LATTICE DECODING OF LAYERED VERTICAL SPACE-TIME CODES

Miguel Bazdresch and Jorge Rodríguez-Guisantes

Instituto Tecnológico y de Estudios Superiores de Occidente
Periférico Sur 8585, Guadalajara, Jalisco, Mexico 45090
phone: + (52) 3336693434, fax: + (52) 33, email: miguelb@ieee.org
web: kino.iteso.mx/~miguelbaz

Ecole Nationale Supérieure des Télécommunications
46 rue Barrault, Paris, France 75013
email: rodriguez@enst.fr
web: www.comelec.enst.fr/~rodrigez

ABSTRACT

In this paper we study the use of lattice decoders in the reception of layered vertical space-time codes with square constellations. We rewrite the vertical code reception problem to make it amenable to lattice decoding. We compare the complexity and probability of error of lattice decoding with those of V-BLAST. Several properties of the behavior of lattice decoders that will have a definite impact on an efficient implementation are identified and characterized. Also, we evaluate the impact of performing the LLL lattice basis reduction on total receiver complexity.

1. INTRODUCTION

Vertical space-time codes, introduced in [1], are a technique used to create diversity in the multiple-input, multiple-output (*MIMO*) channel. In such a communications system, a vector $\mathbf{a}$ of n_T symbols is transmitted by n_T antennas; each antenna transmits one symbol of $\mathbf{a}$. All symbols are drawn from the same signal constellation S; it is assumed that the constellation is square (for example, 16-QAM). All transmit antennas are symbol synchronized. At the receiver, n_R antennas observe the transmitted vector ($n_R \geqslant n_T$). It is assumed that the channel between each pair of antennas presents slow, non-frequency selective Rayleigh fading. It is further assumed that the receiver has perfect knowledge of the channel. In these circumstances, the communications system can be modeled by the equation

$$\mathbf{r} = \mathbf{H}\mathbf{a} + \mathbf{n},$$

where $\mathbf{r} \in \mathbb{C}^{n_R}$ is the received vector, $\mathbf{H} \in \mathbb{C}^{n_R \times n_T}$ is known as the channel matrix, and $\mathbf{n} \in \mathbb{C}^{n_R}$ is a noise vector. Element h_{ij} of $\mathbf{H}$ represents the transfer function between transmit antenna j and receive antenna i, and is a complex number with zero mean and variance equal to 0.5 per dimension. Matrix $\mathbf{H}$ is assumed to be full-rank. Furthermore, the channel is assumed to be constant during the transmission of L symbols. The noise is assumed to be white Gaussian; each element of $\mathbf{n}$ has zero mean and variance $N_0/2$. The problem of the receiver is to estimate the transmitted vector $\mathbf{a}$.

An algorithm known as V-BLAST, first proposed in [2], has proved to be an interesting receiver, presenting attractive error rates at relatively low complexity. It is, however, suboptimal; by design, V-BLAST trades optimality for complexity. An alternative is to use a lattice decoder, which promises much better error rates than V-BLAST, at the cost of higher complexity. Geometrically, a lattice is a regular, periodic, infinite arrangement of points in n dimensions. Formally, let $\mathbf{G}$ be a matrix of real elements, with n rows and m columns, whose rows are linearly independent (which implies $n \leqslant m$). The lattice generated by $\mathbf{G}$ is defined as the set of vectors

$$\Lambda(\mathbf{G}) = \{\mathbf{uG} \colon \mathbf{u} \in \mathbb{Z}^n\}.$$

Matrix $\mathbf{G}$ is called the generator matrix of Λ, and its rows are called the basis vectors of the lattice. Given a lattice Λ and a vector $\mathbf{x} \in \mathbb{R}^m$, the problem of finding a vector $\hat{\mathbf{c}} \in \Lambda$ such that

$$\| \mathbf{x} - \hat{\mathbf{c}} \| \quad \leq \quad \| \mathbf{x} - \mathbf{c} \| \qquad \text{for all } \mathbf{c} \in \Lambda,$$

where $\| \cdot \|$ is the ℓ^2 norm, is called the *closest point problem*; an algorithm that solves this problem is called a lattice decoder. In the case where the received vectors $\mathbf{r}$ form a lattice, a lattice decoder is equivalent to a maximum-likelihood (ML) receiver. In practice, the receiver vectors form a finite set and can, at most, be represented as a subset of a lattice.

In this paper we have studied a receiver based on lattice decoding (the algorithm used is that proposed in [3], which is one of the fastest known to date). We have determined the improvement in error rate, and the cost in complexity, obtained by the use of lattice decoding. We have found that V-BLAST presents, under some circumstances, error rates that are very similar to those of lattice decoders; on the other hand, we have determined that lattice decoding can be less complex than V-BLAST in some circumstances.

We also present results on the benefits of reducing the lattice generator matrix before the actual decoding takes place. We have studied the LLL reduction [4]. The process of reduction consists in finding a matrix that generates the same lattice as the original one, but whose row vectors have smaller length. Since both the original generator matrix and the reduced version generate exactly the same lattice, the LLL reduction has no effect on the bit-error probability of the proposed receivers.

The benefits of matrix reduction have been studied recently in [5]. Just as in [3], however, the complexity results presented there do not take into account the complexity of the LLL reduction itself. We present results that include the complexity of both the lattice decoder algorithm and the matrix reduction. We have observed certain peculiar behaviors of lattice decoding algorithms that will have a definite impact on an efficient implementation.

Our first task was to adapt the vertical code reception problem to lattice decoding. This process is described in Section 2. Section 3 presents a comparison of block error rates between lattice decoding and V-BLAST. Section 4 does the same for their complexity, with and without lattice basis reduction. Our conclusions are presented at the end.

2. LATTICE REPRESENTATION OF VERTICAL CODES

In this section we describe the processing that the receiver must perform on the received signal in order to convert it into a subset of a lattice. We also describe a solution to the problem that arises when the lattice decoder returns as estimate a point that does not belong to the constellation in use.

Assume that a QAM constellation is being used and that it has $2n$ points per side (for n a suitable integer), and that the symbols with least energy have energy $2e_1^2$. Let the real matrix $\hat{\mathbf{H}}$ be equal to

$$\tilde{\mathbf{H}} = \left[\begin{array}{cc} \Re(\mathbf{H}) & \Im(\mathbf{H}) \\ -\Im(\mathbf{H}) & \Re(\mathbf{H}) \end{array} \right],$$

where $\Re(\mathbf{H})$ and $\Im(\mathbf{H})$ are the real and imaginary parts of $\mathbf{H}$, respectively. Let a generator matrix $\mathbf{G} = 2e_1\tilde{\mathbf{H}}^T$. Furthermore, let $\tilde{\mathbf{r}} = [\Re(\mathbf{r}), \Im(\mathbf{r})]$, and let $\mathbf{t}$ be a vector whose elements are equal to $2n-1$. Then, for any received point $\mathbf{r}$, the point

$$\mathbf{x} = \tilde{\mathbf{r}} + \mathbf{t}e_1\mathbf{G}$$

belongs to $\Lambda(\mathbf{G})$. The lattice decoder can, then, operate on $\mathbf{x}$ and $\mathbf{G}$.

In a noiseless environment, the decoder returns a vector $\hat{\mathbf{u}}$ with real integer elements u_i, $1 \leqslant i \leqslant 2n_T$, $0 \leqslant u_i \leqslant 2n-1$, which can be readily mapped to the original QAM constellation in order to produce an estimate $\hat{\mathbf{a}}$. In the presence of noise, however, $\hat{\mathbf{u}}$ might have no direct correspondence with any element of the original constellation; that is, some of its elements might take values less than 0 or larger than $2n-1$.

Several methods have been proposed to handle this problem. These range from simply declaring an erasure, to complex methods involving the projection of $\mathbf{x}$ on the surface of the region defined by the QAM constellation on the lattice [3]. Some of these methods can ensure finding the constellation point closest to the received point, but at the cost of extra complexity.

We propose here a method that is almost as simple as declaring an erasure, but which produces considerably better error rates. Our method consists in performing a rounding operation on the elements of $\hat{\mathbf{u}}$, so that the condition $0 \leqslant u_i \leqslant 2n-1$ is satisfied.

Using these ideas, we propose the algorithm V-LD to receive vertical space-time codes using a lattice decoder. This algorithm is based on the *ClosestPoint* and *Decode* algorithms presented in [3]. It has been modified to operate on the L symbol vectors that are transmitted while the channel is constant. The rounding operation described above is performed in lines 16 to 22.

Algorithm 1 V-LD

Input: an $n_R \times n_T$ matrix $\mathbf{H}$, a set of L $n_R \times 1$ vectors $\mathbf{r}_i$, $i = 1, \ldots, L$, and a signal constellation S. A boolean variable R determines whether to perform the LLL reduction of the lattice basis or not.

Output: a set of L $n_T \times 1$ vectors $\hat{\mathbf{u}}_i \in \mathbb{Z}^{2n_T}$ from which the original information bits can be estimated.

```
1:  Let G = 2e_1 H̃^T.
2:  Let t_s be a 1 × 2n_T vector with elements equal to
        (2n − 1)e_1.
3:  Let t = t_s H̃.
4:  if R is true then
5:      Let G_2 = LLL(G)
6:      Let W = G_2 G^{-1}
7:  else
8:      Let G_2 = G
9:      Let W be the identity matrix
10: end if
11: Compute a 2n_R × 2n_T matrix Q with orthonormal
        columns and a 2n_T × 2n_T lower-triangular matrix
        G_3 with positive diagonal elements, such that
        G_2 = G_3 Q.
12: Let H_3 = G_3^{-1}.
13: for i = 1 to L do
14:     Let x = (r̃_i + t)Q^T.
15:     Let x̂ = Decode(H_3, x) · W
16:     for j = 1 to 2n_T do
17:         if x̂_j > (2n − 1) then
18:             x̂_j = (2n − 1)
19:         else if x̂_j < 0 then
20:             x̂_j = 0
21:         end if
22:     end for
23:     û_i = x̂
24: end for
```

In algorithm V-LD, the LLL reduction is optional and is controlled by a boolean value; this is done in lines 4 to 10. The purpose of the LLL reduction is to speed the execution of *Decode* in line 15.

Figure 1 shows a comparison of the bit-error rates of ML reception, V-LD, and declaring an erasure whenever the point found by *Decode* has no correspondence in the original constellation. Our simulation results indicate that V-LD is a better alternative than the erasure option, performing similarly to ML in many cases.

3. COMPARISON OF BLOCK-ERROR RATES

A block error is defined as the occurrence of at least one bit error in a block of L symbol vectors. We present simulation results comparing V-BLAST to V-LD in figure 2, for $n_T = 4$ and $n_R = 4, 6$ and 8. It is worth remarking

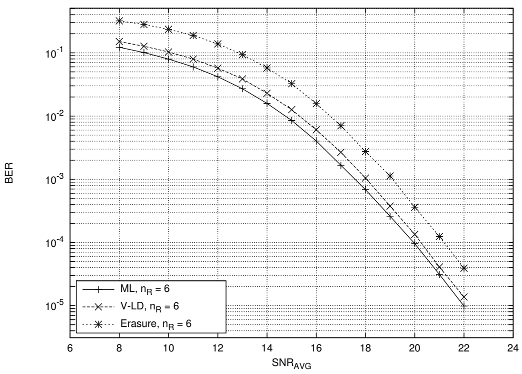

Figure 1: Bit-error rate comparison between ML, V-LD, and declaring an erasure, for $n_T = 4$ and $n_R = 6$.

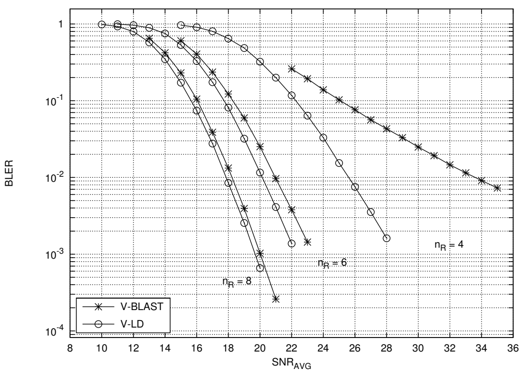

Figure 2: Block-error rate comparison between V-BLAST and V-LD for $n_T = 4$ and $n_R = 4, 6, 8$.

that, for a symmetrical ($n_T = n_R$) system, V-BLAST is not quite able to exploit the existing diversity, and, in consequence, V-LD outperforms it by a substantial amount.

As n_R grows with respect to n_T, however, the two algorithms present similar performance. V-BLAST has proved to be extremely good at exploiting spatial diversity in this case.

4. COMPLEXITY COMPARISONS

Next, we studied two aspects of the complexity of V-LD. We compared it to V-BLAST, and we investigated the LLL reduction's effect on total complexity.

Complexity is expressed as the total number of operations (arithmetic and memory) that each algorithm requires per information bit. This number is referred to as O_b. Lattice decoding is an iterative process whose complexity is very hard to determine in the form of a general formula. For this reason, we developed a simulator (which is available from the authors) that can count each operation performed. It should be emphasized that the results obtained are independent of any specific architectural details of the device where the simulation is run.

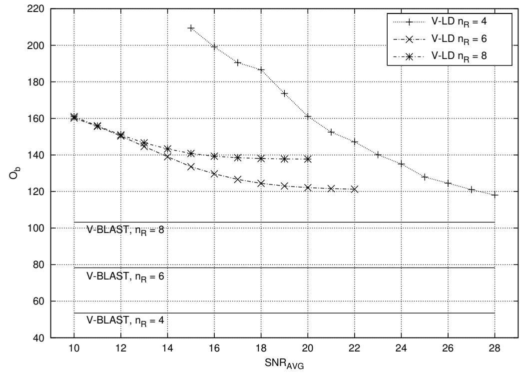

Figure 3: Complexity comparison between V-BLAST and V-LD for $n_T = 4$ and $n_R = 4, 6$, and 8.

4.1 Comparing V-BLAST and V-LD

Figure 3 shows O_b as a function of average SNR for $n_T = 4$ and $n_R = 4, 6$ and 8. Several conclusions can be drawn from it. For instance, it is interesting to see that the complexity of V-LD is a function of average SNR, whereas V-BLAST's is constant. Interestingly, it is seen that, for low SNR, V-LD's complexity is higher for the smaller $n_T = n_R = 4$ system than for the larger antenna combinations. It is only for SNR higher than a certain threshold that the intuitive notion of higher complexity for larger systems holds true.

A possible explanation is that the search size (the number of points that must examined in order to find the closest one) increases as the noise power increases, and decreases as the dimension of the lattice grows. After the noise power reaches a certain point, however, the search size is dominated by the lattice dimension.

The search size grows when the noise power increases because it is very likely that the received point will be surrounded by many lattice points, none of them particularly close to it. Thus, the algorithm needs to examine many points. This effect is countered by the dimension of the lattice: as it grows, the volume occupied by any given number of points increases, effectively separating the points, and providing a measure of noise immunity.

For high SNR, however, the received point will with high probability be found close to a lattice point, which will be quickly determined by any efficient algorithm. The noise plays a much smaller role in this case and the dominant factor in the search size is the lattice dimension.

Finally, from figure 3 it is apparent that V-BLAST is, as expected, less complex than V-LD. We have identified situations, though, where V-LD has lower complexity than V-BLAST. Such a case is depicted in figure 4, where it is seen that, for $n_T = 8$, $n_R = 16$, and SNR larger than 16dB, V-LD can be substantially less complex than V-BLAST.

4.2 Effect of the LLL reduction on V-LD's complexity

It has been shown in [3] that performing the LLL reduction reduces the complexity of the *Decode* algorithm. The net effect of the LLL on total complexity was not

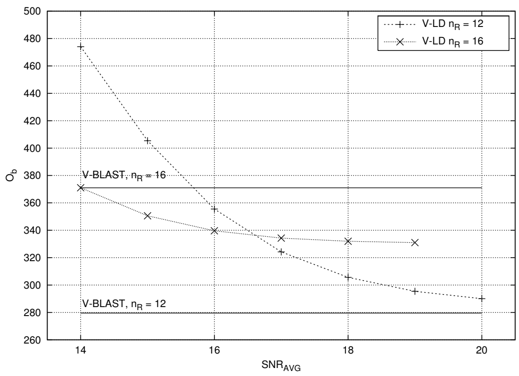

Figure 4: Complexity comparison between V-BLAST and V-LD for $n_T = 8$ and $n_R = 12$ and 16.

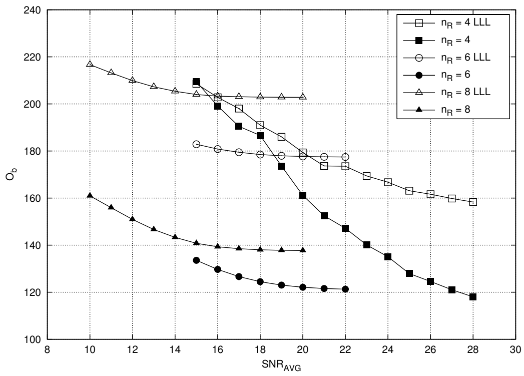

Figure 5: Complexity comparison of V-LD with and without the LLL reduction, for $n_T = 4$ and $n_R = 4, 6$ and 8, with $L = 10$.

studied there, however. Our simulations have shown that, in fact, the reduction has a negative effect on V-LD's performance in many cases. Figure 5 shows results for $n_T = 4$ and $n_R = 4, 6$ and 8, with $L = 10$. The net effect continues to be negative for values of L of at least 100.

The LLL does have positive effects in some situations. Figure 6 shows that for $n_T = n_R = 8$, the LLL does reduce total complexity. The figure also shows that V-LD can suffer from complexity spikes, which are large, unexpected increases in complexity for certain realizations of the channel matrix $\mathbf{H}$. The LLL reduction alleviates this, making the complexity of V-LD more predictable.

5. CONCLUSIONS

An algorithm (called V-LD) that adapts lattice decoding to the reception of layered vertical space-time codes has been proposed. This algorithm is able to decode points that have no direct correspondence to any of the transmitted vectors with little effect on complexity. It has been shown that it has better error-rate performance than another known low-complexity technique, which involves declaring an erasure.

V-LD's error-rate performance and complexity have been studied and compared to V-BLAST's. It has been

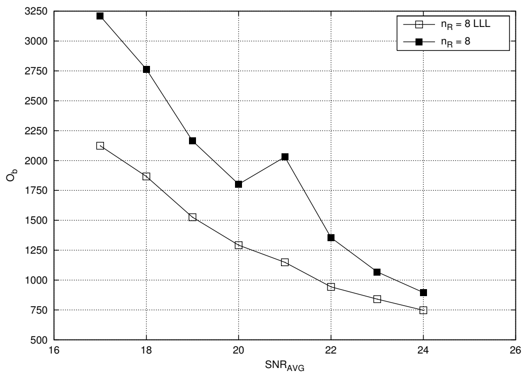

Figure 6: Complexity comparison of V-LD with and without the LLL reduction, for $n_T = n_R = 8$, with $L = 10$.

determined that for $n_R \gg n_T$ V-BLAST's performance is comparable to that of V-LD, while V-LD is less complex than V-BLAST is some situations.

The effects of system size, SNR, and the LLL reduction on V-LD's complexity have been analyzed. These effects will have a definite impact on the design of an efficient implementation of the algorithm.

REFERENCES

[1] G. J. Foschini, "Layered Space-Time Architecture for Wireless Communication in a Fading Environment When Using Multi-Element Antennas," *Bell Lab. Tech. J.*, vol. 1, No. 2, 1996, pp. 41–59.

[2] G. J. Foschini, G. D. Golden, R. A. Valenzuela and P. W. Wolnianksy, "Simplified Processing for High Spectral Efficiency Wireless Communication Employing Multi-Element Arrays," *IEEE J. on Selected Areas of Communications*, vol. 17, pp. 1841–1852, November 1999

[3] E. Agrell, T. Eriksson, A. Vardy and K. Zeger, "Closest Point Search in Lattices," *IEEE J. on Information Theory*, vol 48, August 2002, pp. 2201–2214.

[4] A. K. Lenstra, H. W. Lenstra Jr. and L. Lovász, "Factoring polynomials with rational coefficients," *Mathematische Annalen*, vol. 261, pp. 515–534, 1982.

[5] W. H. Mow, "Universal lattice decoding: a review and some recent results," Communications, 2004 IEEE Int. Conf. on, Vol. 5, pp. 2842–2846, June 2004.