

# New Trellis Codes Based on Lattices and Cosets

A. R. CALDERBANK AND N. J. A. SLOANE, FELLOW, IEEE

**Abstract**—A new technique is proposed for constructing trellis codes, which provides an alternative to Ungerboeck's method of "set partitioning." The new codes use a signal constellation consisting of points from an  $n$ -dimensional lattice  $\Lambda$ , with an equal number of points from each coset of a sublattice  $\Lambda'$ . One part of the input stream drives a generalized convolutional code whose outputs are cosets of  $\Lambda'$ , while the other part selects points from these cosets. Several of the new codes are better than those previously known.

## I. INTRODUCTION

A TRELLIS CODE [3], [4], [12], [25], [28]–[30] is a technique for encoding a data stream into a sequence of real vectors that are transmitted over a noisy channel. Ungerboeck [25] constructed simple trellis codes that provide the same noise immunity as would be obtained by increasing the energy of uncoded transmission by factors ranging from 2 to 4, i.e., have a coding gain of between 3 and 6 dB. Ungerboeck's method is to specify a (conventional) convolutional code, and a rule for mapping the output of the convolutional code into a fixed signal constellation in one- or two-dimensional Euclidean space. In Ungerboeck's construction the output of the convolutional code is matched with the signal point using a technique known as *set partitioning*. This method of construction was also used by Forney *et al.* [12]. Calderbank and Mazo described trellis codes in a way that combined the two steps of the encoder into one in [3]. Further trellis codes were constructed in [4], [28], [29].

The novel feature of the present paper is the use of a generalized convolutional code, whose output symbols are cosets of a lattice.

We do not claim that this technique will replace the construction by set partitioning, to which it is closely related. However it is a useful alternative and one that allows the coding theorist to work with larger constellations and more complicated lattices. As can be seen from the summary in Table XV at the end of the paper, some of the codes obtained are quite powerful.

The general construction is described in Section II, and Section III contains a number of examples. We have not attempted to describe every code that can be obtained by this method. Instead we give a number of different examples in order to illustrate the power of the method. Most of

the examples were found by first choosing a lattice and sublattice, constructing the signal constellation, and then finding the best convolutional code to match the constellation. The code in Example 2b, however, was found by taking the opposite point of view: the convolutional code was chosen first, and then the signal constellation was designed to match it.

The asymptotic performance of these codes is studied in Section IV. The estimate given in Theorem 4 for the average energy of a spherical constellation of points in a lattice may be of independent interest. Also Theorem 9 and Table XIV give the coding gain that is theoretically achievable simply by mapping messages directly into lattice points (without using a trellis code). The final section discusses the codes presented in the text and makes comparisons. Table XV summarizes most of the codes constructed.

Although for logical reasons we give the general construction before the examples, the reader may find it easiest to consider some of the examples when reading the following section. For background information on lattices see [22]–[24], and on groups, rings, etc., see [14], [15], [27].

## II. THE NEW CONSTRUCTION

We shall construct a trellis code that maps a sequence of symbols from an *input alphabet*  $A$  (of size  $a$ ) into a sequence of output points taken from a certain *lattice*  $\Lambda$  in  $n$ -dimensional Euclidean space  $\mathbf{R}^n$ . The set of all possible output points forms the *signal constellation*, which is a finite subset of  $\Lambda$ . (In Example 1a of Section III, which we use to illustrate this section,  $\Lambda$  is the familiar two-dimensional square lattice  $\mathbf{Z}^2$  and the signal constellation is shown in Fig. 3.) We assume that  $A$  is a *ring* (see [15]). Most of our examples are binary codes, i.e.,  $A = \text{GF}(2)$ .

The construction requires three ingredients besides the lattice  $\Lambda$ .

1) A sublattice  $\Lambda' \subseteq \Lambda$ . In all our examples  $\Lambda'$  is obtained from  $\Lambda$  by applying an endomorphism  $\Theta$  of  $\Lambda$  that magnifies distances by a factor of  $\sqrt{m}$  (cf. [1], [2]). Then  $\Lambda$  is an abelian group, and  $\Lambda'$  is a subgroup. We denote the quotient group  $\Lambda/\Lambda'$  by  $R$ . The number of cosets of  $\Lambda'$  in  $\Lambda$ , i.e., the order of  $R$ , or the *index* of  $\Lambda'$  in  $\Lambda$ , is  $m^{n/2}$ . (In Example 1a,  $\Lambda'$  is the sublattice spanned by the vectors  $(2, 2)$  and  $(2, -2)$ . Here  $\Lambda' = \Theta\Lambda$ , where

$$\Theta = \begin{pmatrix} 2 & 2 \\ 2 & -2 \end{pmatrix} \quad (1)$$

Manuscript received February 2, 1985; revised February 27, 1986.

The authors are with the Mathematical Sciences Research Center, AT & T Bell Laboratories, Murray Hill, NJ 07974, USA.

IEEE Log Number 8610921.

is an endomorphism that magnifies distances by  $\sqrt{8}$ . Then  $R = \Lambda/\Lambda'$  has order 8, and representatives for the eight cosets of  $\Lambda'$  in  $\Lambda$  may be found in Table II below.)

A number of examples of  $R$  are listed in Table I.  $\mathbb{Z}^n$  denotes the  $n$ -dimensional integer (or cubic) lattice,  $A_2$  is the familiar two-dimensional hexagonal lattice, and  $D_4$  is the densest lattice sphere packing in four dimensions (see, e.g., [17], [22], [24]).  $\mathbb{Z}_m$  denotes a cyclic group of order  $m$ . (Example 1a corresponds to the second line of the table.)

TABLE I  
THE LATTICES  $\Lambda$ , ENDOMORPHISMS  $\Theta$ , AND QUOTIENTS  $R = \Lambda/\Lambda' = \Lambda/\Theta\Lambda$  USED IN THIS PAPER.  $I$  DENOTES AN IDENTITY MAP. FOR LINE 4 SEE [23], WHERE IT IS ALSO SHOWN THAT  $A_2/\Theta_3 A_2 \cong GF(3)$ ,  $\Theta_3$  BEING AN ENDOMORPHISM OF  $A_2$  THAT MULTIPLIES NORMS BY 3

$\Lambda$	$\Theta$	$R = \Lambda/\Theta\Lambda$
$\mathbb{Z}^2$	$2I$	$\mathbb{Z}_2 \times \mathbb{Z}_2$
$\mathbb{Z}^2$	(1)	$\mathbb{Z}_2 \times \mathbb{Z}_4$
$\mathbb{Z}^2$	$4I$	$\mathbb{Z}_4 \times \mathbb{Z}_4$
$A_2$	$2I$	$GF(4)$
$A_2$	$3I$	$\mathbb{Z}_3 \times \mathbb{Z}_3$
$\mathbb{Z}^4$	$2I$	$\mathbb{Z}_2 \times \mathbb{Z}_2 \times \mathbb{Z}_2 \times \mathbb{Z}_2$
$D_4$	$2I$	$\mathbb{Z}_2 \times \mathbb{Z}_2 \times \mathbb{Z}_2 \times \mathbb{Z}_2$

TABLE II  
COSSET REPRESENTATIVES, LABELS, NORMS, AND MULTIPLICITIES FOR THE COSETS OF  $\Lambda'$  IN  $\Lambda$  FOR EXAMPLE 1a

Coset Representative	Label in Figs. 3, 6, 7	Coset Norm	Coset Multiplicity
(0, 0)	7	0	1
(0, 1)	3	1	1
(0, 2)	5	4	4
(0, -1)	1	1	1
(1, 0)	4	1	1
(1, 1)	0	2	2
(-1, 0)	6	1	1
(1, -1)	2	2	2

We assume that  $\Lambda$  and  $\Lambda'$  are chosen so that the cosets of  $\Lambda'$  in  $\Lambda$  are "compatible" with  $A$ . The precise condition we require is that it is possible to multiply cosets by elements of  $A$ . For all  $a \in A$  and  $r \in R$ , the product  $ar$  must be in  $R$ , and the distributive law

$$a(r_1 + r_2) = ar_1 + ar_2 \tag{2}$$

must hold.<sup>1</sup> (However, we do *not* require the other distributive law,  $(a_1 + a_2)r = a_1r + a_2r$ , to hold. This law does not hold, for instance, in Example 1a.) We shall sometimes write the product as  $ra$  rather than  $ar$ . (This is more convenient when the elements of  $R$  are column vectors, as in the analysis of Example 1a.) In many cases, but not all,  $A$  is a field and  $R$  is a vector space over  $A$ .

The second ingredient is the following.

2) A convolutional code which has  $k_1$  input symbols (taken from the alphabet  $A$ ), and a single output symbol

that is an element of  $R$ . In other words, the outputs from the convolutional code are cosets of  $\Lambda'$  in  $\Lambda$ . The code is described by a generator matrix

$$G = [g_{vk_1}, \dots, g_{v1} | \dots | g_{1k_1}, \dots, g_{11} | g_{0k_1}, \dots, g_{01}] \tag{3}$$

with entries  $g_{ij} \in R$ . The current output symbol depends on the current input block of size  $k_1$  and on the previous  $v$  input blocks. If the input symbols are labeled  $u_{ij}$  as indicated in Fig. 1 (where  $u_{01}, u_{02}, \dots, u_{0k_1}$  is the current input block,  $u_{11}, u_{12}, \dots, u_{1k_1}$  is the previous block, and so on), the current output is

$$r = \sum_{i=0}^v \sum_{j=1}^{k_1} u_{ij} g_{ij}, \tag{4}$$

which by the assumptions in 1) is a well-defined element of  $R$ . The set of possible cosets that occur as outputs of the code will be denoted by  $T$ . In all except Examples 2b and 8,  $T$  is equal to  $R$ .

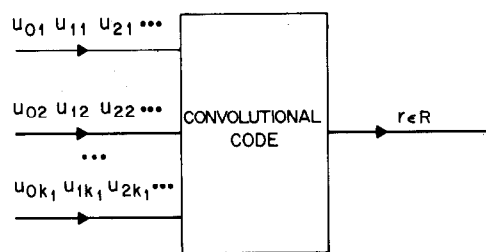


Fig. 1. Convolutional code showing input and output symbols.

At each time instant, or epoch, there are  $a^{k_1}$  possible inputs to the convolutional code and  $|T|$  possible outputs, where  $|T|$  is the number of elements in  $T$ . Therefore the rate of the convolutional code is  $k_1 \log_2 a$  bits/coset, i.e., the fractional rate is

$$\rho_1 := k_1 \log_2 a / \log_2 |T|. \tag{5}$$

(In Example 1a the convolutional code has generator matrix (19),  $k_1 = 2$ ,  $a = 2$ ,  $T = R$ ,  $|T| = 8$ , and the rate  $\rho_1 = 2/3$ .)

The final ingredient follows.

3) The signal constellation, which consists of  $M = |T|a^{k_2}$  points of  $\Lambda$ , partitioned into  $a^{k_2}$  points in each coset in  $T$ .

The complete trellis code is pictured in Fig. 2. The input stream, a sequence of symbols from  $A$ , is divided into blocks of size  $k = k_1 + k_2$ . The first  $k_1$  symbols from each block are fed to the convolutional code, producing an output symbol  $r \in R$ , a coset of  $\Lambda'$  in  $\Lambda$ . The remaining  $k_2$  input symbols are used to select one of the  $a^{k_2}$  points of the constellation that belong to coset  $r$ . The coordinates of this point are then transmitted over the channel. It is the overall rate of the code that is most important. This is  $k \log_2 a$  bits/output symbol, or  $\rho = (k/n) \log_2 a$  bits/dimension. (Example 1a is described in Fig. 4. We have  $k_2 = 2$ ,  $k = k_1 + k_2 = 4$ ,  $n = 2$ , and the overall rate is  $\rho = 2$  bits/dimension.)

Note that increasing  $k_1$  increases the number of states that can be reached from a given state, while increasing  $k_2$

<sup>1</sup>This makes  $\Lambda/\Lambda'$  into a "group with operators" [14, p. 29], [27, p. 138], with elements of the input alphabet  $A$  acting as operators on  $\Lambda/\Lambda'$ .

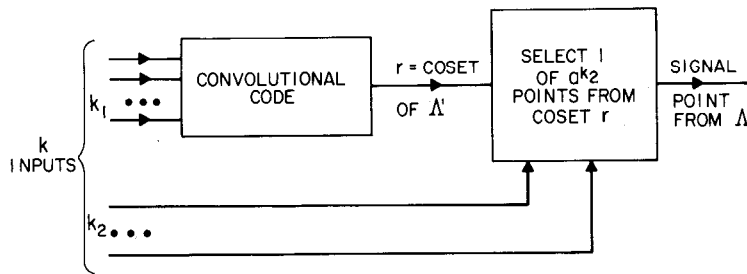


Fig. 2. Overall trellis code.

increases the number of ways to make the transition from one state to another.

*Memory*

The *memory*  $V$  of this code is the minimum number of previous inputs that are required to calculate the current output. From (3),  $V \leq vk_1$ , but since some of the entries in the generator matrix may be zero,  $V$  may be less than  $vk_1$ . The number of states in the encoder is  $a^V$ . (In Example 1a the number of states is  $2^4$ .)

*Trellis Diagram*

As usual it is convenient to describe the convolutional code by a trellis diagram (cf. [19, §9.2]). At each epoch there are  $a^V$  possible states. Each state has  $a^{k_1}$  edges leaving it in the trellis diagram, one edge for each possible input. The edges are labeled with the corresponding output symbols  $r$ , which are cosets of  $\Lambda'$ . A sequence of inputs thus determines a path through the trellis.

It would also be possible to describe our *overall* trellis code by a diagram with multiple edges. The number of states is the same as in the previous diagram, but now each state has  $a^k$  edges leaving it instead of  $a^{k_1}$ . However the simpler diagram is adequate for our analysis.

*Minimal Squared Distance*

The performance of the trellis code is determined by the minimal squared Euclidean distance  $d$  between output sequences corresponding to distinct input sequences (see [24]). The *norm* or *energy* (see [24]) of a point  $x = (x_1, \dots, x_n) \in \mathbf{R}^n$  is

$$N(x) = \sum x_i^2. \tag{6}$$

The norm of a coset  $r \in \Lambda/\Lambda'$ ,  $r = \Lambda' + a$  (say), is defined to be

$$N(r) = \min \{N(x) : x \in \Lambda' + a\}. \tag{7}$$

If  $x \in r$ ,  $y \in s$  ( $r, s \in \Lambda/\Lambda'$ ), then

$$N(x - y) \geq N(r - s). \tag{8}$$

Suppose

$$\begin{aligned} \dots 0, 0, x^{(1)}, x^{(2)}, \dots, x^{(t)}, 0, 0, \dots \quad (x^{(i)} \in \Lambda), \\ \dots 0, 0, y^{(1)}, y^{(2)}, \dots, y^{(t)}, 0, 0, \dots \quad (y^{(i)} \in \Lambda) \end{aligned} \tag{9}$$

are output sequences corresponding to distinct inputs,

where  $x^{(i)} \in r^{(i)}$ ,  $y^{(i)} \in s^{(i)}$  ( $r^{(i)}, s^{(i)} \in \Lambda/\Lambda'$ ) for all  $i$ . Then there are two possibilities: the corresponding sequences of outputs of the convolutional code,

$$\begin{aligned} \dots 0, 0, r^{(1)}, r^{(2)}, \dots, r^{(t)}, 0, 0, \dots \\ \dots 0, 0, s^{(1)}, s^{(2)}, \dots, s^{(t)}, 0, 0, \dots \end{aligned} \tag{10}$$

may be distinct or they may be the same. If they are distinct, then we have two distinct paths through the trellis diagram for the convolutional code that begin and end in the same state. This is called an *error event*. The minimal squared distance between the corresponding outputs is at least

$$\left\{ \min_{\text{error event}} \sum_{i=1}^t N(x^{(i)} - y^{(i)}) \right\},$$

which, using (8), we lower bound by

$$d_1 := \min_{\text{error event}} \left\{ \sum_{i=1}^t N(r^{(i)} - s^{(i)}) \right\}. \tag{11}$$

On the other hand, if the two sequences of cosets coincide, the squared distance between the corresponding outputs is at least

$$d_2 := \min \{N(x) : x \in \Lambda', x \neq 0\}. \tag{12}$$

Let  $\mu$  be the minimal nonzero norm of a vector in  $\Lambda$ . Since the norms in  $\Lambda'$  are  $m$  times the norms in  $\Lambda$ ,

$$d_2 = m\mu. \tag{13}$$

Finally we take

$$d = \min \{d_1, d_2\}. \tag{14}$$

*Figure of Merit*

We assume that the input symbols are independent and identically distributed with a uniform distribution over  $A$ , and that the code is such that all points of the constellation are equally likely. Let the average norm (or energy) of the constellation be  $P$ . Then as our figure of merit for the code we use

$$\frac{d}{P}.$$

We would like to make this as large as possible.

This is admittedly an oversimplification. For example, one could consider imposing an additional constraint on the ratio of peak to average energy. A more realistic figure of merit would also take into account the *path multiplicity*,

which is the number of paths at minimal distance from a given path (cf. [25, eq. (7)]).

The path multiplicity for one of our codes may be obtained from the path multiplicity of the associated convolutional code as follows. For simplicity assume that the signal constellation is large and that the signal points  $x^{(1)}, x^{(2)}, \dots, x^{(t)}$  fall in the interior of the constellation (if there are boundary points then there are fewer minimal distance paths). Given a coset  $r \in \Lambda/\Lambda'$  define the multiplicity  $f(r)$  of the coset  $r$  to be the number of points  $x \in r$  satisfying  $N(x) = N(r)$ . Then there are

$$\prod_{i=1}^t f(r^{(i)} - s^{(i)}) \quad (15)$$

paths  $y^{(1)}, y^{(2)}, \dots, y^{(t)}$ ,  $y^{(i)} \in s^{(i)}$ , at distance  $\sum_{i=1}^t N(r^{(i)} - s^{(i)})$  from  $x^{(1)}, x^{(2)}, \dots, x^{(t)}$ .

Suppose that  $T$  data bits are encoded as  $T/n$   $n$ -dimensional signal points. If the probability that an error event begins is  $p$ , then the probability of correct transmission of the  $T$  data bits is  $(1-p)^{T/n} \approx 1 - (T/n)p$ . For this reason we give the path multiplicity per two-dimensional signal point. (This can also be thought of as the number of channel uses per trellis stage.)

#### Uncoded Transmission

As a basis for comparison we use uncoded transmission at the same rate and in the same dimension, with a constellation in the shape of a cube. The number of points in the uncoded constellation is  $M' = 2^{np} = a^k$ . If  $M'$  is an  $n$ th power, say  $M' = \beta^n$ , we use either the constellation

$$(x_1, \dots, x_n) \quad \text{with } x_i \in \{\pm 1, \pm 3, \dots, \pm(2b-1)\} \quad (16)$$

if  $\beta$  is even,  $\beta = 2b$  say (Fig. 5 below shows an example); or the constellation

$$(x_1, \dots, x_n) \quad \text{with } x_i \in \{0, \pm 2, \pm 4, \dots, \pm 2b\} \quad (17)$$

if  $\beta$  is odd,  $\beta = 2b+1$  say. If  $M'$  is not an  $n$ th power we find the smallest integer  $\beta$  such that  $\beta^n > M'$ , construct the constellation (16) or (17), and remove  $\beta^n - M'$  points of largest norm. The minimal squared distance is  $d^{(u)} = 4$ , and if the average norm is  $P^{(u)}$ , the figure of merit for uncoded transmission is

$$\frac{d^{(u)}}{P^{(u)}}$$

Then the *nominal coding gain* of our code is

$$10 \log_{10} \left( \frac{d}{P} \bigg/ \frac{d^{(u)}}{P^{(u)}} \right). \quad (18)$$

#### The Construction

Most of the codes in Section III were found in the following way. We first choose a lattice  $\Lambda$  and an endomorphism  $\Theta$  that multiplies norms by  $m$ . This determines  $\Lambda' = \Theta\Lambda$ , the vector space  $R = \Lambda/\Lambda'$ , and  $d_2 = m\mu$  (see

(13)). Usually we take  $T = R$ , so  $|T| = |R| = m^{n/2}$ . Knowing  $R$  and the desired rate  $\rho$  we then choose the input alphabet  $A$  (with  $|A| = a$ ) and the parameters  $k_1$  and  $k_2$ .

The second step is to choose a constellation of  $M = m^{n/2} \cdot a^{k_2}$  points in  $\Lambda$  with  $a^{k_2}$  points from each coset of  $\Lambda'$ . We usually found this constellation by the following "greedy algorithm": we choose a center  $c$ , and then, for each coset of  $\Lambda'$ , take the  $a^{k_2}$  points of the coset that are closest to  $c$ . Often there are several ways to choose the points in the outermost shell of the constellation. A good choice for the center  $c$  is a "deep hole" in  $\Lambda$  in the notation of [6], i.e., a point of  $R^n$  maximally distant from  $\Lambda$ . This greedy algorithm is easily implemented on a computer. Our primary goal is to minimize the average signal power. When there are several possible ways to choose the points in the outermost shell, our choice is guided by the ease with which the lattice decoding algorithms described in [8]–[11] can be implemented. For this reason we favor symmetric constellations. (However, we shall not discuss methods for modifying the signal constellations in order to permit differential encoding.)

The third and last step is to find the convolutional code. The requirements are that the memory should be small, while the quantity  $d_1$  (see (11)) should be as large as possible. In view of (14), there is limited advantage in making  $d_1$  larger than  $d_2$  (except perhaps to reduce the number of nearest neighbors, or to construct codes that provide unequal error protection). Just as in the case of ordinary convolutional codes, there seems to be no better method of construction than a systematic search, and that is how the convolutional codes in Section III were mostly found.

### III. EXAMPLES

*1a) A 16-State Binary Code Using the Campopiano-Glazer Constellation that Gains 4.771 dB over Uncoded Transmission at Rate 4 Bits/Two-Dimensional Symbol:* The lattice  $\Lambda$  is the square lattice  $\mathbf{Z}^2$  and is spanned by the vectors (1, 0) and (0, 1). The sublattice  $\Lambda'$  is spanned by the vectors (2, 2) and (2, -2). The lattice  $\Lambda'$  is obtained from  $\Lambda$  by applying an endomorphism  $\Theta = \begin{pmatrix} 2 & 2 \\ 2 & -2 \end{pmatrix}$  that multiplies norms by 8.  $R = \Lambda/\Lambda'$  is isomorphic to  $\mathbf{Z}_2 \times \mathbf{Z}_4$  (see Table I), and  $d_2$ , the minimal nonzero norm in  $\Lambda'$ , is 8. Coset representatives for the eight cosets of  $\Lambda'$  in  $\Lambda$  are listed in Table II, together with the minimal norm of a vector in each coset. We take  $A = \text{GF}(2)$ , and  $k_1 = k_2 = 2$ .

The constellation contains  $M = 8 \cdot 2^2 = 32$  points and is shown in Fig. 3, with the points labeled 0, 1,  $\dots$ , 7 according to the different cosets of  $\Lambda'$ . It is centered at the deep hole (1/2, 1/2) in  $\mathbf{Z}^2$ . We note that the constellation is quite symmetric. The points in cosets 0, 3, 4, and 7 are all equivalent, under rotation through  $\pi/2$ , as are the points in cosets 1, 2, 5, and 6. (The fact that the signal constellation enjoys the same symmetries as the underlying lattice is an advantage in differential encoding; that is, in designing a coding scheme with built-in transparency to 90° phase shifts. Particular methods of differential encod-

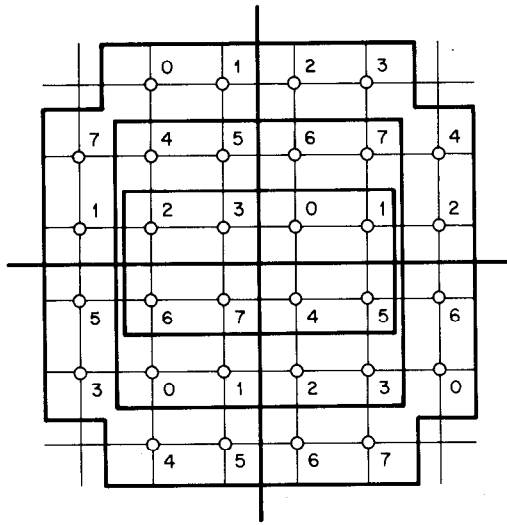


Fig. 3. Signal constellations of size 8, 16 and 32 for Examples 1a, 1b and 1c. Origin is in center, and  $x$  and  $y$  coordinates of points are  $\pm 1/2, \pm 3/2$  or  $\pm 5/2$ .

ing are described in [3], [28].) Note also that the underlying configuration in Fig. 3 is the Campopiano–Glazer 32-point constellation [5]. The average norm of the 32 points is  $P = 5$ . (The differential encoding scheme described in [28] applies to an 8-state code using the Campopiano–Glazer constellation.)

The generator matrix for the convolutional code is

$$G = \left[ \begin{array}{cc|cc|cc} 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 2 & -1 & 0 & 1 & -1 \end{array} \right], \quad (19)$$

where the columns of  $G$  are to be read as cosets of  $\Lambda'$  in  $\Lambda$  according to Table II. If the input symbols are labeled  $u_{ij}$  as in Fig. 1, the output at time  $j$  is  $\Lambda' + Gu(j)$  where

$$u(j)^T = (u_{(j-2)2}, u_{(j-2)1}, u_{(j-1)2}, u_{(j-1)1}, u_{j2}, u_{j1}),$$

and the product  $Gu(j)$  is evaluated in  $R = \mathbb{Z}_2 \times \mathbb{Z}_4$  (see Section II). For this code, and most of the following examples,  $T = R$ . The rate is 2 bits/coset, i.e., the fractional rate is  $\rho_1 = 2/3$ . The encoder is shown in Fig. 4.

The generator matrix in (19), and other generator matrices in this paper, were found by a combination of hand calculation and computer search. In this example the search was simplified by requiring that only one column of  $G$  lie outside the subspace

$$\Lambda^* = \{ \Lambda', \Lambda' + (1, 1), \Lambda' + (1, -1), \Lambda' + (0, 2) \}. \quad (20)$$

We now investigate various properties of this code.

*Lemma 1:* The signal points are used equally often.

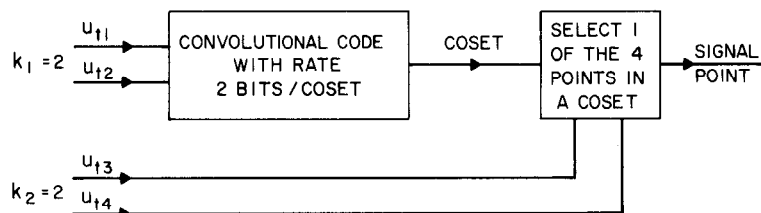


Fig. 4. Encoder for Example 1a.

*Proof:* Observe that the subgroup  $\Lambda^*$  is isomorphic to  $\mathbb{Z}_2 \times \mathbb{Z}_2$ , and that only one column of  $G$  is chosen from outside  $\Lambda^*$ . The coset output at time  $j$ ,  $\Lambda' + Gu(j)$ , is in  $\Lambda^*$  if and only if  $u_{(j-1)1} = 0$ . To prove that the cosets in  $R$  are used equally often we need only prove that the cosets in  $\Lambda^*$  are used equally often. Given  $u(j), v(j)$  with  $u_{(j-1)1} = v_{(j-1)1} = 0$ , let  $x(j) = u(j) \oplus v(j)$  be the modulo 2 sum of  $u(j)$  and  $v(j)$ . Then since  $r + r = 0$  for all  $r \in \Lambda^*$ ,

$$(\Lambda' + Gu(j)) + (\Lambda' + Gv(j)) = \Lambda' + Gx(j), \quad (21)$$

i.e., the map into  $\Lambda^*$  is linear. Since the map is also onto  $\Lambda^*$ , the cosets in  $\Lambda^*$  are used equally often. Points within a coset are equally likely to be selected and so the signal points are used equally often.

*Lemma 2:* If  $u(j)$  and  $v(j)$  are inputs, then the norm of the coset  $(\Lambda' + Gu(j)) - (\Lambda' + Gv(j))$  depends only on the modulo 2 sum  $u(j) \oplus v(j)$ .

*Remark:* This property, which we call *regularity*, makes calculation of the minimal squared distance much easier since it allows us to assume that one of the input sequences is the zero sequence. (Of course, this property automatically holds when  $A$  is a field and  $R$  is a vector space over  $A$ .) This property is shared by the superlinear codes introduced by Marsan *et al.* [20].

*Proof:* We shall show that

$$N\{(\Lambda' + Gu(j)) - (\Lambda' + Gv(j))\} = N\{\Lambda' + G(u(j) \oplus v(j))\}. \quad (22)$$

It is clearly sufficient to prove this in the case when  $u(j)$  and  $v(j)$  have disjoint support. In that case we have

$$\begin{aligned} &(\Lambda' + Gu(j)) - (\Lambda' + Gv(j)) \\ &= (\Lambda' + G(u(j) \oplus v(j))) - (\Lambda' + 2Gv(j)), \end{aligned} \quad (23)$$

and

$$\Lambda' + G(u(j) \oplus v(j)) = \Lambda' + Gu(j) + Gv(j). \quad (24)$$

Note that in (23)

$$\Lambda' + 2Gv(j) = \Lambda' \text{ or } \Lambda' + (0, 2).$$

If  $2Gv(j) \in \Lambda'$  then (22) is true, so we may suppose  $2Gv(j) \in \Lambda' + (0, 2)$ . Hence  $Gv(j) \notin \Lambda^*$  and  $v_{(j-1)1} = 1$ . Since the supports of  $u(j)$  and  $v(j)$  are disjoint, we have  $u_{(j-1)1} = 0$  and  $Gu(j) \in \Lambda^*$ . But, from Table II, adding  $\Lambda' + (0, 2)$  to a coset preserves the norm except when it is added to  $\Lambda'$  or to itself. Therefore, if (22) does not hold, we see from (23), (24) that

$$\Lambda' + Gu(j) + Gv(j) = \Lambda' \text{ or } \Lambda' + (0, 2)$$

and so  $Gu(j) = Gv(j) \in \Lambda^*$ , which is a contradiction.

TABLE III  
THE 16-STATE CODES IN EXAMPLE 1b. ALL HAVE  $d = 6$ .  
THE OVERALL RATE  $\rho = k_2/2 + 1$  BITS/DIMENSION.  
FOR THE UNCODED CONSTELLATIONS  $d^{(u)} = 4$

Coded		Uncoded				Coding Gain (dB)	
$k_2$	$M$	Constellation	$P$	Rate $\rho$	$M'$	$P^{(u)}$	Constellation
0	8	Fig. 3	1.5	1	4	2	$2 \times 2$ array
1	16	Fig. 3	2.5	1.5	8	5	
2	32	Fig. 3	5	2	16	10	Fig. 5
3	64	Fig. 6	10.25	2.5	32	20	Fig. 3
4	128	Fig. 7	20.5	3	64	42	$8 \times 8$ array
5	256		40.69	3.5	128	82	Fig. 7
6	512		81.59	4	256	170	$16 \times 16$ array

To calculate the minimal squared distance  $d_1$  we would normally have to use the trellis diagram of the convolutional code in order to find the minimal norm error event (see (11)). Because this code is regular, however, we may use a much smaller graph  $\Gamma$ . This is a directed graph with 16 vertices labeled by the possible inputs  $(u_{(j-2)2}, u_{(j-2)1}, u_{(j-1)2}, u_{(j-1)1})$ . The vertex labeled  $(u_{(j-2)2}, u_{(j-2)1}, u_{(j-1)2}, u_{(j-1)1})$  is joined to the four vertices  $(u_{(j-1)2}, u_{(j-1)1}, u_{j2}, u_{j1})$ , and the edge is weighted by  $N(\Lambda' + Gu(j))$ . The edge from the vertex labeled  $(0, 0, 0, 0)$  to itself is deleted. Then  $d_1$  is the minimal cost of a path from  $(0, 0, 0, 0)$  to itself. This can be calculated efficiently using Dijkstra's algorithm [16], [21], by modifying Viterbi's decoding algorithm [18, §12.4], [26], or in this case by hand. It is easily seen that  $d_1 = 6$ , and therefore  $d = \min \{d_1, d_2\} = 6$ .

Lemma 3: The convolutional code is not catastrophic.

Proof: We must show that there are no zero cost loops in the graph  $\Gamma$ . The edges weighted by 0 are given below, and the lemma follows because there is no way to construct a loop out of these edges:

- 0010  $\rightarrow$  1001
- 0100  $\rightarrow$  0011
- 0110  $\rightarrow$  1010
- 1000  $\rightarrow$  0010
- 1010  $\rightarrow$  1011
- 1100  $\rightarrow$  0001
- 1110  $\rightarrow$  1000.

The performance of this trellis code is measured against that of uncoded transmission at rate 4 bits/two-dimensional symbol. The uncoded constellation contains  $M' = 16$  points (see (16)) and is displayed in Fig. 5. The average

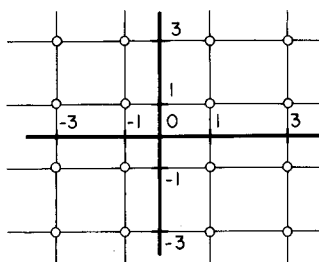


Fig. 5. Uncoded 16-point constellation (called 16-QASK in [25]). Coordinates are  $\pm 1/2, \pm 3/2$ .

norm is 10 (see (58) below), and so the coding gain is

$$10 \log_{10} \left( \frac{6}{5} / \frac{4}{10} \right) = 4.771 \text{ dB.}$$

1b) A Family of 16-State Binary Codes Based on the Convolutional Code of (19): The codes in this family are obtained from Example 1a by varying the parameter  $k_2$  (and the signal constellation) and are shown in Table III. The lattices  $\Lambda, \Lambda'$  and the convolutional code are the same as before, so  $d = 6$  still. For example, the code obtained by taking  $k_2 = 3$  has rate 5 bits/two-dimensional symbol, and the 64-point signal constellation is shown in Fig. 6, where there are now  $2^{k_2} = 8$  points in each coset. The average norm is 10.25. Uncoded transmission at the same rate uses the Campopiano-Glazer constellation, i.e. the points in Fig. 3 (with the labels removed). Thus  $d^{(u)} = 4$ ,

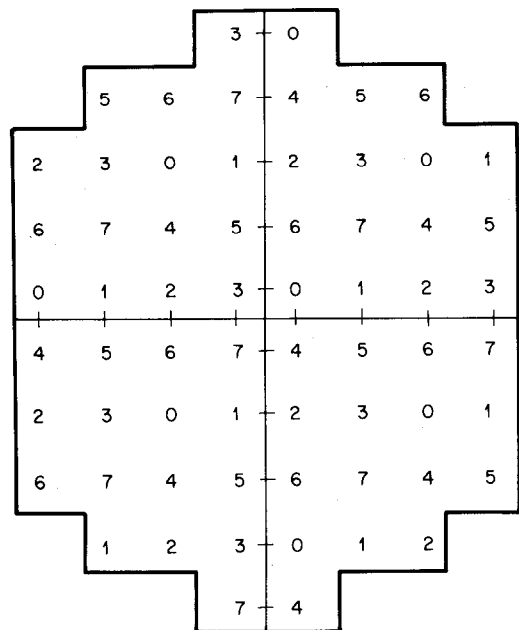


Fig. 6. 64-point signal constellation used in Example 1b. Coordinates are  $\pm 1/2, \pm 3/2, \pm 5/2$  etc.

$P^{(u)} = 20$ , and the coding gain is

$$10 \log_{10} \left( \frac{6}{10.25} / \frac{4}{20} \right) = 4.664 \text{ dB.}$$

This is less than the gain obtained in Example 1a for two

reasons: the constellation of Fig. 6 can be seen by eye to be slightly inferior to that of Fig. 3 (for example the average norm has more than doubled), while the uncoded configuration with which we are comparing it is now very strong (namely Fig. 3 itself).

When  $k_2 = 0$  the 8-point constellation consists of the eight points enclosed by the innermost line in Fig. 3, and the uncoded constellation consists of the four points  $(\pm 1, \pm 1)$ . For  $k_2 = 1$  the 16-point constellation is the  $4 \times 4$  array at the center of Fig. 3, and uncoded transmission uses a  $3 \times 3$  square array with one corner deleted (see (17)). For  $k_2 = 4$  the 128-point constellation is given in Fig. 7.

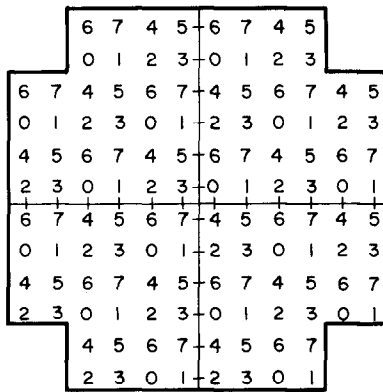


Fig. 7. 128-point signal constellation used in Example 1b. Underlying constellation was given by Campopiano and Glazer [5].

It is shown in Section IV (see (62)) that as  $k_2 \rightarrow \infty$  the coding gain of these codes approaches  $10 \log_{10} \pi = 4.971$  dB.

*1c) Binary Codes Based on Different Convolutional Codes:* These codes are obtained from Example 1a by changing the convolutional code. We specify the convolutional codes by giving generator matrices. As in Example 1a, only one column of the generator matrix is chosen from outside  $\Lambda^*$ . It follows that each code is regular and that the signal points are used equally often. The minimal squared distance  $d_1$  may be calculated (by computer) as in Example 1a. The generator matrix

$$\begin{bmatrix} 1 & 0 & 1 & 1 & 0 \\ 1 & 2 & 0 & 1 & 2 \end{bmatrix} \quad (25)$$

defines an 8-state code with  $d_1 = 5$ ,

$$\begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 2 & -1 & 0 & 1 & -1 & 1 & -1 \end{bmatrix} \quad (26)$$

defines a 64-state code with  $d_1 = 7$ , and

$$\begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\ 2 & 1 & 1 & 1 & 0 & 1 & 0 & 3 & 1 & 3 \end{bmatrix} \quad (27)$$

defines a 256-state code with  $d_1 = 8$ . The computer also verified that these codes (and in fact all the convolutional codes mentioned in Section III) are noncatastrophic.

By varying  $k_2$ , but using the same constellations as in Table III, we obtain the codes shown in Table IV. (The generator matrix

$$\begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 2 & -1 & 1 & 2 & 1 & 2 \end{bmatrix}$$

defines a 32-state code with  $d_1 = 6$ . This has the same value of  $d_1$  as (19) but the multiplicity has been reduced from 56 to 16.)

*2a) A 4-State Binary Code Using the Campopiano-Glazer Constellation that Gains 3.010 dB over Uncoded Transmission at Rate 4 Bits/Two-Dimensional Symbol:* Again the lattice  $\Lambda$  is  $\mathbf{Z}^2$ , but now we take  $\Lambda' = 2\mathbf{Z}^2$ , so that  $\Lambda/\Lambda' \cong \mathbf{Z}_2 \times \mathbf{Z}_2$  and  $m$  (the magnification factor) is 4. The cosets of  $\Lambda'$  are described in Table V and the 32-point constellation in Fig. 8. Again this is the Campopiano-Glazer configuration but with a different labeling. The average norm is 5. We take  $A = \text{GF}(2)$ ,  $k_1 = 1$  and  $k_2 = 3$ ; the overall rate will again be 4 bits/two-dimensional signal, or  $\rho = 2$  bits/dimension.

TABLE V  
COSSET REPRESENTATIVES, LABELS, NORMS,  
AND MULTIPLICITIES FOR  $\mathbf{Z}^2/2\mathbf{Z}^2$

Coset Representative	Label in Fig. 8	Coset Norm	Coset Multiplicity
(0, 0)	0	0	1
(0, 1)	1	1	2
(1, 0)	2	1	2
(1, 1)	3	2	4

Notice that the norm of a coset is the Hamming weight of the coset representative. So we may use a conventional binary convolutional code, and the minimal squared distance  $d_1$  (see (11)) is equal to the free distance of the code. We use the familiar rate 1/2 noncatastrophic code (code CC1 of [19, p. 200]) with generator matrix

$$\begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad (28)$$

TABLE IV  
THE CODES CONSTRUCTED IN EXAMPLE 1c

$k_2$	Rate $\rho$	Uncoded $d/P$	Coded $P$	Coding Gain If			
				$d = 5$ 8 States	$d = 6$ 16 States	$d = 7$ 64 States	$d = 8$ 256 States
0	1	4/2	1.5	2.218	3.010	3.680	4.260
1	1.5	4/5	2.5	3.979	4.771	5.441	6.021
2	2	4/10	5.0	3.979	4.771	5.441	6.021
3	2.5	4/20	10.25	3.872	4.664	5.333	5.913
4	3	4/42	20.5	4.084	4.876	5.545	6.125
5	3.5	4/82	40.687	4.013	4.804	5.474	6.054
6	4	4/170	81.594	4.157	4.949	5.618	6.198
$\infty$	$\infty$			4.180	4.971	5.641	6.221

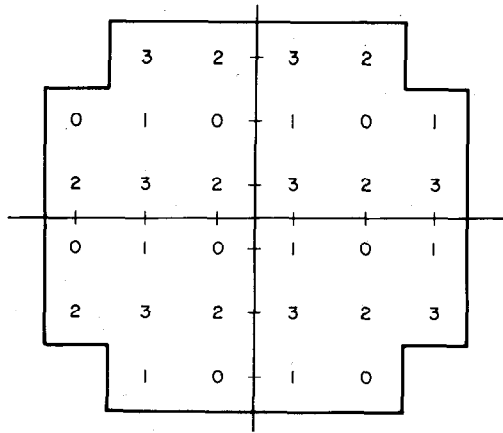


Fig. 8. 32-point constellation in  $Z^2/2Z^2$  used in Example 2a.

Then  $d_1 = 5$ ,  $d_2 = 4$  and  $d = \min\{5, 4\} = 4$ . Uncoded transmission at this rate was described in Example 1a, and  $d^{(u)} = 4$ ,  $P^{(u)} = 10$ . The coding gain is

$$10 \log_{10} \left( \frac{4}{5} / \frac{4}{10} \right) = 3.010 \text{ dB.}$$

As in Example 1b we could now obtain a family of codes by varying  $k_2$ , but we leave this to the reader. In the limit as  $k_2$  increases, it follows from Theorem 8 below that the gain approaches  $10 \log_{10} (2\pi/3) = 3.211$  dB.

2b) *A 4-State Code Using a 32-Point Constellation Not Derived from a Lattice that Gains 3.200 dB over Uncoded Transmission at Rate 4 Bits/Two-Dimensional Symbol:* This example differs from the others in that we begin with a convolutional code and then find a constellation to match it. Consider the rate 1/2 convolutional code (28) used in Example 2a. The outputs are assigned the weights listed in Table VI. The state diagram is shown in Fig. 9, where the

TABLE VI  
WEIGHTS AND MULTIPLICITIES FOR CONVOLUTIONAL CODE  
USED IN EXAMPLE 2b

Output	Weight	Multiplicity
(0,0)	0	1
(1,1)	A	3
(0,1)	B	2
(1,0)	C	1

vertices are labeled by the two previous inputs and the edges are labeled by the weight of the current output. Now we think of the output as picking out a set of eight signal points. We are interested in regular codes so we require that the distance between the sets corresponding to outputs  $u$  and  $v$  depends only on the modulo 2 sum  $u \oplus v$ . A 32-point constellation with this property is shown in Fig. 10. Notice that  $A = 3$ ,  $B = 2$ , and  $C = 1$ . The distance  $d_2$  between points in the same subset is at least 8, so  $d = 8$ , and  $P = (1/8)(61 + 9\sqrt{3}) = 9.573$ . The coding gain is therefore

$$10 \log_{10} \left( \frac{8}{9.573} / \frac{4}{10} \right) = 3.200 \text{ dB,}$$

which is slightly better than Example 2a.

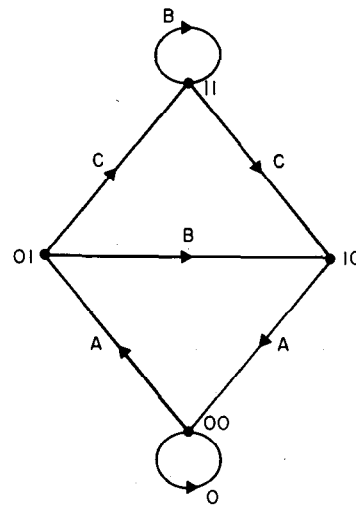


Fig. 9. State diagram of convolutional code used in Example 2b.

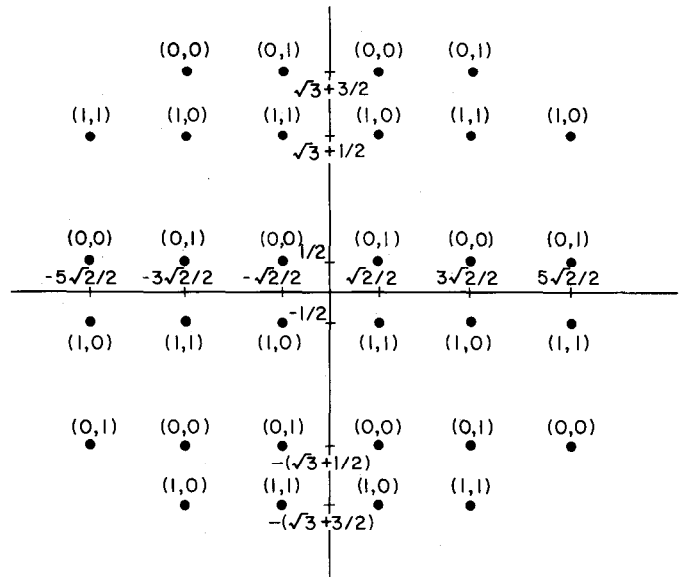


Fig. 10. 32-point constellation used in Example 2b.

3a) *A 16-State Code with Input Alphabet  $A = Z_4$  that Uses a 64-Point Signal Constellation and Gains 4.664 dB over Uncoded Transmission at Rate 4 Bits/Two-Dimensional Symbol:* Again  $\Lambda = Z^2$ , but now  $\Lambda' = 4Z^2$ , so that  $\Lambda/\Lambda' \cong Z_4 \times Z_4$  and  $m = 16$ . The cosets are described in Table VII, and the 64-point constellation (with 4 points in each coset) is given in Fig. 11. The average norm is 10.25. We take  $A = Z_4$ ,  $a = 4$ ,  $k_1 = 1$ ,  $k_2 = 1$ . The rate of the convolutional code (Fig. 12) is 2 bits/coset, i.e.,  $\rho_1 = 1/2$ , and the overall rate is 4 bits/two-dimensional signal.

The generator matrix of the convolutional code is

$$G = \begin{bmatrix} 1 & 1 & 1 \\ 2 & 1 & 2 \end{bmatrix}, \quad (29)$$

where the columns of  $G$  are to be read as cosets of  $\Lambda'$  in  $\Lambda$ . The coset output at time  $j$  is  $\Lambda' + Gu(j)$  where  $u(j) = (u_{(j-2)1}, u_{(j-1)1}, u_{j1})^T$ . Given  $u(j), v(j)$  let  $x(j) = u(j) \oplus v(j)$  be the modulo 4 sum of  $u(j)$  and  $v(j)$ . Then

$$(\Lambda' + Gu(j)) + (\Lambda' + Gv(j)) = \Lambda' + Gx(j),$$



TABLE VII  
COSSET REPRESENTATIVES, LABELS, NORMS, AND MULTIPLICITIES FOR  $Z^2/4Z^2$

Coset Representative	Label in Fig. 11	Coset Norm	Coset Multiplicity	Coset Representative	Label in Fig. 11	Coset Norm	Coset Multiplicity
(0, 0)	3	0	1	(0, 2)	11	4	2
(1, 0)	0	1	1	(1, 2)	8	5	2
(2, 0)	1	4	2	(2, 2)	9	8	4
(-1, 0)	2	1	1	(-1, 2)	10	5	2
(0, -1)	7	1	1	(0, 1)	15	1	1
(1, -1)	4	2	1	(1, 1)	12	2	1
(2, -1)	5	5	2	(2, 1)	13	5	2
(-1, -1)	6	2	1	(-1, 1)	14	2	1

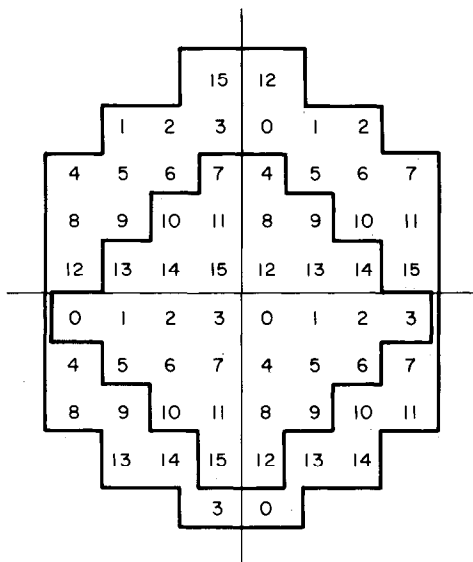


Fig. 11. Constellations of size 32 and 64 for  $Z^2/4Z^2$ .

from which it follows that the cosets of  $\Lambda'$  in  $\Lambda$  are used equally often. Also the code is regular in the sense that the coset  $(\Lambda' + Gu(j)) - (\Lambda' + Gv(j))$  depends only on the modulo 4 difference  $u(j) \ominus v(j)$ . This means that when calculating the minimal squared distance  $d_1$  we may assume that one of the input sequences is the zero sequence. The directed graph used to analyze this code is a little different from the one used in Example 1a: the 16 vertices are labeled with the possible inputs  $(u_{(j-2)1}, u_{(j-1)1})$  and the vertex  $(u_{(j-2)1}, u_{(j-1)1})$  is joined to the four vertices labeled  $(u_{(j-2)1}, u_{(j-1)1}, u_{j1})$ . It is easily seen, using the method of Example 1a, that  $d_1 = 12$ . (A little analysis shows that this is the best possible for a 16-state code). Then  $d = \min \{12, 16\} = 12$ . The zero-weighted edges in

this graph are

$$\begin{aligned} 1 \quad 0 &\rightarrow 0 \quad -1 \\ 2 \quad 0 &\rightarrow 0 \quad 2 \\ -1 \quad 0 &\rightarrow 0 \quad 1 \end{aligned}$$

and since there is no way to construct a loop from these edges, the code is noncatastrophic. The coding gain is

$$10 \log_{10} \left( \frac{12}{10.25} / \frac{4}{10} \right) = 4.664 \text{ dB.}$$

3b) 4, 8, 64, and 128-State Codes with Input Alphabet  $A = Z_4$ , and Rate 4 Bits/Two-Dimensional Symbol: The only difference between these codes and Example 3a is the choice of convolutional code. The generator matrix

$$\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \quad (30)$$

defines a 4-state code, and the trellis diagram is shown in Fig. 13. Then  $d_1 = 8$  (which is the best possible for a

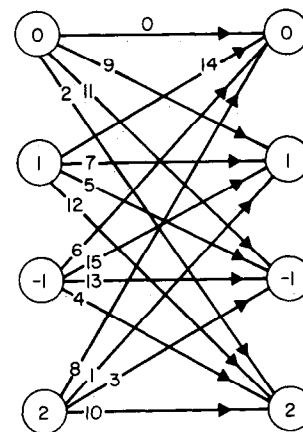


Fig. 13. Trellis diagram for convolutional code of (30). Edges are labeled with cosets of  $Z^2/4Z^2$ .

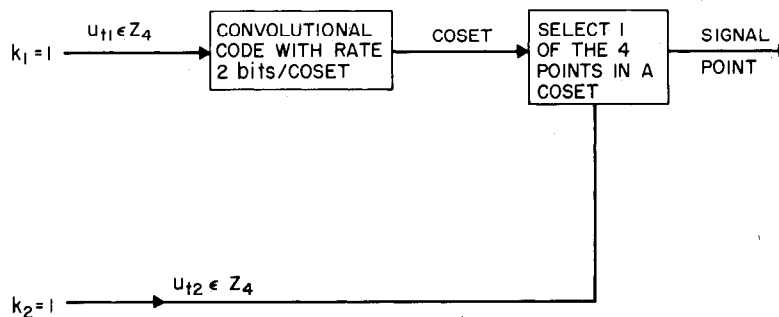


Fig. 12. Convolutional code used in Example 3a.

4-state code) and  $d = 8$ . The coding gain over uncoded transmission at the same rate is

$$10 \log_{10} \left( \frac{8}{10.25} / \frac{4}{10} \right) = 2.903 \text{ dB.}$$

Similarly,

$$\begin{bmatrix} 0 & 1 & 1 \\ 2 & 1 & 2 \end{bmatrix} \quad (31)$$

defines a 16-state code with  $d_1 = d = 11$ . However, since  $1 \binom{0}{2} = 3 \binom{0}{2}$  and  $\binom{0}{0} = 2 \binom{0}{0}$ , only the value modulo 2 of the final symbol matters, and the trellis diagram collapses to eight states. The coding gain is 4.286 dB.

The generator matrix

$$\begin{bmatrix} 1 & 1 & 0 & 1 \\ 2 & 1 & 2 & 2 \end{bmatrix} \quad (32)$$

defines a 64-state code. The methods applied to Example 3a show that cosets are selected equally often and that the code is regular. Here  $d_1 = 14$ , which is found using a directed graph on 64 vertices. Then  $d = 14$ , and the coding gain is 5.333 dB.

Similarly the generator matrix

$$\begin{bmatrix} 0 & 1 & 1 & 1 & 2 \\ 2 & 2 & 0 & 0 & 1 \end{bmatrix} \quad (33)$$

defines a 128-state code with  $d_1 = d = 16$  and a coding gain of 5.913 dB.

3c) Codes with 4, 8, 16, 64, and 128 States, Input Alphabet  $Z_4$ , and Rates 3 or 5 Bits/Two-Dimensional Symbol: These examples are obtained from Examples 3a and 3b by varying the parameter  $k_2$ . The encoder is shown in Fig. 14. For  $k_2 = 1/2$  the rate is 3 bits/two-dimensional symbol. The 32-point constellation is defined by the inner line in Fig. 11, and has average norm 5.266. Uncoded transmission at this rate was described in Example 1b. The codes with 4, 8, 16, 64, and 128 states gain 2.785, 4.168, 4.546, 5.216, and 5.795 dB, respectively.

Similarly, with  $k_2 = 3/2$  we use the 128-point constellation shown in Fig. 15 with  $P = 20.5$  and obtain codes with rate 5 bits/two-dimensional symbol. The coding gains are the same as in Examples 3a and 3b.

4) An 8-State Binary Code Using 32 Signal Points Taken from the Hexagonal Lattice  $A_2$  that Gains 3.529 dB over Uncoded Transmission at Rate 4 Bits/Two-Dimensional Symbol: The lattice  $\Lambda$  is the hexagonal lattice  $A_2$  spanned by the vectors  $(1, 0)$  and  $(-1/2, \sqrt{3}/2)$ . (Decoding al-

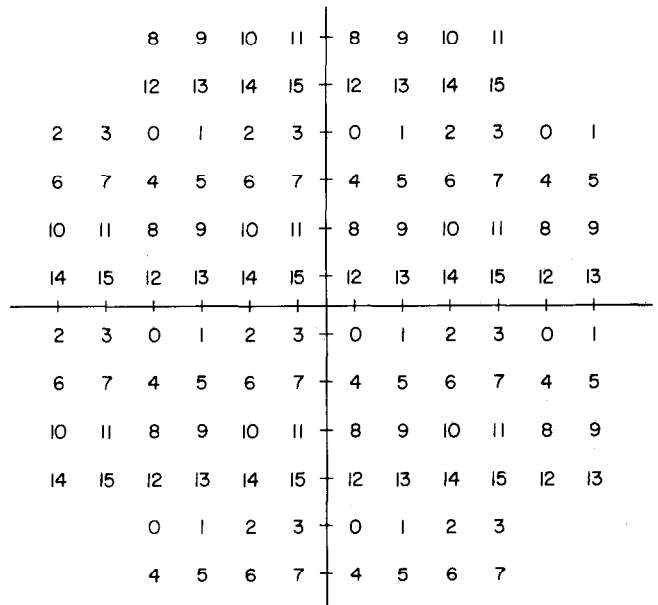


Fig. 15. 128-point constellation in  $Z^2/4Z^2$ .

gorithms for this lattice may be found in [8], [10].) The sublattice  $\Lambda'$  is  $2A_2$ . The quotient  $R = \Lambda/\Lambda' \cong Z_2 \times Z_2$  (actually  $R$  is the field  $GF(4)$ , but we make no use of the multiplicative structure). The map

$$(\xi_1, \xi_2) \in Z^2 \Leftrightarrow \left( \xi_1 - \frac{1}{2}\xi_2, \frac{\sqrt{3}}{2}\xi_2 \right) \in A_2$$

enables us to label the points of  $A_2$  with pairs of integers. The four cosets of  $2A_2$  in  $A_2$  can then be represented by the binary vectors  $(\xi_1, \xi_2) = (0, 0), (0, 1), (1, 0), (1, 1)$ , where  $(\xi_1, \xi_2)$  denotes the coset

$$\left( \xi_1 - \frac{1}{2}\xi_2, \frac{\sqrt{3}}{2}\xi_2 \right) + \Lambda'. \quad (34)$$

The 32-point constellation is shown in Fig. 16, where the points are labeled  $(0, 0), (0, 1), (1, 0), (1, 1)$  corresponding to the different cosets. To minimize the average norm, the constellation is centered at the midpoint of an edge. The average norm  $P$  is  $71/16$ . Note that every coset has norm 1.

The convolutional code is an 8-state code with rate  $\rho_1 = 1/2$  and generator matrix

$$\begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}. \quad (35)$$

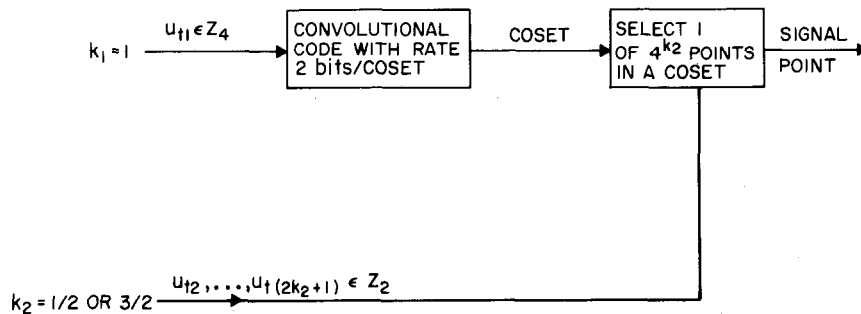


Fig. 14. Convolutional code used in Example 3c.

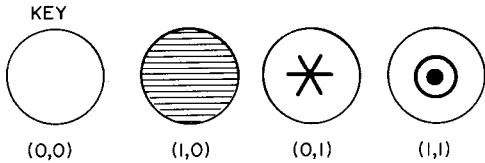
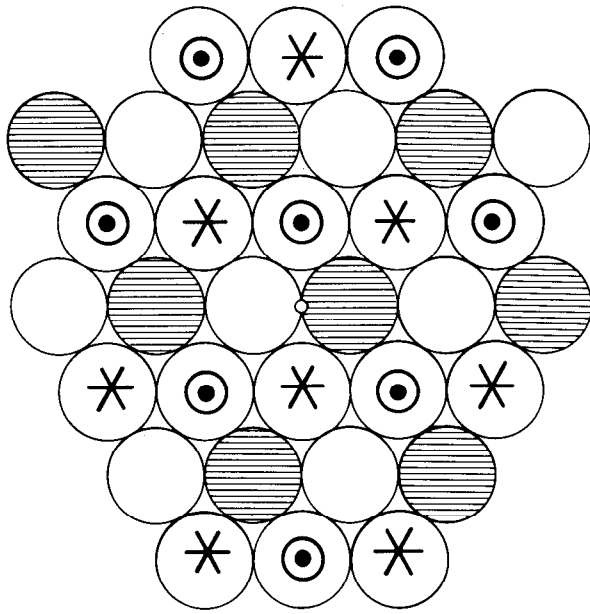


Fig. 16. 32-point constellation in  $A_2/2A_2$ . Distinctive symbols have been used for cosets in order to emphasize symmetry among cosets.

We may regard this code as a conventional binary convolutional code with a nonstandard weighting of outputs. It is easily seen that the minimal squared distance  $d_1 = 4$ .

Thus  $d = 4$ . The coding gain is

$$10 \log_{10} \left( \frac{64}{71} / \frac{4}{10} \right) = 3.529 \text{ dB.}$$

5) A 9-State and a 27-State Code with Input Alphabet  $A = GF(3)$  and Rate  $4 \log_2 3 = 6.340$  Bits/Two-Dimensional Symbol: We take  $\Lambda = A_2, \Lambda' = 3A_2$ , so that  $\Lambda/\Lambda' \cong \mathbf{Z}_3 \times \mathbf{Z}_3$  and  $m = 9$ . The cosets may be represented by ternary vectors  $(\xi_1, \xi_2)$  with  $\xi_i \in \mathbf{Z}_3$ , again using (34) (see Table VIII). The  $3^5 = 243$  point constellation is shown in

TABLE VIII  
COSSET NAMES, LABELS, AND NORMS FOR  $A_2/3A_2$

Coset $(\xi_1, \xi_2)$	Label in Fig. 17	Coset Norm
(0,0)	0	0
(1,0)	1	1
(1,1)	2	1
(0,1)	3	1
(-1,0)	4	1
(-1,-1)	5	1
(0,-1)	6	1
(1,-1)	7	3
(-1,1)	8	3

Fig. 17. This is centered at the midpoint of a triangle (a deep hole) in  $A_2$ , and  $P = 33.519$ . We take  $A = \mathbf{Z}_3$  ( $= GF(3)$ ),  $k_1 = 1$  and  $k_2 = 3$ ; the overall rate is  $4 \log_2 3 = 6.340$  bits/two-dimensional symbol.

The convolutional code is a 9-state code with rate  $\rho_1 = 1/2$  and generator matrix

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -1 & 1 \end{bmatrix}. \tag{36}$$

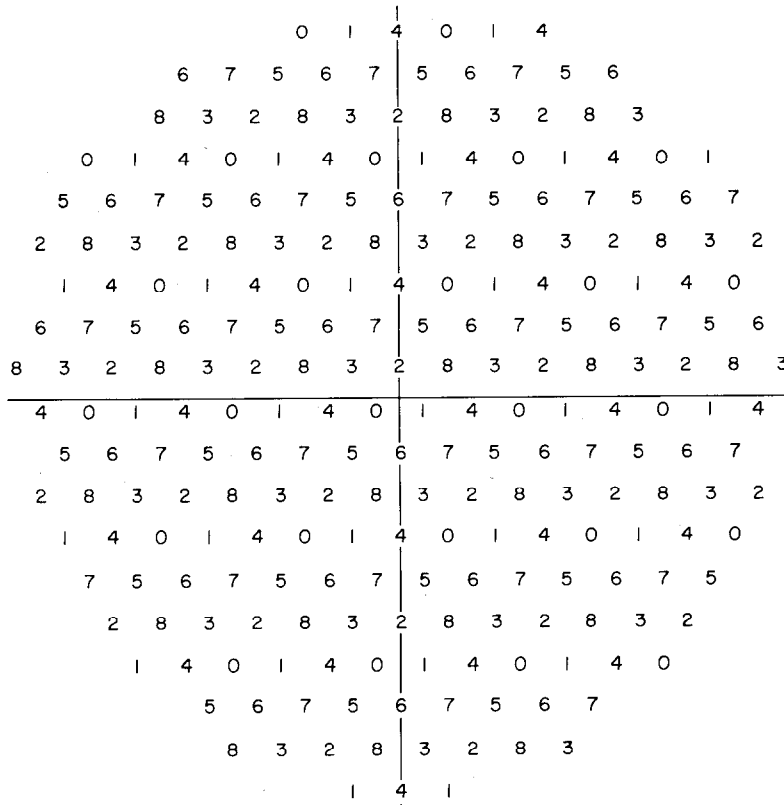


Fig. 17. 243-point constellation in  $A_2/3A_2$ .

If  $\phi(u)$  denotes the output sequence corresponding to the ternary input vector  $u$  then

$$\phi(u) \oplus \phi(v) = \phi(u \oplus v),$$

so this is a conventional convolutional code over GF(3) with a nonstandard weighting of outputs (given by Table VIII). The code has  $d_1 = 7$  (which is the best possible for a 9-state code). Since  $d_2 = 9$  we have  $d = 7$ . Similarly the generator matrix

$$\begin{bmatrix} 1 & | & 1 & | & 1 & | & -1 \\ 0 & | & 1 & | & -1 & | & 0 \end{bmatrix} \quad (37)$$

defines a 27-state code with  $d_1 = d = 8$ .

Uncoded transmission at this rate uses the 81-point constellation shown in Fig. 18, and  $d^{(u)} = 4$ ,  $P^{(u)} = 1390/47 = 51.481$ . The coding gain of the 9-state code is 4.294 dB and that of the 27-state code is 4.874 dB.

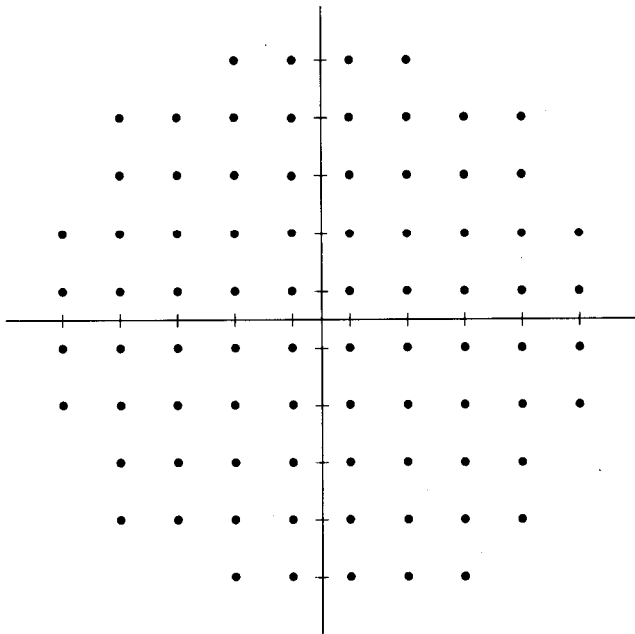


Fig. 18. Uncoded 81-point constellation, obtained by omitting 19 points of greatest norm from  $10 \times 10$  square array (see (17)).

As usual by increasing  $k_2$  we obtain two infinite sequences of codes. The lattice  $A_2$  has density  $\pi/\sqrt{12}$  (see [12]), so from Theorem 8 the limiting gain of these codes is 4.505 dB for the 9-state codes and 5.085 dB for the 27-state codes.

6) *A Family of 8-State Binary Codes Using Signal Points Taken from the Four-Dimensional Odd Integer Lattice  $(2Z + 1)^4$* : This family of codes is described more fully in [4]. (See also [12] for a different  $k = 8$  code.)  $\Lambda = Z^4$ ,  $\Lambda' = 2Z^4$ ,  $R = \Lambda/\Lambda' \cong Z_2 \times Z_2 \times Z_2 \times Z_2$ , and  $m = 4$ . The map

$$\xi = (\xi_1, \xi_2, \xi_3, \xi_4) \rightarrow \Lambda' + \xi$$

allows us to represent cosets of  $\Lambda'$  by binary 4-tuples. The norm of a coset is the Hamming weight of the coset representative. To minimize the average energy, the signal constellations are all centered at the deep hole  $(1/2, 1/2, 1/2, 1/2)$ . If we multiply by 2 in order to clear

fractions, every signal point is in  $(2Z + 1)^4$ . This multiplies norms by 4, so now  $d_2 = 4m = 16$ . We take  $A = GF(2)$ ,  $k_1 = 3$ , and let  $k_2$  vary, the cases  $k_2 = 5$  or 9 being of greatest interest (see [4]).

The convolutional code is an 8-state code with rate  $\rho_1 = 3/4$  and generator matrix

$$\begin{bmatrix} 0 & | & 0 & 0 & 0 & | & 1 & 0 & 0 \\ 0 & | & 0 & 1 & 1 & | & 1 & 1 & 0 \\ 1 & | & 0 & 1 & 0 & | & 1 & 0 & 1 \\ 1 & | & 0 & 0 & 0 & | & 1 & 1 & 1 \end{bmatrix} \quad (38)$$

This may be regarded as a conventional binary convolutional code (it is described by Lin and Costello [12, fig. 10.3, p. 292]) with a nonstandard weighting of outputs (four times the Hamming weight). It has free distance 4, so  $d_1 = 4 \times 4 = 16 = d$ .

The points of the lattice  $(2Z + 1)^4$  lie in shells around the origin consisting of 16 vectors of norm 4, 64 vectors of norm 12, and so on. The  $2^{k+1}$  signal points for coded transmission are obtained by taking all points of energy 4, 12, 20, ... and just enough points of a final shell to bring the total number up to  $2^{k+1}$ . The signal constellation is partitioned into 16 cosets (according to the congruence class of the entries modulo 4) and each coset contains  $2^{k-3}$  signal points. The parameters of the first codes in this family are listed in Table IX.  $\Lambda$  has density  $\Delta = \pi^2/32$ , so from Theorem 8 the gain approaches  $10 \log_{10} \pi = 4.971$  dB as  $k_2 \rightarrow \infty$ . This was also established (by direct calculation) in [4].

TABLE IX  
CODES CONSTRUCTED IN [4] FROM THE LATTICE  $(2Z + 1)^4$

$k = k_1 + k_2$	Rate (bits/dimension)	$M$	$d/P$	$d^{(u)}/P^{(u)}$	Coding Gain (dB)
8	2	$2^9$	16/27	4/20	4.717
12	3	$2^{13}$	16/108.625	4/84	4.904

*Remark:* We may regard transmission of a four-dimensional signal as one use of the channel since a four-dimensional signal space can be realized by using two space-orthogonal electric field polarizations to communicate on the same carrier frequency. Of course it is also possible to regard each four-dimensional symbol as two consecutive two-dimensional symbols.

7a) *A 64-State Binary Code that Gains 6.131 dB over Uncoded Transmission at Rate 8 Bits/Four-Dimensional Symbol*: The lattice  $\Lambda$  is the four-dimensional lattice  $D_4$  which is spanned by the vectors  $(1, 1, 0, 0)$ ,  $(0, 1, -1, 0)$ ,  $(0, 0, 1, -1)$ , and  $(0, 0, 0, 2)$  [17]. The sublattice  $\Lambda'$  is  $2D_4$ , so  $R = \Lambda/\Lambda' \cong Z_2 \times Z_2 \times Z_2 \times Z_2$  and  $m = 4$ .  $\Lambda$  has minimal norm 2, so  $\Lambda'$  has minimal norm  $d_2 = 2m = 8$ . The map

$$\xi = (\xi_1, \xi_2, \xi_3, \xi_4) \rightarrow \Lambda' + \xi \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 2 \end{bmatrix} \quad (39)$$

allows us to identify binary 4-tuples with cosets of  $\Lambda'$ . The

cosets and coset norms are listed in Table X. The third column of the table gives an example of a vector of  $D_4$  in each coset. A bar indicates a negative number. These vectors are in the standard version of  $D_4$  given at the beginning of this paragraph, in which the origin of the coordinates is at a lattice point. To get a signal constellation of minimal energy, as usual we shift the origin. In this case we shift the origin to the point  $(1/2, 1/2, 1/2, 1/2)$ , a deep hole in  $D_4$  [7], and multiply the coordinates by 2 to clear fractions. For example, in the new coordinates there are eight lattice points around the origin, with coordinates  $(\pm 1, \pm 1, \pm 1, \pm 1)$ , with an even number of minus signs. The fourth column of Table X gives an example of a lattice point in each coset of  $2D_4$  in the new coordinates.

TABLE X  
COSSET NAMES, NORMS, AND MULTIPLICITIES FOR  $D_4/2D_4$ . THE LAST TWO COLUMNS GIVE EXAMPLES OF POINTS IN THE COSETS USING COORDINATES IN WHICH THE ORIGIN IS EITHER AT A LATTICE POINT (COLUMN 3) OR AT A DEEP HOLE (COLUMN 4)

Coset Name ( $\xi_1 \xi_2 \xi_3 \xi_4$ )	Coset Norm	Coset Multiplicity	Vector (Origin)	Vector (Hole)
0000	0	1	0000	$\overline{1111}$
0001	4	8	0002	$3\overline{111}$
1010	4	8	$\overline{1111}$	$\overline{3111}$
1011	4	8	1111	1111
0010	2	2	00 $\overline{11}$	$3\overline{111}$
0011	2	2	0011	$\overline{11}11$
1001	2	2	$\overline{11}00$	$113\overline{1}$
1000	2	2	1100	$11\overline{11}$
0110	2	2	010 $\overline{1}$	$31\overline{11}$
0111	2	2	0101	$\overline{1111}$
1101	2	2	$\overline{1010}$	$\overline{3111}$
1100	2	2	1010	$1\overline{111}$
0100	2	2	0 $\overline{110}$	$31\overline{11}$
0101	2	2	0110	$\overline{1111}$
1111	2	2	$\overline{1001}$	$\overline{311}1$
1110	2	2	1001	$1\overline{11}1$

The original version of  $D_4$  consists of the points with coordinates

$$(x_1, x_2, x_3, x_4), x_i \text{ integers, } x_1 + x_2 + x_3 + x_4 \text{ even,} \quad (40)$$

while the shifted version consists of the points

$$\frac{1}{2}(x_1, x_2, x_3, x_4), x_i \text{ odd integers,} \\ x_1 + x_2 + x_3 + x_4 \equiv 0 \pmod{4}. \quad (41)$$

The 512-point signal constellation is described using the new coordinates in Table XI. The second line, for example, refers to the  $4 \cdot 2^3$  points obtained from  $(\pm 3, \pm 1, \pm 1, \pm 1)$  by taking all possible permutations and all sign combinations that satisfy (41). There are 32 points in each coset of  $2D_4$ , and the average norm is  $P = 39$ . On this scale the minimal nonzero norm in a coset is  $d_2 = 32$ .

TABLE XI  
512-POINT CONSTELLATION IN  $D_4/2D_4$

Norm	Vector	Number
4	1111	$1 \cdot 2^3 = 8$
12	3111	$4 \cdot 2^3 = 32$
20	3311	$6 \cdot 2^3 = 48$
28	5111	$4 \cdot 2^3 = 32$
28	3331	$4 \cdot 2^3 = 32$
36	5311	$12 \cdot 2^3 = 96$
36	3333	$2^3 = 8$
44	5331	$12 \cdot 2^3 = 96$
52	5511	$6 \cdot 2^3 = 48$
52	5333	$4 \cdot 2^3 = 32$
52	7111	16
60	5531	64

We take  $A = GF(2)$ ,  $k_1 = 3$ ,  $k_2 = 5$ ; the convolutional code has rate  $\rho_1 = 3/4$ , and the overall rate is 8 bits/four-dimensional symbol.

The generator matrix

$$\left[ \begin{array}{c|cccc|cccc} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \end{array} \right] \quad (42)$$

defines a 64-state convolutional code. If  $\phi(u)$  denotes the output sequence corresponding to the binary input vector  $u$  then

$$\phi(u) \oplus \phi(v) = \phi(u \oplus v), \quad (43)$$

where  $\oplus$  denotes modulo 2 addition. We may therefore regard the code defined by (42) as a conventional binary convolutional code with a nonstandard weighting of outputs (given by Table X). It can be shown that  $d_1 = 4 \times 8 = 32$ , so  $d = d_1 = d_2 = 32$ . Uncoded transmission at this rate uses the constellation (16) with  $n = 4$ ,  $b = 2$ , and  $d^{(u)} = 4$ ,  $P^{(u)} = 20$ . The coding gain is

$$10 \log_{10} \left( \frac{32}{39} \bigg/ \frac{4}{20} \right) = 6.131 \text{ dB.}$$

7b) Other Four-Dimensional Codes: As in Examples 1a–1c we obtain further codes from Example 7a by changing either the convolutional code or the parameter  $k_2$ . The generator matrix

$$\left[ \begin{array}{c|cccc|cccc} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \end{array} \right] \quad (44)$$

defines a 16-state code with  $d_1 = 4 \times 6$ ; and when used with the signal constellation of Table XI produces a code of rate 8 bits/four-dimensional symbol with a coding gain of 4.881 dB.

If we increase  $k_2$  from 5 to 9 a  $2^{13}$ -point signal constellation is required. This may be obtained by extending Table XI, and has  $P = 153.75$  (we do not list the code here). Combining this constellation with the convolutional codes of (44) and (42) we obtain codes of rate 12 bits/four-dimensional signal with coding gains of 5.156 and 6.406 dB, respectively.

If we increase  $k_2$  further we find from Theorem 8 (since  $D_4$  has density  $\pi^2/16$  [12]) that the gain of the 16-state code approaches 5.227 dB and the gain of the 64-state code approaches 6.477 dB.

8) *4- and 8-State Binary Codes with Rate 2 Bits/One-Dimensional Symbol*: The lattice  $\Lambda$  is the integer lattice  $\mathbf{Z}$  and  $\Lambda' = 2n\mathbf{Z}$ , so  $d_2 = 4n^2$ . The 8-point signal constellation is a translate of the constellation shown in Fig. 19.

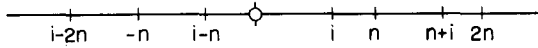


Fig. 19. 8-point constellation used in Example 8.

The constellation involves only 4 cosets of  $\Lambda'$  in  $\Lambda$ . We take  $A = \text{GF}(2)$  and  $k_1 = k_2 = 1$ . The rate of the convolutional code is 1 bit/coset, and the remaining bit selects one of two points in a coset. The generator matrix of the code is

$$G = [n|i|n], \quad (45)$$

and so the cosets involved in the signal constellation are  $\Lambda'$ ,  $\Lambda' + n$ ,  $\Lambda' + i$ , and  $\Lambda' + (n + i)$ . Observe that  $\Lambda^* = \{\Lambda', \Lambda' + n\} \cong \mathbf{Z}_2$  and that only one column of  $G$  is chosen from outside  $\Lambda^*$ . As in Lemma 1 the cosets are used equally often.

Next we prove that the code is regular in the sense that if  $u(j), v(j)$  are inputs then the norm of the coset  $(\Lambda' + Gu(j)) - (\Lambda' + Gv(j))$  depends only on the modulo 2 sum  $u(j) \oplus v(j)$ . The argument is similar to the proof of Lemma 2.

If  $x(j), z(j)$  are input sequences with disjoint support then we must prove that

$$N(\Lambda' + G(x(j) - z(j))) = N(\Lambda' + G(x(j) + z(j))). \quad (46)$$

Now

$$\begin{aligned} \Lambda' + G(x(j) - z(j)) \\ = (\Lambda' + G(x(j) + z(j))) - (\Lambda' + 2Gz(j)), \end{aligned}$$

where

$$\Lambda' + 2Gz(j) = \Lambda' \quad \text{or} \quad \Lambda' + 2i.$$

If  $2Gz(j) \in \Lambda'$  then we are done, so we may suppose  $2Gz(j) \notin \Lambda'$ . Hence  $Gz(j) \notin \Lambda^*$  and  $z_{j-1} = 1$ . Since the supports of  $x(j), z(j)$  are disjoint we have  $x_{j-1} = 0$ ,  $Gx(j) \in \Lambda^*$ . The four possibilities are considered in Table XII. Regularity follows from the symmetry of coset norms about  $\Lambda' + n$ .

TABLE XII

$Gx(j)$	Coset Representatives for		
	$Gz(j)$	$G(x(j) - z(j))$	$G(x(j) + z(j))$
0	$i$	$-i$	$i$
0	$n + i$	$-(n + i)$	$n + i$
$n$	$i$	$n - i$	$n + i$
$n$	$n + i$	$-i$	$i$

The squared distance  $d_1$  is calculated as in previous examples. The first three rows in Table XIII present instances of this construction. Uncoded transmission at this rate uses the constellation (16) with  $n = 1, b = 2, d^{(u)} = 4, P^{(u)} = 5$ . The third code in Table XIII was given by Ungerboeck [25], as were the last two codes, which illustrate that the coding gain can be increased by increasing the memory of the convolutional code (always remembering to choose only one column of  $G$  from outside  $\Lambda^*$ ).

#### IV. ASYMPTOTIC BEHAVIOR

In this section we analyze the asymptotic behavior of these codes as the size of the signal constellation increases while the lattice and sublattice are kept fixed.

We use the notation of Section II. The *determinant* of the lattice  $\Lambda$  is denoted by  $\det \Lambda$ ,  $\Delta$  is the *density* of  $\Lambda$ ,  $\mu$  is the minimal norm, and  $V_n = \pi^{n/2}/\Gamma((n+2)/2)$  is the volume of a unit sphere (see [22], [24]).  $\delta = \Delta/V_n$  is the *center density*. These quantities are related by the formula

$$\Delta = \frac{V_n}{\det \Lambda} \left( \frac{\mu}{4} \right)^{n/2}. \quad (47)$$

We assume that  $\Lambda$  is a *rational* lattice, in other words that there is a constant  $\phi$  such that  $\phi x \cdot y \in \mathbf{Z}$  for all  $x, y \in \Lambda$  [13, p. 115]. (This assumption of rationality could easily be dropped, at the cost of only slightly weakening the error terms in Theorem 5.)

TABLE XIII  
ONE-DIMENSIONAL CODES DESCRIBED IN EXAMPLE 8

Generator Matrix $G$	Constellation	$d_1$	$d_2$	$d$	$P$	Coding Gain
$[3, 2, 3]^\dagger$	$\{\pm 1, \pm 2, \pm 4, \pm 5\}$	20	36	20	$\frac{46}{4}$	3.372
$[5, 3, 5]^\dagger$	$\left\{ \frac{\pm 3}{2}, \frac{\pm 7}{2}, \frac{\pm 13}{2}, \frac{\pm 17}{2} \right\}$	$\frac{58}{4}$	100	$\frac{58}{4}$	$\frac{129}{16}$	3.518
$[2, 1, 2]^*$	$\left\{ \frac{\pm 1}{2}, \frac{\pm 3}{2}, \frac{\pm 5}{2}, \frac{\pm 7}{2} \right\}$	9	16	9	$\frac{21}{4}$	3.310
$[2, 2, 1, 2]^*$	"	10	16	10	$\frac{21}{4}$	3.768
$[2, 2, 0, 1, 2]^*$	"	11	16	11	$\frac{21}{4}$	4.181

<sup>†</sup> Found by Calderbank and Mazo [3].

\* Found by Ungerboeck [25].

*Constellations of Minimal Norm*

We first analyze a constellation of minimal total norm, chosen without regard to how many points fall into each coset of  $\Lambda'$ . Let a constellation of  $M$  points in  $\Lambda$  be constructed by the following process. Choose an arbitrary center  $c$ , draw a sphere around  $c$  of the smallest possible radius, say  $\sqrt{t}$ , such that there are at least  $M$  lattice points inside or on the sphere, and then discard enough points from the boundary of the sphere so as to leave exactly  $M$  points. The radius  $\sqrt{t}$  and the average norm  $P$  of the resulting configuration are given by the following theorem.

*Theorem 4:* As  $M \rightarrow \infty$ ,

$$a) \quad t = \left( \frac{M \det \Lambda}{V_n} \right)^{2/n} (1 + o(1)), \quad (48)$$

$$b) \quad P = \frac{n}{n+2} \left( \frac{M \det \Lambda}{V_n} \right)^{2/n} (1 + o(1)). \quad (49)$$

The leading term in (48) may also be expressed in terms of the density  $\Delta$ , using (47):

$$\left( \frac{M \det \Lambda}{V_n} \right)^{2/n} = \frac{\mu}{4} \left( \frac{M}{\Delta} \right)^{2/n}. \quad (50)$$

The proof is based on the following result, which combines the work of a number of mathematicians (see Fricker [13]). Let  $a_\Lambda(t)$  denote the number of points in  $\Lambda$  of norm  $t$ , and let

$$S_\Lambda(t) = \sum_{m \leq t} a_\Lambda(m).$$

*Theorem 5:* For a rational lattice  $\Lambda$ ,

$$S_\Lambda(t) = \frac{V_n}{\det \Lambda} t^{n/2} (1 + \theta), \quad \text{as } t \rightarrow \infty, \quad (51)$$

where the error term  $\theta$  satisfies

$$\begin{aligned} \theta &= O(t^{-2/3}), & \text{if } n = 2, \\ \theta &= O(t^{-3/4}), & \text{if } n = 3, \\ \theta &= O((\log t)^{2/3}/t), & \text{if } n = 4, \\ \theta &= O(t^{-1}), & \text{if } n \geq 5. \end{aligned}$$

*Corollary 6:* For a rational lattice  $\Lambda$ ,

$$S_\Lambda(t) = \frac{V_n}{\det \Lambda} t^{n/2} (1 + O(t^{-2/3})). \quad (52)$$

*Proof:* Corollary 6 follows from Theorem 5 since the error term is weakest when  $n = 2$ .

*Sketch of Proof of Theorem 4:* a) By solving (52) for  $t$  we obtain (48). b) The total norm of the constellation is given by

$$\sum_{m=0}^t ma_\Lambda(m)(1 + o(1)). \quad (53)$$

(The total norm is exactly  $\sum_{m=0}^t ma_\Lambda(m)$  if 1) the constella-

tion is centered at a lattice point and 2) the total number of points in the sphere of radius  $\sqrt{t}$  is exactly  $N$ . The corrections needed for the general case can be shown to be of lower order; we omit the details.) From Abel's formula for summation by parts,

$$\sum_{m=0}^t ma_\Lambda(m) = tS_\Lambda(t) - \sum_{m=0}^{t-1} S_\Lambda(m). \quad (54)$$

Equation (49) follows from (53), (54), and Corollary 6.

For example, Theorem 4b asserts that, for two-dimensional constellations, doubling the number of points in the constellation also roughly doubles the average energy. But for four-dimensional constellations the average energy is multiplied by about  $\sqrt{2}$  when the number of points in the constellation is doubled.

*Constellations with Equal Numbers of Points in Each Coset*

We now consider constellations of the type needed for our trellis codes. Let  $\Lambda$  be a rational lattice, and let  $\Lambda'$  be a sublattice of index  $i$ . Suppose a constellation of  $M = iv$  points is constructed by the following process. Choose an arbitrary center  $c$  and draw a sphere around  $c$  of the smallest possible radius, say  $\sqrt{t}$ , such that the sphere contains at least  $v$  points from each coset of  $\Lambda'$ . Discard points from the outside of the sphere so as to leave exactly  $v$  points in each coset.

*Theorem 7:* Equations (48) and (49) apply to the constellation just described.

*Sketch of Proof:*  $\Lambda'$  is itself an  $n$ -dimensional lattice with  $\det \Lambda' = i \cdot \det \Lambda$ . Applying (51) to  $\Lambda'$ , we see that a sphere just large enough to contain  $v$  points of  $\Lambda'$  has radius  $\sqrt{t}$  determined by

$$v = \frac{M}{i} = \frac{V_n}{i \det \Lambda} t^{n/2} (1 + O(t^{-2/3})), \quad (55)$$

and therefore is given by (48). Since the cosets of  $\Lambda'$  are also translates of  $\Lambda'$ ,  $t$  need only be increased by a factor  $(1 + o(1))$  in order to include  $v$  points from each coset. Thus the radius of this constellation is essentially the same as the constellation considered in Theorem 4. The formula for the average norm then follows as in Theorem 4.

We apply Theorem 7 to the codes constructed in Section II, where  $i = m^{n/2}$ ,  $v = a^{k_2}$ ,  $d_2 = m\mu$ , and obtain

$$\begin{aligned} \frac{d}{P} &= \frac{d}{d_2} \cdot \frac{d_2}{P} \\ &= \frac{d}{d_2} \cdot m\mu \frac{n+2}{n} \left( \frac{\Delta}{m^{n/2} a^{k_2}} \right)^{2/n} \frac{4}{\mu} (1 + o(1)) \\ &= \frac{4d}{d_2} \frac{n+2}{n} \left( \frac{\Delta}{a^{k_2}} \right)^{2/n} (1 + o(1)). \end{aligned} \quad (56)$$

### Uncoded Constellations

For comparison we analyze an uncoded constellation of  $M' = a^{k_1+k_2}$  points. We assume<sup>2</sup> that  $M'$  has the form  $M' = (2b)^n$ , and use (16) as the uncoded constellation. The minimal squared distance is  $d^{(u)} = 4$ , and the total norm is

$$\begin{aligned} & \sum_{x_1} \sum_{x_2} \cdots \sum_{x_n} (x_1^2 + \cdots + x_n^2) \\ &= 2n(2b)^{n-1}(1^2 + 3^2 + \cdots + (2b-1)^2) \\ &= \frac{1}{3}n(4b^2-1)(2b)^n. \end{aligned} \quad (57)$$

Therefore,

$$P^{(u)} = \frac{1}{3}n(4b^2-1) \quad (58)$$

$$= \frac{1}{3}n(M')^{2/n}(1+o(1)), \quad (59)$$

and the asymptotic formula for the uncoded figure of merit is

$$\frac{d^{(u)}}{P^{(u)}} = \frac{12}{n}(m')^{-2/n}(1+o(1)). \quad (60)$$

*Theorem 8:* As  $k_2 \rightarrow \infty$  the coding gain of the codes described in Section II approaches

$$10 \log_{10} \left\{ \frac{n+2}{3} \frac{d}{d_2} \Delta^{2/n} m^{\rho_1} \right\} \text{ dB} \quad (61)$$

where  $\Delta$  is the density of  $\Lambda$  and  $\rho_1$  (see (5)) is the fractional rate of the convolutional code.

*Proof:* The proof follows from (18), (56), (60), using (5) to write  $a^{2k_1/n} = m^{\rho_1}$ .

*Example:* For the codes in Example 1a of Section III,  $\Lambda = \mathbf{Z}^2$ ,  $\det \Lambda = 1$ ,  $\mu = 1$ ,  $\Delta = \pi/4$  and  $d/d_2 = 6/8$ , so from (61) the gain approaches

$$10 \log_{10} \pi = 4.971 \cdots \quad \text{as } k_2 \rightarrow \infty. \quad (62)$$

The following useful formula describes the coding gain that is theoretically achievable simply by mapping messages directly into lattice points (without using a trellis code).

*Theorem 9:* If messages are represented by a signal constellation of  $M$  points in an  $n$ -dimensional lattice  $\Lambda$  of density  $\Delta$  (or center density  $\delta$ ), a coding gain is attained that approaches

$$10 \log_{10} \left\{ \frac{n+2}{3} \Delta^{2/n} \right\} = 10 \log_{10} \left\{ \frac{n+2}{3} (\delta V_n)^{2/n} \right\} \text{ dB} \quad (63)$$

as  $M \rightarrow \infty$ .

*Proof:* The proof follows directly from (49), (50), (59), or as a special case of Theorem 8.

<sup>2</sup>This assumption does not affect our final estimate.

Table XIV shows the coding gains theoretically achievable from various known lattices according to Theorem 9.

TABLE XIV  
POTENTIAL CODING GAINS OF VARIOUS LATTICES  
(SEE THEOREM 9).  $\delta$  = CENTER DENSITY

Dimension	$\delta$	Name	Reference	Gain (dB)
2	$1/(2\sqrt{3})$	$A_2$	[17]	0.825
4	$1/8$	$D_4$	[17]	1.961
6	$1/(8\sqrt{3})$	$E_6$	[17]	2.832
8	$1/16$	$E_8$	[17]	3.739
12	$1/27$	$K_{12}$	[17]	4.514
16	$1/16$	$\Lambda_{16}$	[17]	5.491
24	1	$\Lambda_{24}$	[17]	7.116
40	16	$\Lambda_{40}$	[17]	7.843
48	$2^{14.039}$	$P_{48q}$	[17]	9.037
64	$2^{22}$	—	[1]	9.397
80	$2^{36}$	—	[1]	10.070
128	$2^{88}$	—	[1]	11.557

### V. DISCUSSION

*The Main Table:* In Table XV we have collected the trellis codes of rates 1.5, 2, 2.5, and 3 bits/dimension constructed in Section III, together with some codes of Ungerboeck [25]. The latter are distinguished by the letter U in column 1. Codes found by Calderbank and Mazo (mentioned in Example 8) have a C in column 1. The third column gives the number of states in the convolutional code, and the fourth column gives the number of points in the signal constellation. The sixth column gives the path multiplicity per two-dimensional signal point, and the last column either gives the example in Section III where the code can be found or a reference.

*Remarks:* 1) In Ungerboeck's Table III, in the column headed  $m = 3$ , the 8-point uncoded constellation used for comparison is his constellation 8-AMPM. Fortunately, this has the same ratio of  $d^{(u)}/P^{(u)} = 4/5$  as our 8-point uncoded constellation (a  $3 \times 3$  array with one corner removed), so we are able to use his values for the coding gain of his codes. For the codes taken from the column headed  $m = 5$  of Ungerboeck's Table III we have calculated the coding gain by subtracting 0.2 dB from Ungerboeck's figure, since his standard of comparison is the 32-AMPM constellation whereas our standard of comparison is the Campopiano-Glazer configuration. 2) Forney *et al.* [12] give a code similar to code 27, but with a gain of only 4.5 dB. 3) The performance of codes 15U, 24U, 29U, 34U, 38U can be improved by changing the signal constellation from 32-AMPM to the Campopiano-Glazer constellation (Fig. 3). For example the coding gain of code 29U is increased from 4.559 to 4.771 dB.

#### Comparisons

Table XV shows that, although in some cases Ungerboeck's codes have a higher coding gain than ours<sup>3</sup> (compare codes 1 and 2 or 41 and 42, for example), there are many cases where our codes have a slightly higher gain

<sup>3</sup>A precise comparison is difficult because the coding gains in [25] are only given to one decimal place. The path multiplicities for these codes are not given in [25] but may be found in [29].



TABLE XV  
SUMMARY OF TRELLIS CODES HAVING RATES 1.5, 2, 2.5, AND 3 BITS/DIMENSION

Number	Rate (bits/dimension)	Number of States	$M$	Gain (dB)	Path Multiplicity	$\Lambda$	Convolutional Code	Signal Constellation	$\S$
1	1.5	4	32	2.785	4	$Z^2$	(30)	Fig. 11	(3c)
2U	1.5	4	16	3.0	4	—	—	16-QASK	[25]
3	1.5	8	16	3.979	16	$Z^2$	(25)	Fig. 3	(1c)
4U	1.5	8	16	4.0	16	—	—	16-QASK	[25]
5	1.5	8	32	4.168	32	$Z^2$	(31)	Fig. 11	(3c)
6	1.5	16	32	4.546	48	$Z^2$	(29)	Fig. 11	(3c)
7	1.5	16	16	4.771	56	$Z^2$	(19)	Fig. 3	(1b)
8U	1.5	16	16	4.8	56	—	—	16-QASK	[25]
9	1.5	64	32	5.216	48	$Z^2$	(32)	Fig. 11	(3c)
10U	1.5	64	16	5.4	56	—	—	16-QASK	[25]
11	1.5	64	16	5.441	48	$Z^2$	(26)	Fig. 3	(1c)
12	1.5	128	32	5.795	228	$Z^2$	(33)	Fig. 11	(3c)
13U	1.5	128	16	6.0	344	—	—	16-QASK	[25]
14	1.5	256	16	6.021	112	$Z^2$	(27)	Fig. 3	(1c)
15U	2	4	32	2.8	4	$Z^2$	—	32-AMPM	[25]
16	2	4	64	2.903	4	$Z^2$	(30)	Fig. 11	(3b)
17	2	4	32	3.010	32	$Z^2$	(28)	Fig. 8	(2a)
18	2	4	32	3.200	27	—	(28)	Fig. 10	(2b)
19U	2	4	8	3.310	—	$Z$	Table XIII	Table XIII	(8)
20C	2	4	8	3.372	—	$Z$	Table XIII	Table XIII	(8)
21C	2	4	8	3.518	—	$Z$	Table XIII	Table XIII	(8)
22	2	8	32	3.529	16	$A_2$	(35)	Fig. 16	(4)
23U	2	8	8	3.768	—	$Z$	Table XIII	Table XIII	(8)
24U	2	8	32	3.768	16	$Z^2$	—	32-AMPM	[25]
25	2	8	32	3.979	16	$Z^2$	(25)	Fig. 3	(1c)
26	2	8	64	4.286	32	$Z^2$	(31)	Fig. 11	(3b)
27	2	8	512	4.717	44	$(2Z+1)^4$	(38)	[4]	(6)
28U	2	16	8	4.181	—	$Z$	Table XIII	Table XIII	(8)
29U	2	16	32	4.559	56	$Z^2$	—	32-AMPM	[25]
30	2	16	64	4.664	48	$Z^2$	(29)	Fig. 11	(3a)
31	2	16	32	4.771	56	$Z^2$	(19)	Fig. 3	(1a)
32	2	16	512	4.881	152	$D_4$	(44)	Table XI	(7b)
33U	2	64	8	5.2	—	$Z$	—	8-AM	[25]
34U	2	64	32	5.2	56	$Z^2$	—	32-AMPM	[25]
35	2	64	64	5.333	48	$Z^2$	(32)	Fig. 11	(3b)
36	2	64	32	5.441	48	$Z^2$	(26)	Fig. 3	(1c)
37	2	64	512	6.131	828	$D_4$	(42)	Table XI	(7a)
38U	2	128	32	5.8	344	$Z^2$	—	32-AMPM	[25]
39	2	128	64	5.913	228	$Z^2$	(33)	Fig. 11	(3b)
40	2	256	32	6.021	112	$Z^2$	(27)	Fig. 3	(1c)
41U	2.5	4	64	2.8	4	$Z^2$	—	64-QASK	[25]
42	2.5	4	128	2.903	4	$Z^2$	(30)	Fig. 15	(3c)
43U	2.5	8	64	3.8	16	$Z^2$	—	64-QASK	[25]
44	2.5	8	64	3.872	16	$Z^2$	(25)	Fig. 6	(1c)
45	2.5	8	128	4.286	32	$Z^2$	(31)	Fig. 15	(3c)
46U	2.5	16	64	4.6	56	$Z^2$	—	64-QASK	[25]
47	2.5	16	64	4.664	56	$Z^2$	(19)	Fig. 6	(1b)
48	2.5	16	128	4.664	48	$Z^2$	(29)	Fig. 15	(3c)
49U	2.5	64	64	5.2	56	$Z^2$	—	64-QASK	[25]
50	2.5	64	64	5.333	48	$Z^2$	(26)	Fig. 6	(1c)
51	2.5	64	128	5.333	48	$Z^2$	(32)	Fig. 15	(3c)
52U	2.5	128	64	5.8	344	$Z^2$	—	64-QASK	[25]
53	2.5	128	128	5.913	228	$Z^2$	(33)	Fig. 15	(3c)
54	2.5	256	64	5.913	112	$Z^2$	(27)	Fig. 6	(1c)
55	3	8	128	4.084	16	$Z^2$	(25)	Fig. 7	(1c)
56	3	8	$2^{13}$	4.904	44	$(2Z+1)^4$	(38)	[4]	(6)
57	3	16	128	4.876	56	$Z^2$	(19)	Fig. 7	(1b)
58	3	16	$2^{13}$	5.156	152	$D_4$	(44)	—	(7b)
59	3	64	128	5.545	48	$Z^2$	(26)	Fig. 7	(1c)
60	3	64	$2^{13}$	6.406	828	$D_4$	(42)	—	(7b)
61	3	256	128	6.125	112	$Z^2$	(27)	Fig. 7	(1c)

(compare codes 4 and 5, 10 and 11, 15 and 16, for example). Codes 28–32, all of which have rate 2 bits/dimension and use a 16-state convolutional code, illustrate this. Code 28, from [25], gives a gain of 4.18 dB using a one-dimensional constellation, while code 29, also from [25], realizes a gain of 4.6 dB by using a 32-point two-di-

mensional constellation (32-AMPM). By using a 64-point constellation in  $Z^2$  with sublattice  $4Z^2$  we obtain 4.66 dB, and with a 32-point constellation in  $Z^2$  with sublattice  $\Theta Z^2$ , where  $\Theta$  is the endomorphism defined in (1), we obtain 4.77 dB. By going to the four-dimensional lattice  $D_4$  and using a 512-point constellation we get 4.88 dB (this

gain is obtained by reducing the average transmitted energy).

The path multiplicity of code 32 is 152 (measured in channel uses per trellis stage) compared with only 56 for code 29. This does not mean that four-dimensional coding schemes are a bad idea. For example, by replacing (38) with the generator matrix

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix},$$

we obtain a 16-state code for the lattice  $(2\mathbf{Z} + 1)^4$ , for which  $d_1 = 5$ ,  $d_2 = 0$ ,  $d = 4$  and the nominal coding gain is unchanged at 4.717 dB. We have doubled the number of states to reduce the path multiplicity by a factor of 4, from 48 to 12. A 16-state code with the same performance is described by Wei in [29].

The codes constructed by Ungerboeck all employ a constellation with twice as many signal points as are required for uncoded transmission at the same rate. Our table shows that in some cases increasing the expansion factor is profitable (compare codes 25 and 26), while in other cases it is not (compare codes 6 and 7).

It is of course to be expected (see the discussion in Forney *et al.* [12]) that by using larger constellations and higher dimensional lattices a bigger gain will be obtained. On the other hand, in some cases the path multiplicity rises sharply when the dimension of the signal constellation is increased (from 2 to 4, for example, as in codes 36 and 37), while in other cases it does not (compare codes 26 and 27). We feel that the merit of the construction technique described in this paper is that it permits the coding theorist to work with larger constellations and more complicated lattices than can be handled by set partitioning.

The two methods are in any case very similar. This can be seen by considering [25, fig. 5]. The successive partitions in this tree just pick out points that lie in different cosets of  $\mathbf{Z}^2$  under the endomorphism of  $\mathbf{Z}^2$  defined by

$$\Theta = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

In fact it was by consideration of such examples that we discovered the present construction.

#### Choice of Lattice

Some interesting conclusions concerning the choice of a lattice may be drawn from Theorem 8. In dimension  $n = 4$ , for example, the brackets in (61) contain a factor  $\sqrt{\Delta}$ , where  $\Delta$  is the density of the lattice. Since  $D_4$  is twice as dense as the cubic lattice  $\mathbf{Z}^4$  [17], we may expect to gain 1.5 dB by choosing the signal constellation from the denser lattice. This is borne out by a comparison of codes 27 and 37 in the table. On the other hand, the convolutional code over  $D_4$  defined by (42) requires four times as many states as the convolutional code over  $\mathbf{Z}^4$  defined by (38).

In two dimensions, the ratio of densities of  $A_2$  and the square lattice  $\mathbf{Z}^2$  is  $2/\sqrt{3} = 1.155$ , and so the potential

gain is only

$$10 \log_{10} (1.155) = 0.625 \text{ dB}.$$

Furthermore, it turns out to be more difficult to find good constellations in  $A_2$  than in  $\mathbf{Z}^2$ , with the result that, as the table shows, trellis codes based on  $A_2$  are not as successful as we initially hoped.

As to the choice of sublattice  $\Lambda'$ , Theorem 8 shows that, other things being equal, the larger the magnification factor  $m$ , i.e. the larger the index  $|\mathcal{R}| = |\Lambda/\Lambda'|$ , the better. On the other hand, as the index increases it becomes more difficult to find the convolutional code.

We are currently investigating higher dimensional lattices. Promising endomorphisms are known for the lattices  $E_8$ ,  $K_{12}$ ,  $\Lambda_{16}$ , and  $\Lambda_{24}$  in 8, 12, 16, and 24 dimensions, respectively [1], [2], [6] (see also [31]).

#### ACKNOWLEDGMENT

We thank J. E. Mazo and the referees for a number of very helpful comments.

#### REFERENCES

- [1] E. S. Barnes and N. J. A. Sloane, "New lattice packings of spheres," *Can. J. Math.*, vol. 35, pp. 117-130, 1983.
- [2] A. Bos, J. H. Conway, and N. J. A. Sloane, "Further lattice packings in high dimensions," *Mathematika*, vol. 29, pp. 171-180, 1982.
- [3] A. R. Calderbank and J. E. Mazo, "A new description of trellis codes," *IEEE Trans. Inform. Theory*, vol. IT-30, pp. 784-791, 1984.
- [4] A. R. Calderbank and N. J. A. Sloane, "Four-dimensional modulation with an eight-state trellis code," *Bell Syst. Tech. J.*, vol. 64, pp. 1005-1018, 1985.
- [5] C. N. Campopiano and B. G. Glazer, "A coherent digital amplitude and phase modulation scheme," *IRE Trans. Commun. Syst.*, vol. CS-10, pp. 90-95, 1962.
- [6] J. H. Conway, R. A. Parker, and N. J. A. Sloane, "The covering radius of the Leech lattice," *Proc. Roy. Soc. London, Ser. A*, vol. 380, pp. 261-290, 1982.
- [7] J. H. Conway and N. J. A. Sloane, "Voronoi regions of lattices, second moments of polytopes, and quantization," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 211-226, 1982.
- [8] —, "Fast quantizing and decoding algorithms for lattice quantizers and codes," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 227-232, 1982.
- [9] —, "A fast encoding method for lattice codes and quantizers," *IEEE J. Select. Areas Commun.*, vol. IT-29, pp. 820-824, 1983.
- [10] —, "On the Voronoi regions of certain lattices," *SIAM J. Algebraic Discrete Methods*, vol. 5, pp. 294-305, 1984.
- [11] —, "Soft decoding techniques for codes and lattices, including the Golay code and the Leech lattice," *IEEE Trans. Inform. Theory*, vol. IT-32, pp. 41-50, 1986.
- [12] G. D. Forney, Jr., R. G. Gallager, G. R. Lang, F. M. Longstaff, and S. U. Qureshi, "Efficient modulation for band-limited channels," *IEEE J. Select. Areas Commun.*, vol. SAC-2, pp. 632-647, 1984.
- [13] F. Fricker, *Einführung in die Gitterpunktlehre*. Boston, MA: Birkhäuser, 1982.
- [14] M. Hall, Jr., *The Theory of Groups*. New York: Macmillan, 1959.
- [15] I. N. Herstein, *Topics in Algebra*. Waltham, MA: Ginn and Co., 1964.
- [16] E. L. Lawler, *Combinatorial Optimization: Networks and Matroids*. New York: Holt, Rinehart and Winston, 1976.
- [17] J. Leech and N. J. A. Sloane, "Sphere packing and error-correcting codes," *Can. J. Math.*, vol. 23, pp. 718-745, 1971.
- [18] S. Lin and D. J. Costello, Jr., *Error Control Coding*. Englewood Cliffs, NJ: Prentice-Hall, 1983.

- [19] R. J. McEliece, *The Theory of Information and Coding*. Reading, MA: Addison-Wesley, 1977.
- [20] M. A. Marsan, G. Albertengo, S. Benedetto, and F. Giachin, "Theoretical aspects and performance evaluation of channel encoding with redundant modulator alphabets," in *Proc. IEEE Global Telecommunications Conf.*, Nov. 1984, New York: IEEE Press, pp. 688-692.
- [21] E. M. Reingold, J. Nievergelt, and N. Deo, *Combinatorial Algorithms: Theory and Practice*. Englewood Cliffs, NJ: Prentice-Hall, 1977.
- [22] N. J. A. Sloane, "Binary codes, lattices and sphere packings," in *Combinatorial Surveys*, P. J. Cameron, Ed. New York: Academic, 1977, pp. 117-164.
- [23] —, "Self-dual codes and lattices," in *Relations Between Combinatorics and Other Parts of Mathematics*, *Proc. Sympos. Pure Math.*, vol. XXXIV. Providence, RI: Amer. Math. Soc., 1979, pp. 273-308.
- [24] —, "Tables of sphere packings and spherical codes," *IEEE Trans. Inform. Theory*, vol. IT-27, pp. 327-338, 1981.
- [25] G. Ungerboeck, "Channel coding with multilevel/phase signals," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 55-67, 1982.
- [26] A. J. Viterbi and J. K. Omura, *Principles of Digital Communication and Coding*. New York: McGraw-Hill, 1979.
- [27] B. L. van der Waerden, *Modern Algebra*. New York: Ungar, 1953.
- [28] L. F. Wei, "Rotationally invariant convolutional channel coding with expanded signal space—II: Nonlinear codes," *IEEE J. Select. Areas Commun.*, vol. SAC-2, pp. 672-686, 1984.
- [29] —, "Trellis-coded modulation with multi-dimensional constellations," *IEEE Trans. Inform. Theory*, to appear.
- [30] S. G. Wilson, H. A. Sleeper, and N. K. Srinath, "Four-dimensional modulation and coding: An alternative to frequency-reuse," in *Science, Systems and Services for Communications*. P. Dewilde and C. A. May, Eds. New York: IEEE/Elsevier-North Holland, 1984, pp. 919-923.
- [31] A. R. Calderbank and N. J. A. Sloane, "An eight-dimensional trellis code," *Proc. IEEE*, vol. 74, pp. 757-759, 1986.
-