A Lattice-Reduction-Aided Soft Detector for Multiple-Input Multiple-Output Channels

David L. Milliner and John R. Barry

School of ECE, Georgia Institute of Technology
Atlanta, GA 30332-0250 USA, {dlm, barry}@ece.gatech.edu

***Abstract* — Lattice basis reduction is a powerful technique that enables a hard-output detector for a multiple-input multiple-output channel to approach maximum-likelihood performance with low complexity. In this work we propose a soft-output detector that combines lattice-reduction-aided detection and list decoding. The proposed algorithm performs nearly as well as the list-sphere detector but with much lower complexity. Numerical results reveal that the complexity per bit for the proposed algorithm *decreases* as the size of the QAM alphabet increases.**

***Index Terms* — Multiple-input multiple-output (MIMO) systems, lattice reduction, soft information, list decoding.**

I. INTRODUCTION

Lattice-reduction (LR) aided detection has recently emerged as a low-complexity strategy for performing hard-output detection for multiple-input multiple-output (MIMO) channels with quadrature-amplitude modulation (QAM) inputs [1]. The basic idea behind LR-aided detection is to perform detection using a *reduced* lattice basis instead of the original lattice basis thereby realizing decision regions much closer to those of the maximum-likelihood (ML) detector. With one recent exception [2], LR-aided detectors to date have been exclusively hard-output detectors [1], [3]-[6]. In this work we propose a soft-output detector based on lattice reduction.

One well-studied method for generating soft information from a MIMO channel is the list sphere detector (LSD) [7], which generates a list from which it computes bit posterior probabilities. Other methods for computing soft information include breadth-first algorithms [8], semi-definite relaxation [9], iterative tree search [10], space-time Chase decoding [11], modified sphere decoding and Monte Carlo methods [12]. The notion that LR-aided detectors might be used for estimating bit probabilities was mentioned briefly in [3], but no details were presented. In contrast to previous methods, we propose to use lattice reduction to help generate the list. The proposed detector is called the *LR-aided list detector (LRLD)*. An alternative approach designed only for 4-QAM was developed independently and recently reported in [2].

The remainder of this paper is organized as follows. Section II presents our system model. Section III reviews LR-aided hard detection. Section IV presents the proposed soft-output detector. Section V presents a complexity analysis, Section VI presents performance results, and Section VII concludes the paper.

II. SYSTEM MODEL

Separating the real and imaginary parts of an N_T-input N_R-output complex baseband channel leads to an equivalent real channel with $2N_T$ real inputs $\boldsymbol{a} = [a_1, \dots a_{2N_T}]^T$ and $2N_R$ real outputs $\boldsymbol{r} = [r_1, \dots a_{2N_R}]^T$:

$$\boldsymbol{r} = \mathbf{H}\boldsymbol{a} + \boldsymbol{w}\ , \tag{1}$$

where $\mathbf{H} = [\boldsymbol{h}_1, \dots \boldsymbol{h}_{2N_T}]$ is a $2N_R \times 2N_T$ channel matrix whose i-th column is $\boldsymbol{h}_i$, and where $\boldsymbol{w}$ is noise. We assume that the columns of $\mathbf{H}$ are linearly independent, which implies that $N_R \geq N_T$. We assume the noise components are independent identically distributed (i.i.d.) $\mathcal{N}(0, N_0/2)$, i.e., zero-mean Gaussian random variables with variance $N_0/2$. We assume that the channel inputs are chosen uniformly and independently from the pulse-amplitude modulation (PAM) alphabet $\mathcal{A} = \{\pm\alpha, \pm 3\alpha, \dots \pm(M-1)\alpha\}$, where $|\mathcal{A}| = M$ and where M^2 is the size of the equivalent complex QAM alphabet. The constant α is related to the complex alphabet energy E by $\alpha = \sqrt{3E/(2(M^2-1))}$.

The PAM alphabet $\mathcal{A}$ is a scaled and translated subset of the integers. Before proceeding it will be convenient to translate (1) into an equivalent channel whose inputs are actually integers. In particular, adding the constant vector $((M-1)/2)\sum_i \boldsymbol{h}_i$ to $\boldsymbol{r}/(2\alpha)$ yields the following equivalent channel model:

$$\boldsymbol{r}' = \mathbf{H}\boldsymbol{b} + \boldsymbol{w}' \tag{2}$$

where the components of $\boldsymbol{w}'$ are i.i.d. $\mathcal{N}(0, \sigma^2)$ with variance $\sigma^2 = (M^2-1)/(12E/N_0)$, and

$$\boldsymbol{a} = \alpha(1 - M + 2\boldsymbol{b}). \tag{3}$$

This research was supported in part by National Science Foundation grants 0431031 and 0121565.

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2006 proceedings.

The advantage of this transformation is that the elements of $\boldsymbol{b}$ are now drawn from an alphabet $\mathcal{B} = \{0, 1, \ldots M-1\}$ that is a subset of the integers. The scaling and translation that transforms (1) into (2) is reversible, and hence without loss of generality we will assume in the remainder of the paper that the receiver operates on the effective model (2) with integer inputs.

III. Hard Detection using Lattice Reduction

The *lattice* spanned by $\mathbf{H}$ is the set of all linear combinations of its columns with integer coefficients. Because the components of $\boldsymbol{b}$ are integers, the noiseless channel output $\mathbf{H}\boldsymbol{b} = \sum_i b_i \boldsymbol{h}_i$ is a point in the lattice. Furthermore, because the components of $\boldsymbol{b}$ are restricted to $\mathcal{B}$, there are only M^{2N_T} possible values for $\mathbf{H}\boldsymbol{b}$. The problem of hard-output maximum-likelihood detection can thus be seen as finding which of these lattice points is closest to $\boldsymbol{r}'$.

The basis of a lattice is not unique. If $\mathbf{H}$ is a basis, the product $\tilde{\mathbf{H}} = \mathbf{HT}$ will also be a basis whenever the matrix $\mathbf{T}$ is *unimodular* (having integer entries and a determinant of ± 1) [13]. Note that the inverse of a unimodular matrix is also unimodular.

Roughly speaking, lattice *reduction* is the process of transforming an original basis $\mathbf{H}$ into a new (reduced) basis $\tilde{\mathbf{H}} = \mathbf{HT}$ that is as "orthogonal" as possible. A celebrated yet suboptimal polynomial-time lattice reduction algorithm was proposed by Lenstra, Lenstra, and Lovász (LLL) [14]. A concise description of this algorithm is provided in [5].

In terms of a reduced basis $\tilde{\mathbf{H}} = \mathbf{HT}$, the channel model (2) can be rewritten as:

$$\boldsymbol{r}' = \tilde{\mathbf{H}}\mathbf{c} + \boldsymbol{w}' \tag{4}$$

where we have introduced $\mathbf{c} = \mathbf{T}^{-1}\boldsymbol{b}$. *We may thus interpret* $\boldsymbol{r}'$ *as the noisy output of an equivalent channel whose input is* $\mathbf{c}$ *and whose channel matrix is* $\tilde{\mathbf{H}}$. Furthermore, because $\boldsymbol{b}$ and $\mathbf{T}^{-1}$ are both comprised of integers, the effective input $\mathbf{c}$ is also comprised of integers, $\mathbf{c} \in \mathbb{Z}^{2N_T}$.

An LR-aided detector operates in three steps [1], [3]-[6]: first, it performs lattice reduction to find a reduced basis $\tilde{\mathbf{H}} = \mathbf{HT}$; second, it adopts the viewpoint of (4) and uses a low-complexity MIMO detector (such as a linear or decision-feedback detector) to recover $\mathbf{c}$; and third, it recovers the inputs to the original channel using the relationship $\boldsymbol{b} = \mathbf{T}\mathbf{c}$.

The simplest LR-aided detector applies the Moore-Penrose pseudoinverse $\tilde{\mathbf{H}}^+$ of $\tilde{\mathbf{H}}$ to $\boldsymbol{r}'$, and rounds the result to estimate $\mathbf{c}$. Better performance is achieved by the LR-aided minimum-mean square error (LR-MMSE) detector [5], which is based on a reduction of an extended channel matrix:

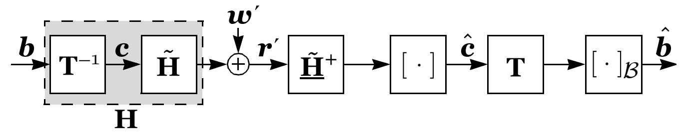

Fig. 1. The MMSE LR-aided MIMO hard-output detector.

$$\underline{\tilde{\mathbf{H}}} = \begin{bmatrix}\mathbf{H}\\ \sigma\mathbf{I}\end{bmatrix}\mathbf{T}. \tag{5}$$

The receiver then estimates $\mathbf{c}$ using

$$\hat{\mathbf{c}}_{\text{LR-MMSE}} = [\underline{\tilde{\mathbf{H}}}^+ \underline{\boldsymbol{r}}'], \tag{6}$$

where $\underline{\boldsymbol{r}}' = [\boldsymbol{r}^T, \mathbf{0}]^T$ and where $[\,\cdot\,]$ denotes a component-wise rounding to the integers. The final decision vector is $\hat{\boldsymbol{b}} = [\mathbf{T}\hat{\mathbf{c}}]_{\mathcal{B}}$, where $[\,\cdot\,]_{\mathcal{B}}$ denotes a component-wise rounding to the nearest element of $\mathcal{B}$. In fact, because the arguments to $[\,\cdot\,]_{\mathcal{B}}$ are already integers, we may interpret this as a hard-limiting operation, mapping all negative integers to zero, and mapping integers greater than $M-1$ to $M-1$. This border limiting approach is equivalent to the one used in [3]. A block diagram of the LR-MMSE detector is shown in Fig. 1.

IV. LR-aided List Detector

The previous section described a hard-output detector, whose goal is to find the lattice point $\mathbf{H}\boldsymbol{b}$ that is closest to $\boldsymbol{r}'$. The goal of a *list* detector is to find an unordered *list* of the l *closest* lattice points $\mathbf{H}\boldsymbol{b}$ to $\boldsymbol{r}'$. We define the *candidate list* $\mathcal{L} \subseteq \mathcal{B}^{2N_T}$ as the set of l corresponding input vectors $\boldsymbol{b}$. Once found, the candidate list can be used to approximate the *a posteriori* probabilities (APPs). The approximation becomes exact when the candidate list encompasses all possible input vectors, $\mathcal{L} = \mathcal{B}^{2N_T}$. In this section we propose a strategy for generating a candidate list that builds on the LR-aided detector.

Conceptually our approach for computing per-bit likelihoods can be broken into three key steps. First, using a LR-aided hard detector we produce an initial decision vector. Finding this decision vector was the topic of Section III. Second, we generate a candidate list of decision vectors, using the initial decision vector as a seed. Lastly, using the candidate list we compute the per-bit likelihoods or log-likelihood ratios (LLRs) of the transmitted vector using standard LLR calculations.

We now describe how to generate the list of candidate vectors. The basic idea is to exploit the fact that all possible estimates for $\mathbf{c}$ lie on the integer lattice $\mathbb{Z}^{2N_T}$. Specifically we propose to enumerate the elements of $\mathbb{Z}^{2N_T}$ within a

1-4244-0357-X/06/$20.00 ©2006 IEEE
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2006 proceedings.

List Generation Algorithm

Input: $\hat{\mathbf{c}}$, $\mathbf{T}$, l_{des}, $\mathbf{D}$.   Output: $\mathcal{L}$

(1) Initialize $i = 0$ and $\mathcal{L} = \varnothing$.

(2) While $|\mathcal{L}| < l_{\text{des}}$,

(3) $\quad \hat{\mathbf{c}}_i = \hat{\mathbf{c}} + \boldsymbol{d}_i$

(4) $\quad \hat{\boldsymbol{b}}_i = [\mathbf{T}\hat{\mathbf{c}}_i]_{\mathcal{B}}$

(5) $\quad \mathcal{L} = \mathcal{L} \cup \hat{\boldsymbol{b}}_i$

(6) $\quad i = i + 1$

(7) end

Fig. 2. The proposed list-generation algorithm.

hypersphere centered at $\hat{\mathbf{c}}$. The search radius is initialized to zero and is increased until the desired list length l_{des} has been achieved. The details are described below.

Let us define a perturbation matrix $\mathbf{D}$ via a one-time offline precomputation whose columns represent vectors that perturb the hard decision $\hat{\mathbf{c}}$. The columns of $\mathbf{D}$ are ordered in nondecreasing Euclidean norm. In theory the number of columns in $\mathbf{D}$ would be infinite, but in practice the size of $\mathbf{D}$ can be fixed based on design requirements. An example of $\mathbf{D}$ for a 2×2 system would be

$$\mathbf{D} = \begin{bmatrix} 0 & 1 & 0 & -1 & 0 & 1 & \cdots \\ 0 & 0 & 1 & 0 & -1 & 1 & \cdots \end{bmatrix} = [\boldsymbol{d}_0, \boldsymbol{d}_1, \boldsymbol{d}_2, \ldots]. \tag{7}$$

Fig. 2 presents the pseudocode for a serial implementation of the proposed list generation algorithm. The column index counter i is initialized to zero. The algorithm proceeds by sequentially (or in parallel) adding the column vectors of $\mathbf{D}$ to $\hat{\mathbf{c}}$. Each $\hat{\mathbf{c}}_i$ that results is transformed to a candidate input vector using $\hat{\boldsymbol{b}}_i = [\mathbf{T}\hat{\mathbf{c}}_i]_{\mathcal{B}}$. The list is populated via the union of $\hat{\boldsymbol{b}}_i$ and the current list $\mathcal{L}$, which prohibits repeated entries. The algorithm terminates as soon as the desired list length l_{des} is achieved. Note that we have made the design decision to force the product $\mathbf{T}\hat{\mathbf{c}}_i$ to be an element of $\mathcal{B}^{2N_T}$. This choice helps to generate the list quickly. A slower alternative would be to discard those $\mathbf{T}\hat{\mathbf{c}}_i$ that are not in $\mathcal{B}^{2N_T}$, which might lead to a more accurate list.

After we obtain a candidate list we can compute the bit probabilities using LLR calculations equivalent to those of (12) in [7] after replacing $\boldsymbol{y}$ with $\boldsymbol{r}'$ and $\boldsymbol{s}$ with $\hat{\boldsymbol{b}}$.

V. Complexity Analysis

The complexity of the LRLD and LSD were analyzed by counting the number of basic operations for each algorithm. We account for the preprocessor complexity of both algorithms, performed once for each realization of the channel, as well as for the core processing performed each signaling interval. The preprocessor complexities are primarily governed by the QR decomposition in the LSD and the LLL algorithm in the LRLD. The core processing for both approaches consists of list generation, LLR computation and, for the LRLD, a LR-MMSE hard detection step.

One distinguishing feature of the LRLD is that the majority of its operations are integer computations as opposed to floating-point operations. Almost always this leads to a savings in terms of both speed and integrated circuit area. For our numerical results we adopt the speedup associated with integer operations versus floating-point operations on a Texas Instruments TMS6700 digital signal processor, where single-precision additions require four times as many clock cycles as integer additions, and single-precision multiplies require four times as many clock cycles as integer multiplies. Therefore, when tallying operations, we divide the number of integer operations by 4 before adding to the number of floating-point operations. While this factor of 4 is not fixed across architectures, and even the overhead costs on the architecture we selected can be mitigated to some extent with proper scheduling, it provides a rough cost estimate when comparing floating-point and integer operations.

The complexity of the proposed detector is a random variable that depends on the channel input, channel matrix, and noise realizations. The complexity is random because the list-generation algorithm can terminate quickly or not, depending on how many valid $\mathbf{c}$ vectors are near the seed $\hat{\mathbf{c}}$. The probability density function of the complexity (operations per bit) is approximated in Fig. 3 by a histogram for (a) 16-QAM and (b) 64-QAM alphabets. These results are for a 4×4 system with an SNR of 10 dB and a list length of $l_{\text{des}} = 256$. The histograms were found by counting operations for more than 115,000 signaling intervals comprising over 4700 distinct realizations of i.i.d. Rayleigh fading channels.

The histograms show that the LRLD is significantly less complex than the LSD. On average, for 16-QAM, the number of LRLD operations is 8.7 times less than the LSD. The disparity is even more dramatic with 64-QAM, where the average number of LRLD operations is 77.2 times less than for LSD. Admittedly, the average complexity is not always a relevant benchmark. The maximum complexity may be more

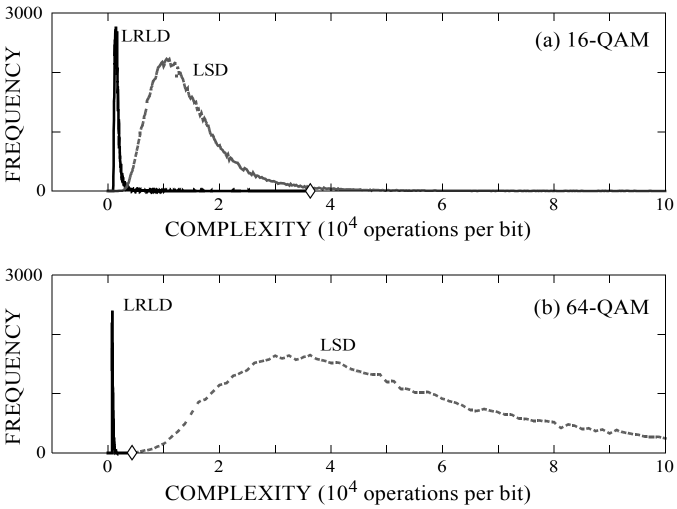

Fig. 3. Complexity histograms for (a) 16-QAM and (b) 64-QAM. The diamonds mark the maximum complexities of the LRLD. The maximum complexities of the LSD are off the scale and not shown.

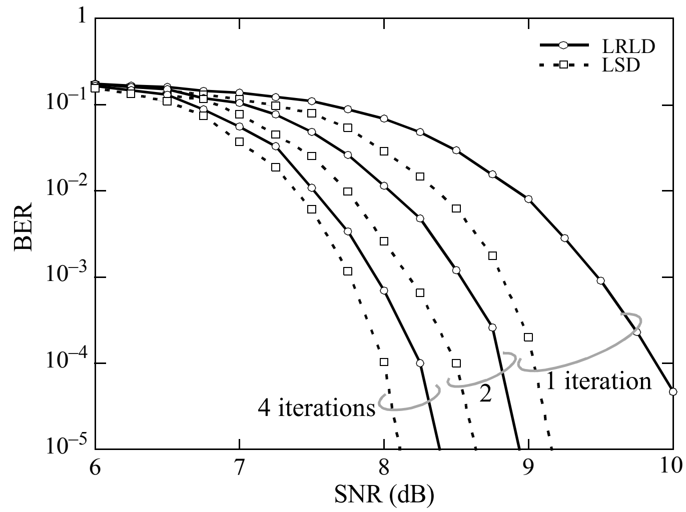

Fig. 4. BER performance for the LRLD and LSD for 1, 2, and 4 outer iterations, for a 4×4 system with 16-QAM.

relevant in practice. For 16-QAM, the maximum number of observed LRLD operations during the $115{,}000$ signaling intervals simulated is 3.1 times less complex than the LSD. The maximum number of observed operations is over 1000 times less complex for 64-QAM.

Comparing Fig. 3(a) to Fig. 3(b), we observe the noteworthy trend that the complexity per bit for the LRLD *decreases* significantly with a growth in QAM size. In contrast, the complexity of the LSD increases from (a) to (b). A possible explanation for the reduction in complexity of the LRLD is that its list-generation algorithm has an easier time generating a candidate list when the elements of $\mathcal{B}^{2N_T}$ are more densely packed on the integer lattice.

VI. Performance Results

We now present the simulated error performance of the LRLD in an iterative space-time bit-interleaved coded modulation (ST-BICM) system and compare it to that of the LSD. More details on this standard system configuration can be found in [7]. Additionally, we constrain the *a priori* information from the outer decoder as in [15] but clip the soft-output of the inner MIMO detector to ±8. Constraining of *a priori* information has been shown to improve the performance of the LSD and the same is true for the LRLD. Finally, the LLR of the MIMO detector was computed as in (12) of [7] using the max-log approximation to simplify calculations.

We assume a 4-input 4-output channel with 16-QAM inputs. As in [7], message bits are encoded by a rate-$1/2$ parallel turbo code with length 18432 and parity generator $(1+D^2)/(1+D+D^2)$ and a length-9216 S-random interleaver with spread parameter 10. Coded bits are interleaved by a S-random interleaver of size 18432 and spread 20 before being Gray mapped to 16-QAM symbols, so that each codeword frame spans 1152 signaling intervals. We assume a quasistatic fading environment with a coherence time of 48 signaling intervals, so that each codeword spans 24 coherence times. For each frame we performed four outer iterations between the detector and decoder; for each outer iteration we performed eight iterations within the turbo decoder.

In Fig. 4 we compare the bit-error rate performance of the LSD and the LRLD after one, two and four outer iterations with $l_{\text{des}} = 512$ for both algorithms. We see that, after four outer iterations, the performance of the LRLD is within 0.3 dB of the LSD.

Fig. 5 illustrates and compares the performance-versus-complexity trade-off of the LRLD and the LSD. The performance is measured by the SNR required to achieve $BER = 10^{-5}$ after four outer iterations, while the complexity is measured by the required number of operations per bit. There are 4 curves in the figure, representing the average and maximum complexity for each algorithm. Each curve is parameterized by the list length. As in Section V, simulations were run over more than $115{,}000$ signaling intervals comprising more than 4700 distinct i.i.d. Rayleigh fading channels, but for each system configuration the SNR used

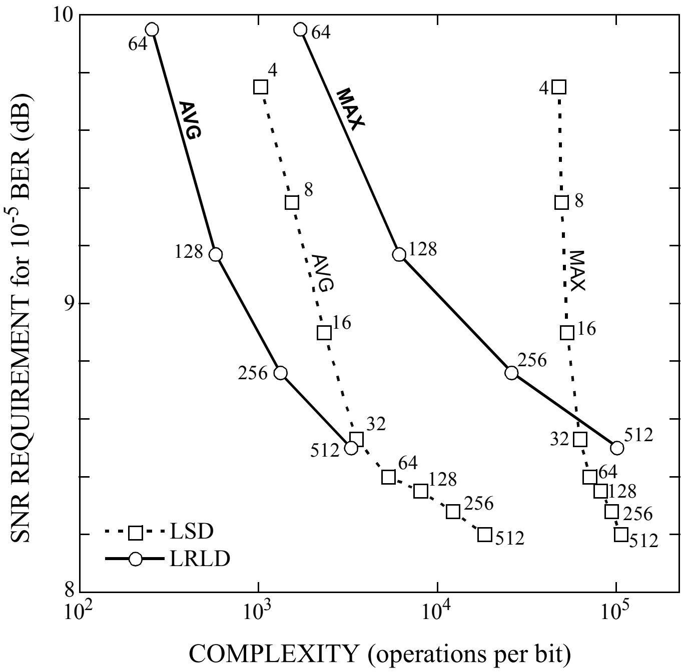

Fig. 5. Performance versus complexity for LRLD and LSD, assuming 4×4 16-QAM.

corresponds to a bit-error rate of 10^{-5} for the system configuration under test. Slicing Fig. 5 horizontally (fixing performance), we note that the LRLD is significantly less complex than the LSD in terms of average and maximum complexity when $l_{\text{des}} \leq 256$. Similarly, slicing Fig. 5 vertically (fixing complexity), the LRLD will achieve better performance than the LSD when $l_{\text{des}} \leq 256$. Specifically, comparing the LRLD with list length 128 to the LSD with list length 8, we see that the proposed algorithm is able to simultaneously improve performance by 0.2 dB and reduce maximum complexity by a factor of 8. Finally, we note that the proposed algorithm has a smaller spread between the average and maximum complexities for most list lengths, a desirable property for system designers seeking a more predictable complexity.

VII. CONCLUSION

This paper has proposed the LR-aided list detector for soft-output detection of MIMO channels with QAM inputs. Simulation results indicate that the LRLD exhibits a favorable performance-complexity trade-off with large QAM alphabets. Specifically, we observed the trend that the complexity per bit for the LRLD *decreases* as the size of the QAM alphabet increases.

REFERENCES

[1] H. Yao and G. W. Wornell, "Lattice-reduction-aided detectors for MIMO communication systems," *Proc. Global Telecommun. Conf.*, vol. 1, pp. 424-428, Nov. 2002.

[2] P. Silvola, K. Hooli, M. Juntti, "Suboptimal soft-output MAP detector with lattice reduction," *IEEE Signal Processing Letters*, in press.

[3] C. Windpassinger, L. H.-J. Lampe, and R. F. H. Fischer, "From lattice-reduction-aided detection towards maximum-likelihood detection in MIMO systems," *Proc. Int. Conf. on Wireless and Optical Commun.* (*WOC*), pp. 144-148, July 2003.

[4] W. Mow, "Universal lattice decoding: principle and recent advances," *Wireless Commun. and Mobile Comp.*; 2003; 3(5): 553-569.

[5] D. Wübben, R. Böhnke, V. Kühn, and K.-D. Kammeyer, "Near-maximum-likelihood detection of MIMO systems using MMSE-based lattice-reduction," *IEEE Conf. on Commun.*, vol. 2, pp. 798-802, June. 2004.

[6] D. W. Waters and J. R. Barry, "A reduced-complexity lattice-aided decision-feedback detector," *IEEE International Conf. on Wireless Networks, Communications, and Mobile Computing*, Maui, Hawaii, June 13-16, 2005.

[7] B. Hochwald, S ten Brink, "Achieving Near-Capacity on a Multiple-Antenna Channel", *IEEE Trans. Info. Theory,* vol 51, No. 8 pp389-399, Mar 2003.

[8] D. Le Ruyet, T. Bertozzi, B. Ozbek, "Breadth first algorithms for APP detectors over MIMO channels," *IEEE International Conf. on Comm 2004,* vol. 2, pp. 926-930, June 20-24, 2004.

[9] B. Steingrimsson, Z. Luo, K.M. Wong, "Soft quasi-maximum likelihood detection for multiple-antenna wireless channels," *IEEE Trans. on Signal Processing*, vol. 51, no. 11, pp. 2710-2718, Nov. 2003.

[10] Y. L. C. de Jong, T. J. Willink, "Iterative Tree Search Detection for MIMO Wireless Systems," *IEEE Trans. on Commun.,* vol. 53, no. 6, Jun. 2005.

[11] D. J. Love, S. Hosur, A. Batra, and R. W. Heath Jr., "Space-Time Chase Decoding," *IEEE Trans. on Wireless Comm.,* vol. 4, no. 5, pp. 2035-2039, Sept. 2005.

[12] H. Zhu, B. Farhang-Boroujeny, and R. R. Chen, "On performance of sphere decoding and Markov chain Monte Carlo detection methods," *IEEE Signal Processing Letters,* vol. 12, no 10, Oct. 2005.

[13] E. Agrell, T. Eriksson, A. Vardy, and K. Zeger, "Closest Point Search in Lattices," *IEEE Trans. on Information Theory,* vol. 48, no. 8, pp. 2201-2214, August 2002.

[14] A. K. Lenstra, H. W. Lenstra, and L. Lovász, "Factoring Polynomials with Rational Coefficients," *Math. Ann,* vol. 261, pp. 515-534, 1982.

[15] J. Liu and J. Li, "Turbo Processing for an OFDM-Based MIMO System," *IEEE Trans. on Wireless Comm,* vol. 4, No. 5, Sept. 2005.