VI. Conclusion

Separation results have been obtained for causal coding problems with channel feedback information. These results were used to show that causal coding is useless for symmetric channels.

We hope to extend those results to control systems in a subsequent paper.

References

[1] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech J.*, vol. 22, pp. 379–423, 1948.
[2] R. G. Gallager: *Information Theory and Reliable Communication*. New York: Wiley, 1968.
[3] H. S. Witsenhausen, "On the structure of real-time source coders," *Bell Syst. Tech. J.*, vol. 58, pp. 1437–1451, 1979.
[4] A. W. Drake, "Observation of a Markov process through a noisy channel," Sc.D. thesis, Department of Electrical Engineering, M.I.T., Cambridge, MA, 1962.
[5] J. L. Devore, "A note on the observation of a Markov source through a noisy channel," *IEEE Trans. Inform. Theory*, vol. IT-20, pp. 762–764, 1974.
[6] I. I. Gihman and A. V. Skorohod, *Controlled Stochastic Processes*. New York: Springer Verlag, 1979.
[7] J. Marschak and K. Miyasawa, "Economic comparability of information systems," *Int. Economic Rev.*, vol. 9, no. 2, pp. 137–174, 1968.
[8] G. Munson, "Causal information transmission with feedback," Ph.D. dissertation, School of Electrical Engineering, Cornell University, Ithaca, NY, 1981.
[9] J. G. Dunham, "An iterative theory for code design," preprint, 1980.
[10] D. Teneketzis, "Communication in decentralized control," Ph.D. dissertation, Department of Electrical Engineering and Computer Science, M.I.T., Cambridge, MA, 1979.
[11] C. Striebel, "Sufficient statistics in the optimum control of stochastic systems," *J. Math. Anal. Appl.*, vol. 12, pp. 576–592, 1965.

A Fast Encoding Method for Lattice Codes and Quantizers

JOHN H. CONWAY AND N. J. A. SLOANE, FELLOW, IEEE

***Abstract*—In an earlier paper the authors described a very fast method which, for the root lattices A_n, D_n, E_n, their duals and certain other lattices, finds the closest lattice point to an arbitrary point of the underlying space. If the lattices are used as codes for a Gaussian channel, the algorithm provides a fast decoding procedure, or if they are used as vector quantizers the algorithm performs the analog-to-digital conversion efficiently. The present paper offers a solution to the inverse problem for the same lattices (the encoding problem for channel codes or the digital-to-analog part of quantizing), namely, given an integer k, to find the kth code vector, and to the closely related problem of finding the index k of a given code vector.**

I. Introduction

A EUCLIDEAN CODE is a finite set of points $x_1, \cdots, x_M$ in n-dimensional real Euclidean space $\boldsymbol{R}^n$. Euclidean codes can be used as signal sets for a Gaussian channel, as representative points (or output vectors) in a vector quantizer, and in many other applications—for a survey with an extensive bibliography see [11]. There are a number of desirable properties that a Euclidean code should have:

1) the number M of code vectors should be large;
2) the total energy $\Sigma\|x_i\|^2$ (or alternatively the peak energy max $\|x_i\|^2$) should be small;
3) the minimum distance between the x_i should be large (or alternatively the probability of incorrect decoding should be small, if the code is used on a Gaussian channel);
4) given k, we should be able to find the kth code vector x_k easily;
5) given x_k, we should be able to find its index k easily; and
6) given an arbitrary point $z \in \boldsymbol{R}^n$, it should be easy to find the closest code vector x_k.

These requirements are of course not independent. (Some of the trade-offs among 1), 2), and 3) are discussed in a classic paper by Slepian [9], and we are currently investigating the error probability of the Voronoi codes described below. That work will be reported elsewhere.) We are concerned in this paper with *lattice codes*, which consist of a subset of points of some lattice in $\boldsymbol{R}^n$, i.e., which are a set of centers of a lattice packing of equal n-dimensional

Manuscript received June 24, 1982; revised February 16, 1983.
J. H. Conway is with the Department of Pure Mathematics and Mathematical Statistics, University of Cambridge, Cambridge CB2 1SB, England.
N. J. A. Sloane is with the Mathematics and Statistics Research Center, Bell Laboratories, Murray Hill, NJ 07974.

0018-9448/83/1100-0820$01.00 ©1983 IEEE

spheres. (See, for example, Figs. 1, 2, and 3. Other examples will be found in Section II and in [1], [3], and [11].) The code is defined by specifying a lattice Λ in $\boldsymbol{R}^n$ and a certain region of the space $\boldsymbol{R}^n$, and consists of all lattice points inside this region.

For these codes the minimum distance is the minimum distance between lattice points, and the number of code vectors is determined by the density of the lattice (see [10]). Then requirements 1) and 3) amount to saying that the lattice should have a high density, and 2) states that the region of space defining the code should be as nearly spherical as possible.

Very fast algorithms were given in [3] (see also [1]) for solving problem 6) for a large class of lattice codes. In the present paper we propose a solution to the inverse problems 4) and 5) for the same lattices. Since if the codes are used for a Gaussian channel 6) is the decoding problem while 4) and 5) are encoding problems, we might expect 4) and 5) to be rather easier to solve than 6). However for these codes 4), 5), and 6) appear to be of comparable difficulty.

In order to solve problems 4) and 5) we shall only consider lattice codes that are defined by certain very special regions of space, regions that we may call Voronoi-shaped, and so we shall refer to these as *Voronoi codes.*

Voronoi codes have two drawbacks, as we shall see: only certain rates can be attained, and because the Voronoi-shaped regions defining them are not spherical, they do not in general have the lowest possible total energy (although in the examples that we give the difference is small). So we do not claim to have said the last word on this subject.

In the next section we define the new codes and give examples in two, four, and eight dimensions. Then Sections III and IV give the encoding algorithms that solve problems 4) and 5).

Our notation follows [2]. In particular the norm $\|x\|^2$ of a vector x is its squared length $x \cdot x$.

II. Voronoi Codes

Let Λ be a lattice in $\boldsymbol{R}^n$. Around each lattice point x is its Voronoi region $V(x)$, consisting of all points of $\boldsymbol{R}^n$ which are at least as close to x as to any other lattice point, or more formally

$$V(x) = \{z \in \boldsymbol{R}^n : \|x - z\| \leq \|y - z\|, \text{ for all } y \in \Lambda\}.$$

The Voronoi region $V(0)$ around the origin will simply be called the Voronoi region of the lattice. The Voronoi regions of many lattices are studied in detail in [2]. For example, Fig. 1 shows the hexagonal lattice A_2 and some of its Voronoi regions.

The codes to be constructed will contain $M = r^n$ lattice points, where $r = 1, 2, 3, \cdots$, i.e., they have rate

$$\frac{1}{n} \log_2 M = \log_2 r \text{ bits/dimension.}$$

Let V_r, denote the Voronoi region for the lattice $r\Lambda = \{rx: x \in \Lambda\}$. Thus V_r is simply the Voronoi region for Λ magnified r times. The codes are specified by giving the integer r and a vector $a \in \boldsymbol{R}^n$. Then

the Voronoi code $C_\Lambda(r, a)$ consists of all vectors $x - a$ for $x \in \Lambda \cap (a + V_r)$.

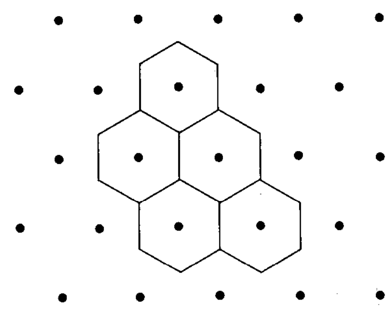

Fig. 1. A portion of the two-dimensional hexagonal lattice A_2, showing the hexagonal Voronoi regions around some of the lattice points.

In words, the code consists of all lattice vectors inside the Voronoi-shaped region V_r around the point a.

We choose a so that the boundary of $a + V_r$ does not contain any point of Λ. Then, since the quotient group $\Lambda / r\Lambda$ has order r^n, $C_\Lambda(r, a)$ contains r^n code points, as claimed. In Section III we shall see how to label every code point $x \in C_\Lambda(r, a)$ with its *index vector*

$$\text{index}(x) = (k_1, \cdots, k_n), \qquad \text{all } k_i \in \{0, 1, \cdots, r-1\}.$$

Examples

The value $r = 4$ (corresponding to 2 bits/dimension) is commonly used [6], and we shall adopt it in our examples. Although we are most interested in higher dimensions, where the advantages of these codes are more apparent, it is instructive to begin with a two-dimensional example.

The Lattice A_2

The hexagonal lattice A_2 (shown in Fig. 1) may be defined as the integer span of the vectors

$$v_1 = (1, 0), \quad v_2 = \left(\frac{1}{2}, \frac{\sqrt{3}}{2}\right).$$

The Voronoi region V_4 for $4A_2$ is enclosed by the broken lines in Fig. 2. By shifting V_4 slightly to the left, centering it around the point $a = (-\frac{1}{4}, 0)$, we obtain the Voronoi code $C_{A_2}(4, a)$ enclosed by the solid lines in Fig. 2.

The Best Choice for a

There are infinitely many choices for a, and we should like to find one that leads to a code of the smallest total energy, so as to satisfy requirement 2) of Section I. If $C = \{x_1, \cdots, x_M\}$ is any Euclidean code in $\boldsymbol{R}^n$ we define its *centroid* and *average energy* by

$$\hat{x} = \frac{1}{M} \sum_{i=1}^{M} x_i,$$

$$P(C) = \frac{1}{d^2 M} \sum_{i=1}^{M} \|x_i - \hat{x}\|^2,$$

where d is the minimum distance between code vectors.

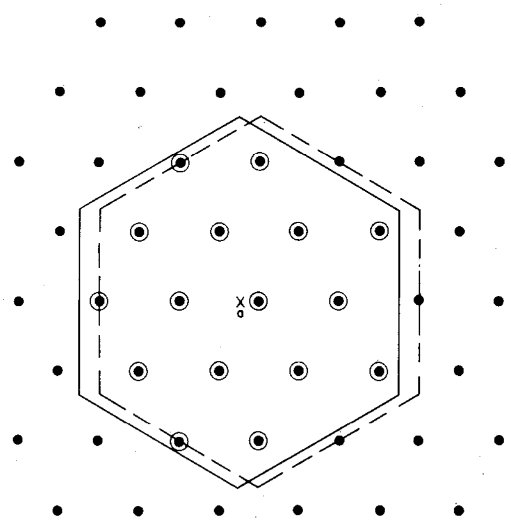

Fig. 2. The Voronoi region V_4 for $4A_2$ (broken line), the shifted region $a + V_4$ (solid line), and the Voronoi code $C_{A_2}(4, a)$ consisting of all 16 lattice points inside $a + V_4$ (hollow circles). This code has the lowest average energy $P(C)$ of any known set of 16 points in the plane.

(The factor d^2 is inserted to make $P(C)$ dimensionless.) Our Voronoi codes usually have centroid at a, and so $\hat{x} = 0$. So as to be able to compare different classes of codes we define

$$P_u(n, r), P_l(n, r), \text{ and } P_V(n, r)$$

to be the minimum of $P(C)$ when C ranges over

unrestricted, lattice, or Voronoi

codes, respectively, of dimension n and size n^r. Certainly

$$P_u(n, r) \leqslant P_l(n, r) \leqslant P_V(n, r).$$

At present (except for trivial cases) the exact values of these quantities appear to be unknown. In two dimensions the best code of any type containing 16 codewords seems to be the Voronoi code shown in Fig. 2, and so it seems plausible that

$$P_u(2,4) = P_l(2,4) = P_V(2,4) = \frac{35}{16} = 2.1875. \quad (1)$$

We find it slightly surprising that this code is superior to the compact arrangement shown in Fig. 3, for which $P(C) = 2.207 \cdots$. The best arrangement of 16 points discovered by Foschini *et al.* [6] (see [6, fig. 6(d) and table I]) is very similar to our Fig. 2, but has $P(C) = 2.243 \cdots$. Of course their code was designed to satisfy a different criterion, that of having minimum error probability. The 1-5-10 arrangement given in [6, fig. 7(f)] is still worse. The reader is invited to try arranging 16 pennies into an even better configuration than Fig. 2, or alternatively to prove that the conjecture stated in (1) is correct.

Four Dimensions

For most purposes the best lattice in four dimensions is that consisting of all points with integer coordinates (x_1, x_2, x_3, x_4) with $x_1 + \cdots + x_4$ even. It is usually denoted by D_4, and has been extensively studied [2], [3], [5], [8], [10], [11]. (In [5] this lattice is denoted by $\{3,3,4,3\}$.)

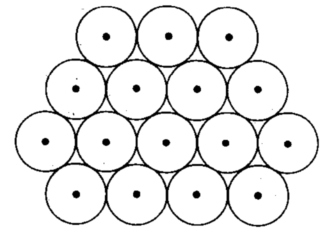

Fig. 3. Another subset of 16 points of the hexagonal lattice, but having greater average energy than that shown in Fig. 2.

Alternatively, D_4 may be defined as the integer span of the vectors v_1, v_2, v_3, v_4 that form the rows of the *generator matrix*

$$G = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}. \quad (2)$$

The best four-dimensional Voronoi code containing $4^4 = 256$ vectors that we have found is the code $C_{D_4}(4, a)$, where

$$a = \left(0, \frac{3}{16}, \frac{11}{32}, \frac{17}{32}\right). \quad (3)$$

The vectors in this code are listed in lexicographic order in Table I. The code has average energy $P(C) = 3.438 \cdots$, which we believe to be optimal among Voronoi codes, and so conjecture that $P_V(4,4) = 3.438 \cdots$.

This code was found by an iterative computer program, which takes an initial vector a and finds the r^n lattice points that are closest to a, using the algorithm of Section IV, then replaces a by the centroid of this set of points, and repeats. In our examples this procedure always converged after 2 or 3 iterations.

Remark

Since there are no lattice points on the boundary of $a + V_r$, small changes in a produce the same Voronoi code. Even relatively large changes have little effect on the code. For example, if instead of (3) we use the simpler vector

$$a' = \left(0, \frac{2}{8}, \frac{3}{8}, \frac{4}{8}\right)$$

then the code $C_{D_4}(4, a')$ has average energy $3.441 \cdots$, only slightly worse than before.

For comparison we describe two other lattice codes of size 256 that can be obtained from the D_4 lattice. First, if we simply take a set of 256 points that are closest to the origin, we obtain a code with $P(C) = 3.480 \cdots$, inferior to the Voronoi code. This code consists of the origin, and then successive layers of 24, 24, 96, 24, and 87 points,[1] for a total of 256. On the other hand a code having lower energy than our Voronoi code is obtained if we take the set of 256 points of D_4 that are closest to the nonlattice[2] point $(1,0,0,0)$. This code consists of successive layers of 8, 32, 48, 64, and 104 points, the 8 points closest to $(1,0,0,0)$ for example being $(0,0,0,0)$, $(2,0,0,0)$, $(1, \pm 1, 0, 0)$,

[1]The theta series of this lattice, which gives the number of points in each layer around the origin, may be found in [11].

[2]This is actually a point at maximum distance from D_4, or in other words a *deep hole* in that lattice in the notation of [4].

TABLE I
FOUR - DIMENSIONAL VORONOI CODE $C_{D_4}(4, a)$ OF SIZE 256, WHERE a IS GIVEN BY (3)*

-3	0	0	1	-3	0	1	0	-3	1	0	0	-3	1	1	1
-2	-1	-1	0	-2	-1	-1	2	-2	-1	0	-1	-2	-1	0	1
-2	-1	1	0	-2	-1	1	2	-2	-1	2	-1	-2	-1	2	1
-2	0	-1	-1	-2	0	-1	1	-2	0	0	0	-2	0	0	2
-2	0	1	-1	-2	0	1	1	-2	0	2	0	-2	0	2	2
-2	1	-1	0	-2	1	-1	2	-2	1	0	-1	-2	1	0	1
-2	1	1	0	-2	1	1	2	-2	1	2	-1	-2	1	2	1
-2	2	-1	-1	-2	2	-1	1	-2	2	0	0	-2	2	0	2
-2	2	1	-1	-2	2	1	1	-2	2	2	0	-2	2	2	2
-1	-2	-1	0	-1	-2	-1	2	-1	-2	0	-1	-1	-2	0	1
-1	-2	1	0	-1	-2	1	2	-1	-2	2	-1	-1	-2	2	1
-1	-1	-2	0	-1	-1	-2	2	-1	-1	-1	-1	-1	-1	-1	1
-1	-1	-1	3	-1	-1	0	-2	-1	-1	0	0	-1	-1	0	2
-1	-1	1	-1	-1	-1	1	1	-1	-1	1	3	-1	-1	2	0
-1	-1	2	2	-1	-1	3	1	-1	0	-2	-1	-1	0	-2	1
-1	0	-1	-2	-1	0	-1	0	-1	0	-1	2	-1	0	0	-1
-1	0	0	1	-1	0	0	3	-1	0	1	-2	-1	0	1	0
-1	0	1	2	-1	0	2	-1	-1	0	2	1	-1	0	3	0
-1	1	-2	0	-1	1	-2	2	-1	1	-1	-1	-1	1	-1	1
-1	1	-1	3	-1	1	0	-2	-1	1	0	0	-1	1	0	2
-1	1	1	-1	-1	1	1	1	-1	1	1	3	-1	1	2	0
-1	1	2	2	-1	1	3	1	-1	2	-1	0	-1	2	-1	2
-1	2	0	-1	-1	2	0	1	-1	2	1	0	-1	2	1	2
-1	2	2	-1	-1	2	2	1	-1	3	0	0	-1	3	1	1
0	-3	0	1	0	-3	1	0	0	-2	-1	-1	0	-2	-1	1
0	-2	0	0	0	-2	0	2	0	-2	1	-1	0	-2	1	1
0	-2	2	0	0	-2	2	2	0	-1	-2	-1	0	-1	-2	1
0	-1	-1	-2	0	-1	-1	0	0	-1	-1	2	0	-1	0	-1
0	-1	0	1	0	-1	0	3	0	-1	1	-2	0	-1	1	0
0	-1	1	2	0	-1	2	-1	0	-1	2	1	0	-1	3	0
0	0	-3	1	0	0	-2	0	0	0	-2	2	0	0	-1	-1
0	0	-1	1	0	0	-1	3	0	0	0	-2	0	0	0	0
0	0	0	2	0	0	0	4	0	0	1	-1	0	0	1	1
0	0	1	3	0	0	2	0	0	0	2	2	0	0	3	1
0	1	-2	-1	0	1	-2	1	0	1	-1	-2	0	1	-1	0
0	1	-1	2	0	1	0	-1	0	1	0	1	0	1	0	3
0	1	1	-2	0	1	1	0	0	1	1	2	0	1	2	-1
0	1	2	1	0	1	3	0	0	2	-1	-1	0	2	-1	1
0	2	0	0	0	2	0	2	0	2	1	-1	0	2	1	1
0	2	2	0	0	2	2	2	0	3	0	1	0	3	1	0
1	-2	-1	0	1	-2	-1	2	1	-2	0	-1	1	-2	0	1
1	-2	1	0	1	-2	1	2	1	-2	2	-1	1	-2	2	1
1	-1	-2	0	1	-1	-2	2	1	-1	-1	-1	1	-1	-1	1
1	-1	-1	3	1	-1	0	-2	1	-1	0	0	1	-1	0	2
1	-1	1	-1	1	-1	1	1	1	-1	1	3	1	-1	2	0
1	-1	2	2	1	-1	3	1	1	0	-2	-1	1	0	-2	1
1	0	-1	-2	1	0	-1	0	1	0	-1	2	1	0	0	-1
1	0	0	1	1	0	0	3	1	0	1	-2	1	0	1	0
1	0	1	2	1	0	2	-1	1	0	2	1	1	0	3	0
1	1	-2	0	1	1	-2	2	1	1	-1	-1	1	1	-1	1
1	1	-1	3	1	1	0	-2	1	1	0	0	1	1	0	2
1	1	1	-1	1	1	1	1	1	1	1	3	1	1	2	0
1	1	2	2	1	1	3	1	1	2	-1	0	1	2	-1	2
1	2	0	-1	1	2	0	1	1	2	1	0	1	2	1	2
1	2	2	-1	1	2	2	1	1	3	0	0	1	3	1	1
2	-1	-1	0	2	-1	-1	2	2	-1	0	-1	2	-1	0	1
2	-1	1	0	2	-1	1	2	2	-1	2	-1	2	-1	2	1
2	0	-1	-1	2	0	-1	1	2	0	0	0	2	0	0	2
2	0	1	-1	2	0	1	1	2	0	2	0	2	0	2	2
2	1	-1	0	2	1	-1	2	2	1	0	-1	2	1	0	1
2	1	1	0	2	1	1	2	2	1	2	-1	2	1	2	1
2	2	-1	-1	2	2	-1	1	2	2	0	0	2	2	0	2
2	2	1	-1	2	2	1	1	2	2	2	0	2	2	2	2
3	0	0	1	3	0	1	0	3	1	0	0	3	1	1	1

*The table gives the lattice points $x + a$ for $x \in C_{D_4}(4, a)$.

$(1, 0, \pm 1, 0)$, and $(1, 0, 0, \pm 1)$. For this code $P(C) = 3.375$, and so we have

$$P_u(4,4) \leqslant 3.375, P_l(4,4) \leqslant 3.375.$$

These results are summarized in Table II, which compares the best codes known in dimensions 2, 4, and 8.

Eight Dimensions

Just as in four dimensions, there is one eight-dimensional lattice which is better for most purposes than any other lattice. This is the Gosset lattice E_8 ([2]–[5], [8], [10], [11]), which may be defined by the generator matrix

$$G = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix}. \tag{4}$$

TABLE II
AVERAGE POWER $P(C)$ OF VARIOUS CODES OF SIZE 4^n IN n DIMENSIONS

Code	$n = 2$	$n = 4$	$n = 8$
Best Voronoi Code Known	2.1875	$3.438 \cdots$	$4.577 \cdots$
Code Centered at 0		$3.480 \cdots$	$4.546 \cdots$
Code Centered at a Nonlattice Point		3.375	$4.516 \cdots$

The best eight-dimensional Voronoi code containing $4^8 = 65536$ vectors that we have found is the code $C_{E_8}(4, a)$, where

$$a \doteq (0.01535, 0.05002, 0.08310, 0.14786, 0.18069, 0.21463, 0.25040, 0.71103), \tag{5}$$

for which the average energy is $P(C) = 4.577 \cdots$. (This was found by the computer program described above.) Again we believe that this is optimal and that $P_V(8,4) = 4.577 \cdots$.

For comparison a set of 65536 points of E_8 that are closest to the origin (consisting of successive layers of sizes[1] 1, 240, 2160, 6720, 17520, 30240 and 8655) has $P(C) = 4.546 \cdots$, which is slightly better than the Voronoi code. A set centered at the point[3] $(1,0,0,0,0,0,0,0)$ is better still: taking successive layers of sizes 16, 128, 448, 2016, 3584, 5504, 8192, 12112, 16128, and 16384 of E_8 vectors around this point gives $P(C) = 4.516 \cdots$. Thus

$$P_u(8,4) \leqslant 4.516 \cdots, P_l(8,4) \leqslant 4.516 \cdots.$$

On the other hand the fast encoding algorithms to be described in the next two sections do not apply to these lower energy codes.

III. SOLUTION TO PROBLEM 5): FINDING THE INDEX OF A CODE VECTOR

Let Λ be an n-dimensional lattice spanned by vectors $v_1, \cdots, v_n$. Then $v_1, \cdots, v_n$ are also a basis for $\boldsymbol{R}^n$, and therefore have a *dual basis* $v_1^*, \cdots, v_n^*$ (cf. [7, sec. 3.5]), satisfying

$$v_i \cdot v_i^* = 1, \quad v_i \cdot v_j^* = 0, \qquad i \neq j,$$

for $i, j = 1, \cdots, n$. The dual basis is easily computed: if G is a generator matrix for Λ, with rows $v_1, \cdots, v_n$, then the vectors of the dual basis are the *columns* of G^{-1}.

Now suppose a code vector $x \in C_\Lambda(r, a)$ is given, say $x = \sum_{i=1}^n c_i v_i - a$, $c_i \in \boldsymbol{Z}$. Then $(x + a) \cdot v_j^* = c_j$. So Problem 5) may be solved as follows.

Given a code vector x, calculate $c_j = (x + a) \cdot v_j^*$ for $j = 1, \cdots, n$, and find $0 \leqslant k_j \leqslant r - 1$ such that $k_j \equiv c_j \pmod r$. Then $\text{index}(x) = (k_1, \cdots, k_n)$.

In other words, compute the vector $(x + a)G^{-1}$ and reduce all its entries modulo r to the range $[0, r-1]$. The result is $\text{index}(x)$.

[3]A point of maximum distance from E_8 (see [2]).

Examples

Consider D_4, with generator matrix given by (2). Then

$$G^{-1} = \begin{bmatrix} \frac{1}{2} & 0 & 0 & 0 \\ -\frac{1}{2} & 1 & 0 & 0 \\ -\frac{1}{2} & 0 & 1 & 0 \\ -\frac{1}{2} & 0 & 0 & 1 \end{bmatrix}.$$

As an illustration, the first code vector in Table I is $x = (-3,0,0,1) - a$, and we find $(x + a)G^{-1} = (2,0,0,1)$, $\text{index}(x) = (2,0,0,1)$, since $r = 4$.

For E_8, defined by (4), we have

$$G^{-1} \begin{bmatrix} \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{1}{2} & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{1}{2} & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ -\frac{1}{2} & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ -\frac{1}{2} & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ -\frac{1}{2} & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ -\frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ \frac{5}{2} & -1 & -1 & -1 & -1 & -1 & -1 & 2 \end{bmatrix}$$

If $x = (7,-1,-1,-1,0,0,0,0) - a$ then

$$(x + a)G^{-1} = (5,-1,-1,-1,0,0,0,0),$$
$$\text{index}(x) = (1,3,3,3,0,0,0,0).$$

IV. Solution to Problem 4): Finding a Code Vector from its Index

Given an index vector $(k_1,\cdots,k_n)$ with $0 \leqslant k_i \leqslant r - 1$, we wish to find that vector x in the Voronoi code $C_\Lambda(r, a)$ for which $\text{index}(x) = (k_1,\cdots,k_n)$.

We first form $x' = \Sigma k_i v_i$, which has index equal to $(k_1,\cdots,k_n)$ but need not be in the code. The desired lattice point x is then the unique solution to

$$x \equiv x' \pmod{r\Lambda}, \qquad x \in a + V_r \tag{6}$$

(by the definition of $C_\Lambda(r, a)$). Equation (6) is easily solved if there is an algorithm (such as those we have described in [3]) available for solving Problem 6), i.e., for finding the closest point of Λ to an arbitrary point of the space. In fact, if we set $z = r^{-1}(x' - a)$, and λ is the closest point of Λ to z, then $x = x' - r\lambda$ is the desired lattice point. So Problem 4) may be solved as follows.

Given the index vector $(k_1,\cdots,k_n)$, calculate $x' = \Sigma k_i v_i$ and $z = r^{-1}(x' - a)$. Find the closest point $\lambda \in \Lambda$ to z (using, for example, the algorithms in [3]). Then $x = x' - r\lambda - a$ is in the Voronoi code $C_\Lambda(r, a)$ and has index$(k_1,\cdots,k_n)$.

For example, let us consider the four-dimensional Voronoi code $C_{D_4}(4, a)$ obtained from the D_4 lattice in Section II using (3). Suppose the given index vector is $(2,0,0,1)$. We calculate

$$\begin{aligned} x' &= (5,0,0,1), \\ z &= \tfrac{1}{4}(5, -\tfrac{3}{16}, -\tfrac{11}{32}, \tfrac{15}{32}) \\ &= (1.25, -0.05\cdots, -0.09\cdots, 0.12\cdots) \\ \lambda &= (2,0,0,0,0), \quad \text{by our algorithm [3]}, \\ x &= (-3,0,0,1) - a, \end{aligned}$$

in agreement with the example given in the previous section.

References

[1] J. H. Conway and N. J. A. Sloane, "Fast 4- and 8-dimensional quantizers and decoders," in *IEEE National Telecommun. Conf. Rec.*, vol. 3, pp. F4.2.1 to F4.2.4, 1981.

[2] ——, "Voronoi regions of lattices, second moments of polytopes, and quantization, *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 211–226, 1982.

[3] ——, "Fast quantizing and decoding algorithms for lattice quantizers and codes," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 227–232, 1982.

[4] ——, "The covering radius of the Leech lattice," in *Proc. Royal Soc., Series A*, vol. 380, pp. 261–290, 1982.

[5] H. S. M. Coxeter, *Regular Polytopes*, 3rd ed. New York, Dover, 1973.

[6] G. J. Foschini, R. D. Gitlin, and S. B. Weinstein, "Optimization of two-dimensional signal constellations in the presence of Gaussian noise," *IEEE Trans. Commun.*, vol. COM-22, pp. 28–38, 1974.

[7] K. Hoffman and R. Kunze, *Linear Algebra*. Englewood Cliffs, NJ: Prentice-Hall, 1961.

[8] J. Leech and N. J. A. Sloane, "Sphere packing and error-correcting codes," *Can. J. Math.*, vol. 23, pp. 718–745, 1971.

[9] D. Slepian, "Bounds on communication," *Bell Syst. Tech. J.*, vol. 42, pp. 681–707, 1963.

[10] N. J. A. Sloane, "Binary codes, lattices and sphere packings," in *Combinatorial Surveys*, P. J. Cameron, Ed. New York: Academic, 1977, pp. 117–164.

[11] ——, "Tables of sphere packings and spherical codes," *IEEE Trans. Inform. Theory*, vol. IT-27, pp. 327–338, 1981.