the average frequency distribution

\[ H_{\text{avg}}(s) := \frac{1}{N} \sum_{i=1}^{N} H(s, i). \]

The behavior of the distributions for different values of \( K \) is discussed in Fig. 4. For \( M = 9 \) and \( K = 12 \), 16, 20, 24 the average frequency distribution \( H_{\text{avg}}(s) \) (solid lines) and the approximation \( H_{\text{app}}(s) \) (dashed lines) are compared to the Maxwell–Boltzmann distribution \( H_{M-B}(s) \) (dash-dotted lines). Here, the parameter \( \lambda \) is chosen so that the entropy is equal to \( K/8 \). Even for low \( K \) the approximation \( H_{\text{app}}(s) \) is very close to the true average frequency distribution \( H_{\text{avg}}(s) \). The approximation improves as \( K \) increases. Unfortunately, the Maxwell–Boltzmann distribution \( H_{M-B}(s) \) does not provide a good estimate of \( H_{\text{app}}(s) \). Shells with low index occur less often than expected from the optimal entropy–power tradeoff.

Finally, in Table II the average energy \( \sigma^2_{\text{av}} \) of the signal points in V.34 are summarized. For a symbol rate of 3200 Hz, the true average energy \( \sigma^2_{\text{av}} \) (cf. (2)), the approximate energy \( \sigma^2_{\text{app}} \) based on \( H_{\text{app}}(s) \), and the energy \( \sigma^2_{\text{M-B}} \) derived from the Maxwell–Boltzmann distribution are given for all possible data rates and associated mapping parameters \( K \), \( M \), and \( q \) [5, Table 10, expanded]. The underlying signal constellation is specified in [5, Fig. 5]. Again, the exact calculation and the approximation are very close. Obviously, the energies derived from the Maxwell–Boltzmann distribution underestimate the actual energies as they are lower bounds. The approximation (13) provides much better results.

IV. CONCLUSIONS

In this correspondence, a simple but general method for the calculation of the frequencies of the shells in shell-mapping schemes was derived. As an example, the method was shown in detail for the shell-mapping scheme specified for the international telephone-line modem standard ITU Recommendation V.34. The method starts with partial histograms that give the number of occurrences of shells within all possible combinations of \( n \)-tuples of shells with some fixed total cost. These histograms can be calculated easily using the generating functions that are needed in the encoder in any case. Then, the shell-mapping encoder is run with a specific input, namely, the maximum \( K \)-tuple. To each step of the encoding procedure a partial histogram can be assigned. Summing up these parts yields the final histograms. Thus the calculation has approximately the same complexity as the mapping encoder itself. With the knowledge of the frequencies of shells, the exact average transmit power can be calculated. Numerical examples are given for V.34.

ACKNOWLEDGMENT

The author wishes to acknowledge F. D. Neeser for valuable discussions and is indebted to the anonymous reviewers for their comments which improved the correspondence.

REFERENCES


A Universal Lattice Code Decoder for Fading Channels

Emanuele Viterbo, Member, IEEE, and Joseph Boutros, Member, IEEE

Abstract—We present a maximum-likelihood decoding algorithm for an arbitrary lattice code when used over an independent fading channel with perfect channel state information at the receiver. The decoder is based on a bounded distance search among the lattice points falling inside a sphere centered at the received point. By judicious choice of the decoding radius we show that this decoder can be practically used to decode lattice codes of dimension up to 32 in a fading environment.

Index Terms—Maximum-likelihood decoding, modulation, lattices, wireless channel.

I. INTRODUCTION

Lattice codes are used in digital transmission as high-rate signal constellations. They are obtained by carving a finite number of points from an \( n \)-dimensional lattice in the Euclidean space \( \mathbb{R}^n \). For the basic notations in lattice theory the reader can refer to [1]. Maximum-likelihood (ML) decoding of a lattice code used over an additive white Gaussian noise (AWGN) channel is equivalent to finding the closest lattice point to the received point. Many very efficient algorithms are now available for ML decoding some well-known root lattices [1]. Several Leech lattice decoders have been proposed with an ever-improving efficiency; a review of these decoders can be found in [2]. The above algorithms are strictly dependent on the special structure of the lattice being decoded (e.g., its being a binary lattice). Other algorithms for general nearest neighbor encoding in vector quantization are valid for any unstructured codebook. They do not take full advantage of the lattice structure which is useful for large

Manuscript received April 1, 1996; revised January 2, 1999.
E. Viterbo is with AT&T Shannon Laboratories, Florham Park, NJ 07932 USA, on leave from Politecnico di Torino, I-10129 Torino, Italy.
J. Boutros is with École Nationale Supérieure des Télécommunications, 75634 Paris, France.
Communicated by N. Seshadri, Associate Editor for Coding Techniques.

0018–9448/99$10.00 © 1999 IEEE
that is, we search for the shortest vector \( \mathbf{w} \) in the translated lattice \( \mathbf{r} - \Lambda \) in the \( n \)-dimensional Euclidean space \( \mathbb{R}^n \).

We write \( \mathbf{x} = \mathbf{uM} \) with \( \mathbf{u} \in \mathbb{Z}^n \), \( \mathbf{r} = \rho \mathbf{M} \) with \( \rho = (\rho_1, \ldots, \rho_n) \in \mathbb{R}^n \), and \( \mathbf{w} = \xi \mathbf{M} \) with \( \xi = (\xi_1, \ldots, \xi_n) \in \mathbb{R}^n \). Note that \( \rho \) and \( \xi \) are real vectors. Then we have \( \mathbf{w} = \sum_{i=1}^{n} \xi_i \mathbf{v}_i \), where \( \xi_i = \rho_i - u_i \), \( i = 1, \ldots, n \) define the translated coordinate axes in the space of the integer component vectors \( \mathbf{u} \) of the cubic lattice \( \mathbb{Z}^n \).

The sphere of square radius \( C \) and centered at the received point is transformed into an ellipsoid centered at the origin of the new coordinate system defined by \( \xi \)

\[
||w||^2 = Q(\xi) = \xi M M^T \xi^T = \xi G \xi^T = \sum_{i=1}^{n} \sum_{j=0}^{n} g_{ij} \xi_i \xi_j \leq C. \quad (3)
\]

Cholesky’s factorization of the Gram matrix \( G = M M^T \) yields \( G = R R^T \), where \( R \) is an upper triangular matrix. Then

\[
Q(\xi) = \xi R^T R \xi^T = ||R \xi||^2 = \sum_{i=1}^{n} \left( \sum_{j=0}^{n} r_{ij} \xi_j \right)^2 \leq C. \quad (4)
\]

Substituting \( q_{ij} = r_{ij}^2 \) for \( i = 1, \ldots, n \) and \( q_{ij} = r_{ij}^2 / r_{ii} \) for \( i = 1, \ldots, n \), \( j = i + 1, \ldots, n \), we can write

\[
Q(\xi) = \sum_{i=1}^{n} q_{ii} \left( \xi_i + \sum_{j=0}^{n} q_{ij} \xi_j \right)^2 \leq C. \quad (5)
\]

Starting from \( \xi_n \) and working backwards, we find the equations of the border of the ellipsoid. The corresponding ranges for the integer components \( u_n \) and \( u_{n-1} \) are

\[
\begin{align*}
- \sqrt{\frac{C}{q_{nn}} + \rho_n} & \leq u_n \leq \sqrt{\frac{C}{q_{nn}} + \rho_n} \\
- \sqrt{\frac{C - q_{nn} \xi_n^2}{q_{n-1,n-1} + \rho_{n-1} + q_{n-1,n} \xi_n}} & \leq u_{n-1} \leq \sqrt{\frac{C - q_{nn} \xi_n^2}{q_{n-1,n-1} + \rho_{n-1} + q_{n-1,n} \xi_n}}
\end{align*}
\]

where \( [x] \) is the smallest integer greater than \( x \) and \( [x] \) is the greatest integer smaller than \( x \). For the \( i \)th integer component we have

\[
\begin{align*}
- \frac{1}{q_{ii}} \left( C - \sum_{j=0}^{n} q_{ij} \left( \xi_i + \sum_{j=0}^{n} q_{ij} \xi_j \right)^2 \right) + \rho_i + \sum_{j=0}^{n} q_{ij} \xi_j & \leq u_i \leq \frac{1}{q_{ii}} \left( C - \sum_{j=0}^{n} q_{ij} \left( \xi_i + \sum_{j=0}^{n} q_{ij} \xi_j \right)^2 \right) \\
& + \rho_i + \sum_{j=0}^{n} q_{ij} \xi_j
\end{align*}
\]

(6)

The search algorithm proceeds very much like a mixed-radix counter on the digits \( u_i \), with the addition that the bounds change whenever there is a carry operation from one digit to the next. In practice, the bounds can be updated recursively by using the following equations:

\[
S_i = S_i(\xi_{i+1}, \ldots, \xi_n) = \rho_i + \sum_{j=0}^{n} q_{ij} \xi_i
\]

\[
T_i = T_i(\xi_{i+1}, \ldots, \xi_n) = C - \sum_{j=0}^{n} q_{ij} \left( \xi_i + \sum_{j=0}^{n} q_{ij} \xi_j \right)^2 \\
= T_i - q_{ii}(S_i - u_i)^2.
\]
When a vector inside the sphere is found, its square distance from the center (the received point) is given by

\[ d^2 = C - T_1 + q_{11} (S_1 - u_1)^2. \]

This value is compared to the minimum square distance \( d^2 \) (initially set equal to \( C \)) found so far in the search. If it is smaller then we have a new candidate closest point and the search continues like this until all the vectors inside the sphere are tested.

The advantage of this method is that we never test vectors with a norm greater than the given radius. Every tested vector requires the computation of its norm, which entails a number of operations that can greatly speed up the decoder. In practice, the choice of \( C \) can be adjusted according to the noise variance \( \sigma^2 \).

The kernel of the universal decoder (the enumeration of lattice points inside a sphere of radius \( \sqrt{C} \)) requires the greatest number of operations. The complexity of this additional test depends on the shape of the constellation. For cubic-shaped constellations it only entails checking that the vector components lay within a given range. For a spherically shaped signal set it is sufficient to compute the length of the code vector found in the search sphere in order to check if it is within the outermost shell of the constellation.

### III. The Sphere Decoder with Fading

For ML decoding with perfect CSI at the receiver, the problem is to minimize metric (1). Let \( M \) be the generator matrix of the lattice \( \Lambda \) and let us consider the lattice \( \Lambda_c \) with generator matrix

\[ M_c = M \text{diag}(\alpha) \]

We can imagine this new lattice \( \Lambda_c \), in a space where each component has been compressed or enlarged by a factor \( \alpha \). A point of \( \Lambda_c \) can be written as \( x^{(c)} = (x_1^{(c)}, \ldots, x_n^{(c)}) = (x_1, \ldots, \alpha_n) \). The metric to minimize is then

\[ m(x | r, \alpha) = \sum_{i=1}^{n} |r_i - x_i^{(c)}|^2. \]

This means that we can simply apply the lattice decoding algorithm to the lattice \( \Lambda_c \), when the received point is \( r \). The decoded point \( \hat{x}^{(c)} \in \Lambda_c \), has the same integer components \( (\hat{u}_1, \ldots, \hat{u}_n) \) as \( \hat{x} \in \Lambda \).

The additional complexity required by this decoding algorithm comes from the fact that for each received point we have a different
compressed lattice $\Lambda_s$. So we need to compute a new Cholesky factorization of the Gram matrix for each $\Lambda_s$, which requires $O(n^3/3)$ operations. We also need $M^{-1} = \text{diag}(1/\alpha_1, \ldots, 1/\alpha_n)M^{-1}$ to find the $\rho_i$'s, but this only requires a vector-matrix multiplication since $M^{-1}$ is precomputed. The complete flowchart of the algorithm is given in Fig. 2.

The choice of $C$ in this case is more critical. In fact whenever we are in the presence of deep fades then many points fall inside the search sphere and the decoding can be very slow. This is also evident from the fact that the Gram matrix of $\Lambda_s$ may have a very small eigenvalue which gives a large exponent $d$ in (1). This problem may be partially overcome by adapting $C$ according to the values of the fading coefficients $\alpha_i$.

Fig. 3 shows the performance of the rotated lattice constellation $Z^{31,12}$ on the Rayleigh channel with a spectral efficiency of 2 bits/dimension. This lattice is a rotated version of the cubic lattice in dimension 24 with a diversity order equal to 12 given in [5]. For all rotated cubic lattices in [5], we can set the search radius $C = 1$ and thus the enumeration complexity increases as $O(n^3)$ if we do not take into account the fading.

Fig. 3 compares the performance of the $Z^{24,12}$ constellation to the 16-QAM on a Gaussian channel which has the same spectral efficiency. We observe that such a high modulation diversity can bring the bit error rate within 2 dB from the Gaussian channel’s curve. To show the effectiveness of rotated constellations with respect to other TCM schemes especially designed for the fading channel, we have also plotted the bit-error rate of the optimal 64-state TCM over an 8-PAM signal set. We recall that the asymptotic slope of the error curve reflects the diversity order of the coding scheme. Then, we observe that the diversity of the TCM scheme is much lower than the one of the rotated constellation.

IV. CONCLUSION

Decoding arbitrary signal constellations in a fading environment can be a very complex task. When the signal set has no structure it is only possible to perform an exhaustive search through all the constellation points. Some signal constellations, which can be efficiently decoded when used over the Gaussian channel, become hard to decode when used over the fading channel since their structure is destroyed. Fortunately, for lattice constellations this is not the case since the faded constellation still preserves a lattice structure and only a small additional complexity is required. The algorithm we presented was successfully run to simulate systems using lattice constellations of dimensions up to 32 which seem to be sufficient to approach the performance of the Gaussian channel when dealing with a Rayleigh fading one.

REFERENCES