k,l}\}$ are calculated as $Q_{k,k} = R_{k,k}^2$ and $Q_{k,l} = R_{k,l}/R_{k,k}$ for $k < l$. After that, the zero forcing solution $\mathbf{z} = \mathbf{H}^{+}\mathbf{y}$ is determined, where $\mathbf{H}^{+}$ denotes the pseudo-inverse of $\mathbf{H}$. This can be done most efficiently by solving the lower triangular system $\mathbf{R}^{\mathcal{H}}\mathbf{w} = \mathbf{H}^{\mathcal{H}}\mathbf{y}$ for **w** and solving the upper triangular system $\mathbf{Rz} = \mathbf{w}$ for **z**. The last step is to obtain the modified radius $r'^2 = r^2 - ||\mathbf{y}||^2 + ||\mathbf{Hz}||^2$. Using these quantities, the sphere decoding problem can be expressed in the following way: find only those **x** signal vectors that satisfy $||\mathbf{R}(\mathbf{z} - \mathbf{x})||^2 \le r'^2$.

4.2. Searching Stage

The searching stage generates the sent signal vectors **x** satisfying the sphere constraint and selects the decoded signal vector. The flowchart of the decoding framework is shown in Figure 1, implementing a greedy, constrained depth-first tree search algorithm. The variables x_k and z_k denote the k-th coordinates of **x** and **z**, respectively, and the quantity T_k can be thought of as the remaining squared distance between the partial solution $x_{N-1}, x_{N-2}, ..., x_{k+1}$ and the surface of the sphere. The variable d indicates whether a node in the search tree is reached from above ("DOWN"), i.e. it is visited for the first time, or from below ("UP"), i.e. it has been visited before. The variable n_c counts the number of valid candidate solutions that have been generated.

At level k ($k = N-1, N-2, ..., 0$), the index k is checked, and if it is non-negative, the bottom level of the tree has nor been reached yet. The function SYMLIST generates the list of possible source symbols $s_{k,i}$, whose corresponding channel symbols $x_{k,i} = \Omega(s_{k,i})$ satisfy the normalized partial constraint $|\alpha_k - x_{k,i}|^2 \leq T_k/Q_{k,k}$. The SYMLIST function takes α_k and the normalized partial constraint $T_k/Q_{k,k}$ as inputs and produces 3 outputs. The first output, n_k ($0 \leq n_k \leq B$), is the number of symbols satisfying the current partial constraint, so there will be n_k branches emanating from the current node. The second output is the symbol list $\mathbf{s}_k = [\, s_{k,0}, s_{k,1}, \ldots, s_{k,n_k-1} \,]^T$, and the vector $\mathbf{d}_k = [\, d_{k,0}, d_{k,1}, \ldots, d_{k,n_k-1} \,]^T$ whose coordinates are the metrics $d_{k,i} = |\alpha_k - x_{k,i}|^2$. The symbols in $\mathbf{s}_k$ are ordered according to increasing $d_{k,i}$ values. The variable p_k is the k-th symbol pointer, pointing to the current element of $\mathbf{s}_k$. If the value of k becomes negative, we have reached the bottom level, so a valid candidate solution **x** has been found. At this point, the radius of the sphere is reduced to further decrease the decoding complexity by adjusting all partial constraint values such that the last solution satisfies the constraints with equality (the "surplus" partial constraint T_{-1} is subtracted from each partial constraint). The last **x** vector is the best solution so far, so the corresponding source symbols are saved in the $\{s_i^{best}\}$ variables by overwriting the previous solution. The source symbol lists are also modified by the UPDATE function, which keeps only those source symbols on the list whose corresponding d_k metric values satisfy the new partial constraints. Since the symbols are ordered according to the corresponding d_k values, this can be done simply by changing the value of n_k at each level.

5. The Proposed Search Methods

The algorithm of Figure 1 is only a general framework that performs a greedy, constrained depth-first tree search. The heart of the decoding algorithm is the function SYMLIST, which creates the ordered list of source symbols satisfying the partial constraint at the given level. The efficiency of the sphere decoding algorithm largely depends on the efficiency of this function. Moreover, by using different implementations of the SYMLIST function, the general framework can be used to perform modulation-independent decoding, and it can also be tailored to the properties of a particular modulation method. In the following subsections, this possibility will be explored further.

The header of the algorithm can be described as (**s**, **d**, n)= SYMLIST(α, T), where the inputs are the value of α and the partial constraint T, and the outputs are n, the number of symbols on the list, the symbol list $\mathbf{s} = [\, s_0, s_1, \ldots, s_{n-1} \,]^T$, and the corresponding metric list $\mathbf{d} = [\, d_0, d_1, \ldots, d_{n-1} \,]^T$.

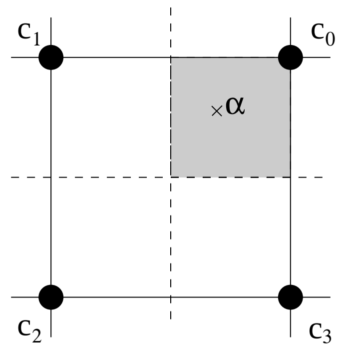

Figure 2: The fast QAM search algorithm

5.1. The Modulation-Independent Search

First, we assume that there is no apriory information available on the used constellation, so we have to develop an algorithm that would work with any memoryless modulation method. In this case, to create the source symbol list, we need to enumerate all possible source symbols, and sort the ones that satisfy the partial constraint, and put them onto the symbol list.

The algorithm goes over all possible source symbols $s = 0, 1, ..., B-1$, and for each source symbol, it calculates the corresponding channel symbol $x = \Omega(s)$ and the metric $d = |\alpha - x|^2$. Then, it checks whether the partial constraint $d \leq T$ is satisfied. If not, the execution continues with the next s value. If the constraint is satisfied, the source symbol s is inserted into the list **s**, the metric d is inserted into the metric list **d**, and the number of items on the list gets incremented. At the end, the metric list and the symbol list are sorted according to the metric values such that the symbol with the smallest metric will be the first on the list.

5.2. The Fast Search for Square QAM Modulation

The previously proposed, most efficient sphere decoding approach [5] enumerates all signal points x satisfying the partial constraint at all tree levels, and then sorts them. However, it was observed that most of the time, the first candidate solution **x** found by the greedy tree search (i.e. the greedy solution) is actually the ML solution, so only one leaf node is explored during the decoding process with high probability. This means that enumerating and sorting all signal points that satisfy the partial constraints is redundant because most source symbols on the symbol lists get eliminated via radius reduction and the subtrees corresponding to their values will never be explored.

To further improve the computational efficiency of the sphere decoding algorithm, we propose a fast search approach by making the most probable case more efficient. The proposed fast search algorithm performs a nearest neighbor search. The symbol list generation algorithm SYMLIST enumerates only a few nearest signal points x to α and ignores the signal points that are further away, so the unnecessary enumeration and sorting operations can be avoided.

If the used constellation is B-ary square QAM, the source symbol s can be expressed as the concatenation of its real and imaginary parts: $s = (s^R, s^I)$, $s^R, s^I \in \{0, 1, \ldots, \sqrt{B}-1\}$, and the complex channel symbol corresponding to this source symbol is

$$x = \Omega(s) = \left(s^R - \frac{\sqrt{B}-1}{2} \right) + j \left(s^I - \frac{\sqrt{B}-1}{2} \right).$$

The average energy of the constellation is $E_{avg} = \frac{B-1}{6}$. For the simplicity of the explanation, we do not consider Gray bit-mapping and neglect the edge effects, assuming that the point α falls in a region where it has four neighboring signal points, as shown in Figure 2. The basic idea of the fast search method is to identify these four constellation points efficiently by exploiting the geometrical properties of the QAM constellation. The first step is to express the value of α in the coordinate system of the real and imaginary parts of the source symbols as

$$s_\alpha^R = \text{Re}\{\alpha\} + \frac{\sqrt{B}-1}{2}, \quad s_\alpha^I = \text{Im}\{\alpha\} + \frac{\sqrt{B}-1}{2}.$$

Then, the source symbols corresponding to the four neighboring constellation points can be identified easily by rounding the values of s_α^R and s_α^I up or down. For example, in Figure 2, the source symbol $c_0 = (c_0^R, c_0^I)$ corresponding to the upper right signal point can be obtained as $c_0^R = \lceil s_\alpha^R \rceil$, and $c_0^I = \lceil s_\alpha^I \rceil$. After determining the 4 source symbols $c_i = (c_i^R, c_i^I)$, $i = 0, 1, 2, 3$, that correspond to the 4 nearest neighbors of α, we need to determine the order in which they should be put on the list. The space between the neighboring QAM constellation points is divided into 4 quadrants, as shown by the dashed lines in Figure 2, and the fractional parts of s_α^R and s_α^I are calculated to determine which quadrant the value s_α falls in by comparing the fractional parts to 0.5. Once the index of the quadrant is known, the source symbol corresponding to the signal point closest to α (c_0 in Figure 2) can be easily determined. The signal point furthest away from α (c_2 in Figure 2) can also be identified. Based on this approach, the order between the other two source symbols (c_1 and c_3 in Figure 2) cannot be decided, but our simulations have shown that the actual order of the symbols is not important as long as the source symbol corresponding to the closest signal point is the first on the list, so we set arbitrary (for example, random) order for those two signal points.

Finally, for each of the 4 source symbols, the algorithm calculates the corresponding channel symbol and checks whether the partial constraint is satisfied. If so, the symbol and the corresponding metric value are put on the lists. Note that the symbols on the list will not be perfectly ordered some of the time. However, the algorithm ensures that the source symbol corresponding to the closest constellation point to α will always be the first. As can be seen, the algorithm avoids the enumeration of all channel symbols satisfying the partial constraint and avoids the sorting operation altogether.

6. Simulation Results

To illustrate the performance of the proposed sphere decoding algorithm, we provide some simulation results. The simulated communication system had 2 transmit antennas ($K = 2$), 2 receive antennas ($L = 2$), and the 2×2 orthogonal design (2) with 16 and 64 QAM modulations was used as the SF code ($N = 2$). The OFDM modulation had 128 sub-carriers with an OFDM symbol period of 128 μs. The frequency selective MIMO channel was modeled by the COST 207 Typical Urban 6-ray power delay profile. The initial radius was set to $r = 10$ to ensure ML performance.

We compared the complexity of the searching stages of four different decoding algorithms: the ML decoding algorithm (performing exhaustive search), the sphere decoding algorithm described in [5], and the proposed decoding algorithm with the modulation-independent and the fast QAM-specific nearest neighbor searching symbol list generation algorithms. The complexity comparison of the preprocessing stages can be found in [10].

Figure 3 depicts the bit error rate (BER) curve with 16 QAM as the function of the average SNR. It can be observed that all sphere decoding algorithms have the same performance as the ML decoding algorithm in the SNR range of interest. The average number of real floating point operations (FLOPs) per code block for the searching stages in the 16 QAM case is shown in Figure 4. The FLOP count for the ML decoding algorithm was 20736. Based on the figures, we can make several observations. First, using a sphere decoder, the computational complexity of the SF decoder can be reduced by orders of magnitude without perceptible performance loss in the meaningful BER range. Second, the complexity of the modulation-independent algorithm is higher than the QAM-specific approaches, which was expected. However, we would like to point out that even without any apriory knowledge about the used constellation, the decoding complexity was significantly reduced compared to the ML decoding algorithm. Third, by using the proposed sphere decoding framework and the fast QAM-specific symbol list generation algorithm, considerable complexity reduction can be achieved. The number of FLOPs of the searching stage was reduced to about 58% of the FLOP count of the algorithm described in [5].

The BER performance and the decoding complexity of the 64 QAM case are shown in Figures 5 and 6, respectively. The FLOP count for the ML decoding algorithm was 331776. The tendencies observed here are similar to the 16 QAM case. The main difference is that the complexity of the QAM-specific fast nearest neighbor searching algorithm essentially stayed the same, while the complexity of the ML decoder, the algorithm of [5] and the modulation-independent algorithm has increased considerably. This is intuitively expected since the number of nearest neighbor signal points of an arbitrary point in the QAM constellation does not increase as the size of the constellation increases. As a consequence, the achieved complexity reduction of the proposed fast search algorithm is more significant than in the 16 QAM case: the average FLOP count was reduced to about 43% of that of the algorithm in [5].

7. Conclusion

We proposed a sphere decoding algorithm for decoding SF block codes. The algorithm was based on a decoding framework performing a greedy, constrained

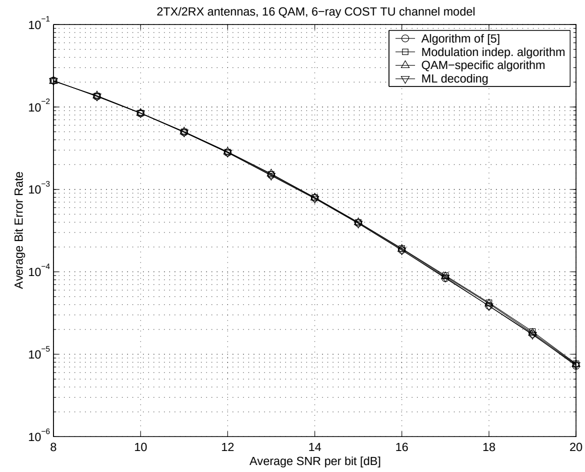

Figure 3: Average bit error rate, 16QAM

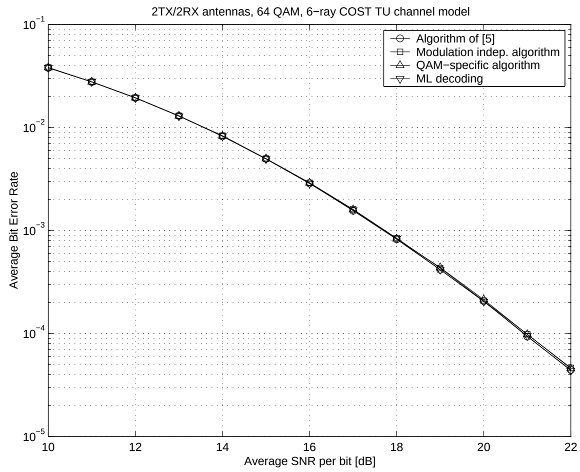

Figure 5: Average bit error rate, 64QAM

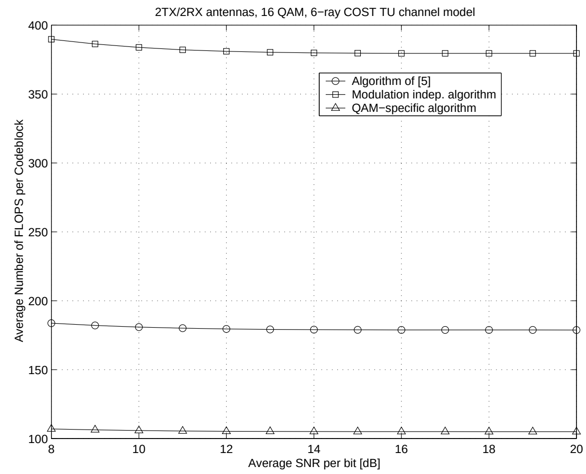

Figure 4: Average number of FLOPS, 16QAM

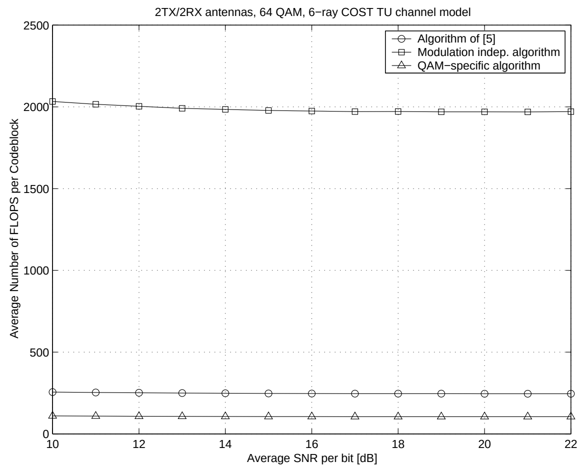

Figure 6: Average number of FLOPS, 64QAM

depth-first tree search. We developed two decoding approaches: the modulation-independent algorithm that does not utilize any information about the used modulation method, and the fast QAM-specific fast algorithm that takes advantage of the special structure of the QAM constellation and limits the search to the nearest neighbor signal points. The simulation results demonstrated significant complexity reduction compared to previously existing algorithms without any performance loss. For problems of larger size (larger K, L and B values), even more pronounced complexity reduction is expected.

REFERENCES

[1] S. Alamouti, "A simple transmit diversity technique for wireless communications", *IEEE JSAC*, vol. 16, no. 8, pp. 1451-1458, 1998.

[2] V. Tarokh, H. Jafarkhani, and A. R. Calderbank, "Space-time block codes from orthogonal designs", *IEEE Trans. Inform. Theory*, vol. 45, no. 5, pp. 1456-1467, 1999.

[3] E. Viterbo and J. Boutros, "A universal lattice code decoder for fading channels", *IEEE Transactions on Information Theory*, vol. 45, no. 5, pp. 1639-1642, 1997.

[4] M. Damen, A. Chkeif and J. Belfiore, "Lattice code decoder for space-time codes", *IEEE Communication Letters*, vol. 4, no. 5, pp. 161-163, 2000.

[5] A. Chan and I. Lee, "A new reduced complexity sphere decoder for multiple antenna systems", *Proc. of IEEE ICC*, vol. 1, pp. 460-464, 2002.

[6] M. Damen, K. Abed-Meraim and J. Belfiore, "Generalized lattice decoder for asymmetrical space-time communication architecture", *Proc. of IEEE ICASSP*, vol. 5, pp. 2581-2584, 2000.

[7] H. Vikalo and B. Hassibi, "Maximum likelihood sequence detection of multiple antenna systems over dispersive channels via sphere decoding", *EURASIP Journal on Applied Signal Processing*, vol. 2002, no. 5, pp. 525-531, 2002.

[8] B. Hochwald and S. ten Brink, "Achieving near-capacity on a multiple-antenna channel", *IEEE Transactions on Communications*, vol. 51, no. 3, pp. 389-399, March 2003.

[9] G. Golub and C. Van Loan, *Matrix computations*, Johns Hopkins University Press, 1996.

[10] Z. Safar, "Efficient Coding and Decoding Methods for Multi-Antenna Wireless Communication Systems", *Ph.D. Dissertation*, University of Maryland, College Park, MD, USA, 2003.