Trellis-Coded Modulation with Multidimensional Constellations

LEE-FANG WEI, MEMBER, IEEE

Abstract—Trellis-coded modulation schemes using four-, eight-, or 16-dimensional constellations have a number of potential advantages over the usual two-dimensional schemes: a smaller constituent two-dimensional constellation, easier tolerance to phase ambiguities, and a better trade-off between complexity and coding gain. A number of such schemes are presented and evaluated. Starting with a variety of multidimensional lattices, we show how to select multidimensional constellations, how to partition them into subsets, how to construct trellis codes using those subsets, and how to map bits to constellation points. Simplifications of the Viterbi decoding algorithm are presented. We conclude that there are multidimensional trellis-coded modulation schemes that perform better for the same complexity than do two-dimensional schemes.

I. Introduction

TRELLIS-CODED modulation schemes using two-dimensional (2D) constellations have been shown to improve the error performance of synchronous data links without sacrificing data rate or requiring more bandwidth [1]–[3]. In these schemes, to send Q information bits in each signaling interval, a 2D constellation of 2^{Q+1} points is used. The constellation is partitioned into 2^{m+1} subsets with enlarged intrasubset minimum Euclidean distance. Of the Q bits that arrive in each signaling interval, m enter a rate-$m/m+1$ trellis encoder, and the resulting $m+1$ coded bits specify which subset is to be used. The remaining information bits specify which point from the selected subset is to be transmitted.

An eight-state nonlinear trellis code with 4-dB coding gain has now been adopted in the international CCITT standards V.32 for 9.6-kbit/s transmission over the switched telephone network and V.33 for 14.4-kbit/s transmission over private lines [2], [4], [5]. The 2D constellations used in those two standards are the 32-point cross constellation (32-CR) and the 128-point cross constellation (128-CR), respectively. The use of a nonlinear trellis code allows the scheme to be immune to the 90° phase ambiguities of those constellations [2], [6].

To improve the performance of the eight-state trellis code further, more states may be used. However, the returns are diminishing. The coding gain increases more slowly and the error coefficient (the multiplicity of minimum-Euclidean-distance error events) of the code starts to dominate performance.

Manuscript received August 9, 1985; revised September 23, 1986.

The author was with the Codex Corporation, Mansfield, MA 02048. He is now with AT&T Bell Laboratories, Crawford Hill Laboratory, Holmdel, NJ 07733, USA.

IEEE Log Number 8612409.

An inherent cost of these coded schemes is that the size of the 2D constellation is doubled over uncoded schemes. This is due to the fact that a redundant bit is added every signaling interval. Without that cost, the coding gain of those coded schemes would be 3 dB greater. Using a multidimensional (> 2) constellation with a trellis code of rate $m/m+1$ can reduce that cost because fewer redundant bits are added for each 2D signaling interval [3]. For example, that cost is reduced to about 1.5 or 0.75 dB if four-dimensional (4D) or eight-dimensional (8D) constellations are used, respectively. Additional coding gain may also be derived from the multidimensional constellation itself [3], [7]–[12]. These observations motivated the investigation of trellis-coded modulation using multidimensional constellations.

Trellis-coded modulation schemes using 4D constellations have been reported in several papers [3], [13]–[15]. In both [3] and [15], the 4D constellation is taken from the 4D rectangular lattice. The 4D constellation is partitioned into 16 4D subsets with four times larger intrasubset minimum squared Euclidean distance (MSED). In [3], the partitioning of the 4D constellation is based on the partitioning of each constituent 2D constellation into four 2D subsets; each 4D subset is formed by concatenating a pair of 2D subsets. In [15], the partitioning of the 4D constellation is done algebraically without referring to the partitioning of the constituent 2D constellations. However, the results of the two partitionings are the same. Three of the $2Q$ information bits arriving in each block of two signaling intervals enter a rate-3/4 eight-state trellis encoder with a minimum free Hamming distance of four. The resulting four coded bits specify which 4D subset is to be used. The remaining information bits specify which point from the selected 4D subset is to be transmitted. The 4D constellation therefore has 2^{2Q+1} points. The mapping of coded bits to 4D subsets is such that the Hamming distance between any two different groups of four coded bits is proportional to the MSED between the two corresponding 4D subsets. The coding gain therefore is approximately 4.5 dB, which is a gain of 6 dB from the trellis code if the 4D constellation were not expanded from 2^{2Q} to 2^{2Q+1} points, less 1.5 dB due to that expansion.

Both references, however, did not address other issues such as error coefficient, phase ambiguities of constellation, and complexity. It was not clear whether those eight-state 4D trellis-coded modulation schemes performed better for the same complexity than did two-dimensional

0018-9448/87/0700-0483$01.00

schemes. It was also not clear how to construct other multidimensional trellis-coded modulation schemes.

In this paper, a novel geometrical approach to partitioning multidimensional lattices into sublattices with enlarged intrasublattice MSED is described in Section II. The approach simplifies both the construction of multidimensional trellis-coded modulation schemes and the corresponding maximum-likelihood decoding. It therefore opens the door to extensive studies of trellis-coded modulation using various multidimensional lattices. The lattices considered are the 4D, 8D, and 16D rectangular lattices, the densest 4D lattice D_4, the densest 8D lattice E_8, and an 8D lattice which is the union of E_8 and a rotated version of E_8. This latter 8D lattice has not been used before and will be referred to as DE_8 (D stands for double) in this paper. The partitioning of these lattices is based on the partitioning of their constituent 2D rectangular lattices. Further, the partitioning of a multidimensional lattice is done in an iterative manner. That is, the partitioning of a $2N$-dimensional lattice is based on the partitioning of the constituent N-dimensional lattices, which is in turn based on the partitioning of the constituent $N/2$-dimensional lattices. To make the resulting trellis-coded modulation schemes transparent to the phase ambiguities of a multidimensional lattice, the partitioning of the lattice into sublattices is also done such that each sublattice is rotationally invariant to as many phase ambiguities as possible.

In Section III, we show how to construct a finite multidimensional constellation from an infinite multidimensional lattice. The construction of the multidimensional constellation makes it possible to convert a complicated multidimensional constellation mapping into multiple simple constituent 2D constellation mappings. The size of the constituent 2D constellations and the peak-to-average power ratio (of the multidimensional constellation) are also reduced as a result of this construction process. Both the small size and small peak-to-average power ratio are desirable in a communication system where impairments other than additive Gaussian noise, such as linear or nonlinear distortion or phase jitter, are also present, of which voiceband data transmission is an example. With all of these desirable characteristics, the construction process leads the way to practical applications of multidimensional trellis-coded modulation. The partitioning of the multidimensional lattice underlies the partitioning of the multidimensional constellation.

Section IV is the main section of this paper. A number of trellis codes using various partitionings of various multidimensional constellations obtained in the previous two sections are presented. General principles for constructing those codes are discussed. The codes are evaluated in terms of their coding gain, error coefficient, transparency to phase ambiguities, size of constituent 2D constellations, peak-to-average power ratio, and complexity.

A simplified maximum-likelihood decoding algorithm is described in Section V. In that algorithm, the point in each multidimensional subset closest to a received multidimensional point is also found in an iterative manner, as in the partitioning of multidimensional lattices described in Section II. Section VI compares various trellis-coded modulation schemes using multidimensional or 2D constellations and concludes the paper.

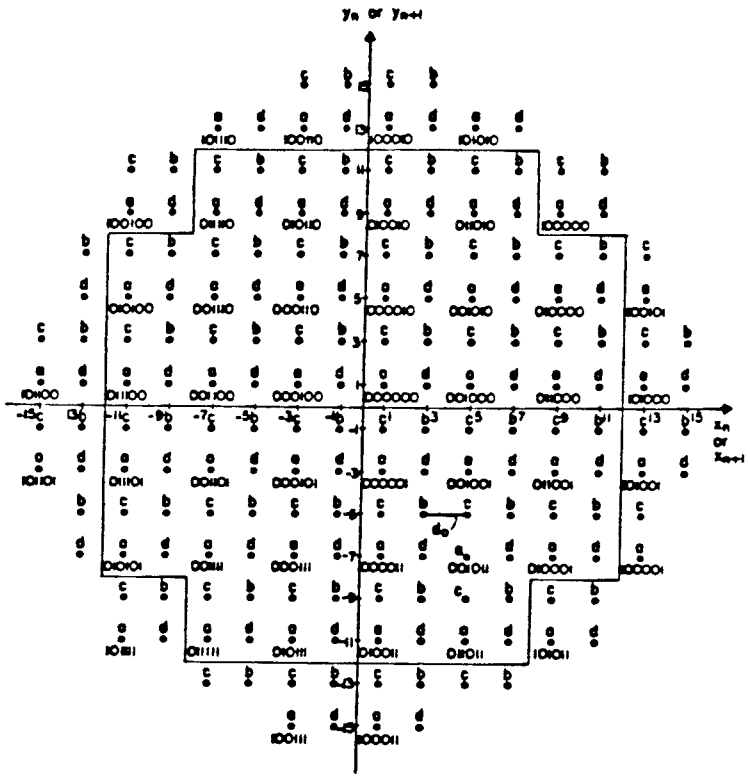
II. Partitioning of Multidimensional Lattices

In this section, we show how a multidimensional lattice may be geometrically partitioned into sublattices with enlarged intrasublattice MSED, based iteratively on a partitioning of the constituent 2D lattices. We first give an example, showing how a 4D rectangular lattice with MSED d_0^2 may be partitioned into eight sublattices with MSED $4d_0^2$. We then give general principles for partitioning multidimensional lattices. Those principles are applied to partition the 4D, 8D, and 16D rectangular lattices, the densest 4D lattice D_4, the densest 8D lattice E_8, and a previously unused 8D lattice DE_8. These partitionings will be used in Section IV to construct trellis-coded modulation schemes. The relationships between the sublattices and known lattices such as D_4 and E_8 will also be noted and exploited.

To partition the 4D rectangular lattice with MSED d_0^2 into eight 4D sublattices with MSED $4d_0^2$, each constituent 2D rectangular lattice with MSED d_0^2 is first partitioned into two 2D families $A \cup B$ and $C \cup D$ with MSED $2d_0^2$, which are further partitioned into four 2D sublattices A, B, C, and D with MSED $4d_0^2$, as shown in Fig. 1 (note that the infinite 2D rectangular lattices underlying the finite constellations of Figs. 1–3 are meant when those constellations are referred to in this section). Each 2D sublattice comprises those points designated by the same lower case letter.

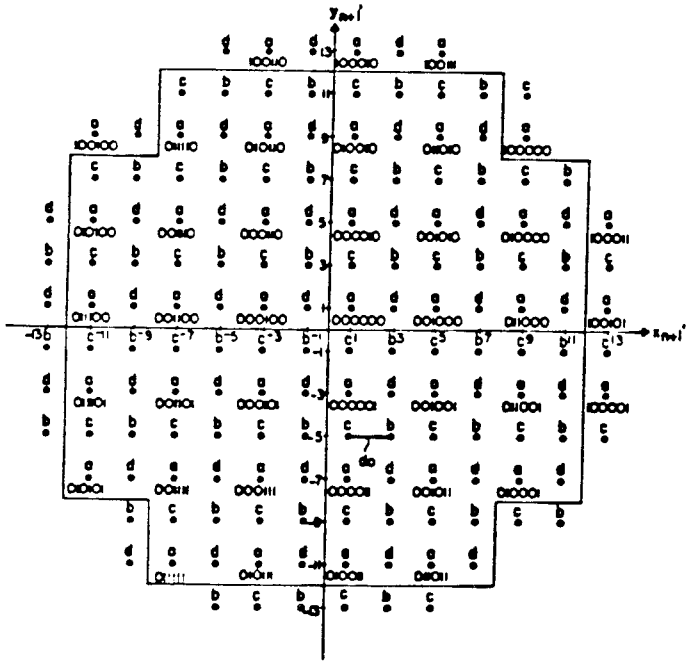
Sixteen 4D types may then be defined, each corresponding to a concatenation of a pair of 2D sublattices, and denoted as (A, A), $(A, B), \cdots$, and (D, D). The MSED of each 4D type is $4d_0^2$, the same as that of the constituent 2D sublattices. The 16 4D types may be grouped into eight 4D sublattices, denoted as 0, 1, $\cdots$, and 7, as shown in Table I. The grouping, while yielding only half as many 4D sublattices as 4D types, is done in a way which maintains the MSED of each 4D sublattice at $4d_0^2$. The advantages of grouping are that with fewer 4D sublattices, the construction of trellis codes using those sublattices is simplified, and the complexity of the corresponding maximum-likelihood decoding is reduced. Furthermore, note that the 4D sublattices of Table I are invariant under 180° rotation, while the 4D types are not. Therefore, the construction of rotationally invariant trellis codes using those sublattices does not need to consider 180° rotation and is thus simplified. The advantages of grouping will be even greater in the case of eight- or higher-dimensional lattices.

The MSED of the 4D sublattices is verified to be $4d_0^2$ as follows. The two first constituent 2D sublattices associated with the two 4D types in each 4D sublattice span a 2D family $A \cup B$ or $C \cup D$, and likewise for the two second constituent 2D sublattices associated with each 4D sublattice. Because the MSED of each 2D family is $2d_0^2$, the MSED of each 4D sublattice is $4d_0^2$.



NUMBER BENEATH EACH POINT: $Z2_{n+1}$ $Z3_{n+1}$ $Z4_{n+1}$ $Z5_{n+1}$ $Z6_{n+1}$ $Z7_{n+1}$ $(l = 0 \text{ or } 1)$

Fig. 1. 192-point 2D constellation partitioned into four subsets.



NUMBER BENEATH EACH POINT $Z2_{n+1}$ $Z3_{n+1}$ $Z4_{n+1}$ $Z5_{n+1}$ $Z6_{n+1}$ $Z7_{n+1}$ $(l = 0,1,2,3)$

Fig. 2. 160-point 2D constellation partitioned into four subsets.

The 4D sublattices are further grouped into two 4D families $U^3_{i-0}i$ and $U^7_{i-4}i$ with MSED $2d_0^2$. The two 4D families may be obtained from the two 2D families in the same way that we obtain 4D sublattices from 2D sublattices.

Before we proceed to describe the general partition principles, we clarify our terminology. A lattice is partitioned into families, subfamilies, and sublattices with (strictly) increasing MSED. Only the bottom level of a partitioning is referred to as sublattice. This level will be assigned to the state transition, or equivalently, specified by the output bits of a trellis code (see Section IV). The intermediate levels, if any, are named first as family, and then as subfamily. If there are more than two intermediate levels, then all the additional levels are referred to as subfamily. Each of these levels of a partitioning of a lattice will play a different role in our construction of trellis codes to be described in Section IV.

Associated with each partitioning of a $2N$-dimensional lattice are partitionings of its constituent N-dimensional and two-dimensional lattices into families, subfamilies, and sublattices with increasing MSED. The $2N$-, N-, and two-dimensional sublattices all have the same MSED. A $2N$-dimensional sublattice may be further partitioned into types and subtypes with the same MSED. Each $2N$-dimensional type is a concatenation of a pair of N-dimensional sublattices. Further, each $2N$-dimensional subtype is a concatenation of N two-dimensional sublattices. The $2N$-dimensional type plays an important role in the partitioning of the $2N$-dimensional lattice into sublattices, and in the corresponding maximum-likelihood decoding to be described in Section V. The $2N$-dimensional subtype plays a role in the $2N$-dimensional constellation mapping to be described in Section IV.

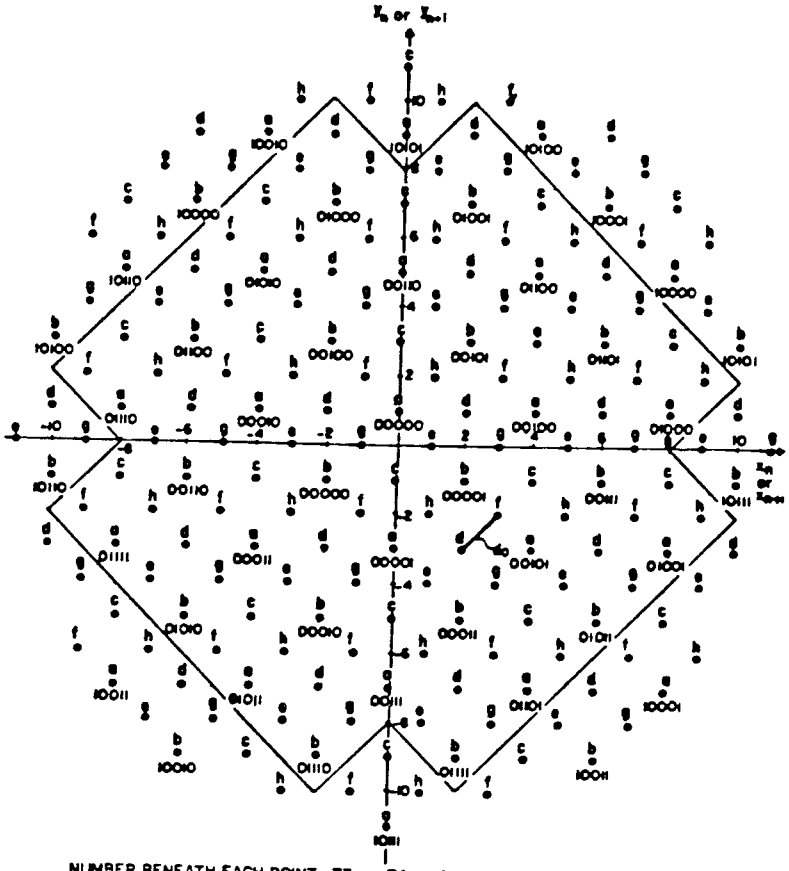


Fig. 3. 192-point 2D constellation partitioned into eight subsets.

TABLE I
EIGHT-SUBLATTICE PARTITIONING OF 4D RECTANGULAR LATTICE

4D Sublattice (Subset)	4D subset $Y0_n$	$I1_n$	$I2'_n$	$I3'_n$	4D Types	$Z0_n$	$Z1_n$	$Z0_{n+1}$	$Z1_{n+1}$
0	0	0	0	0	(A, A)	0	0	0	0
	0	0	0	1	(B, B)	0	1	0	1
1	0	0	1	0	(C, C)	1	0	1	0
	0	0	1	1	(D, D)	1	1	1	1
2	0	1	0	0	(A, B)	0	0	0	1
	0	1	0	1	(B, A)	0	1	0	0
3	0	1	1	0	(C, D)	1	0	1	1
	0	1	1	1	(D, C)	1	1	1	0
4	1	0	0	0	(A, C)	0	0	1	0
	1	0	0	1	(B, D)	0	1	1	1
5	1	0	1	0	(C, B)	1	0	0	1
	1	0	1	1	(D, A)	1	1	0	0
6	1	1	0	0	(A, D)	0	0	1	1
	1	1	0	1	(B, C)	0	1	1	0
7	1	1	1	0	(C, A)	1	0	0	0
	1	1	1	1	(D, B)	1	1	0	1

In general, the partitioning of a $2N$-dimensional lattice into families, subfamilies, and sublattices with increasing MSED may be done as follows. Suppose that the desired MSED of each $2N$-dimensional sublattice is DIST. The first step is to partition its constituent N-dimensional lattices into families, subfamilies, and sublattices with increasing MSED. Each finer partitioning of the N-dimensional lattice increases the MSED by a factor of two, with the MSED of each N-dimensional sublattice also equal to DIST. The second step is to form $2N$-dimensional types, each type corresponding to a concatenation of a pair of N-dimensional sublattices. The MSED of each $2N$-dimensional type is thus also DIST. Those $2N$-dimensional types are then grouped into $2N$-dimensional sublattices with the same MSED DIST, based on the N-dimensional subfamilies. To reduce the number of $2N$-dimensional sublattices,

we should group as many $2N$-dimensional types into a $2N$-dimensional sublattice as possible. Let us say that there are M N-dimensional sublattices in each N-dimensional subfamily. Each $2N$-dimensional sublattice then comprises M $2N$-dimensional types. The M first constituent N-dimensional sublattices of the M $2N$-dimensional types in each $2N$-dimensional sublattice span an N-dimensional subfamily, and likewise for the M second constituent N-dimensional sublattices associated with each $2N$-dimensional sublattice.

Those $2N$-dimensional sublattices are further grouped into $2N$-dimensional subfamilies and families with decreasing MSED. Each grouping reduces the MSED by a factor of two. The $2N$-dimensional subfamilies or families with a certain MSED may be obtained from the N-dimensional subfamilies or families with the same MSED by following the same principles used earlier to obtain $2N$-dimensional sublattices.

To simplify the construction of rotationally invariant trellis codes using those sublattices, the grouping of $2N$-dimensional types into sublattices should be done in such a way that each sublattice is invariant under as many rotations as possible, each rotation corresponding to a phase ambiguity of the lattice. If it is not possible to make sublattices invariant to all rotations, then each rotation should at least take a sublattice into another sublattice.

The principles just described may be used iteratively to partition a multidimensional lattice based on a partitioning of the constituent 2D lattices. For example, using those principles, the 8D rectangular lattice with MSED d_0^2 may be partitioned into 16 8D sublattices with MSED $4d_0^2$, based iteratively on the partitioning of each constituent 2D rectangular lattice into four 2D sublattices A, B, C, and D as shown in Figs. 1 and 2.

The partitioning of the 8D rectangular lattice is described in the following. The first and second constituent 2D rectangular lattices form a constituent 4D rectangular lattice. As in the example given earlier, this 4D rectangular

TABLE II
16-SUBLATTICE PARTITIONING OF 8D RECTANGULAR LATTICE

8D Sublattice (Subset)	$Y0_n$	$I1_n$	$I2_n$	$I3_n$	8D Types
0	0	0	0	0	(0,0), (1,1), (2,2), (3,3)
1	0	0	0	1	(0,1), (1,0), (2,3), (3,2)
2	0	0	1	0	(0,2), (1,3), (2,0), (3,1)
3	0	0	1	1	(0,3), (1,2), (2,1), (3,0)
4	0	1	0	0	(4,4), (5,5), (6,6), (7,7)
5	0	1	0	1	(4,5), (5,4), (6,7), (7,6)
6	0	1	1	0	(4,6), (5,7), (6,4), (7,5)
7	0	1	1	1	(4,7), (5,6), (6,5), (7,4)
8	1	0	0	0	(0,4), (1,5), (2,6), (3,7)
9	1	0	0	1	(0,5), (1,4), (2,7), (3,6)
10	1	0	1	0	(0,6), (1,7), (2,4), (3,5)
11	1	0	1	1	(0,7), (1,6), (2,5), (3,4)
12	1	1	0	0	(4,0), (5,1), (6,2), (7,3)
13	1	1	0	1	(4,1), (5,0), (6,3), (7,2)
14	1	1	1	0	(4,2), (5,3), (6,0), (7,1)
15	1	1	1	1	(4,3), (5,2), (6,1), (7,0)

TABLE III
32-SUBLATTICE PARTITIONING OF 4D RECTANGULAR LATTICE

4D Family	4D Subfamily	4D Sublattice (Subset)	$Y0_n$	$I1_n$	$I2_n$	$I3'_n$	$I4'_n$	$I5_n$	4D Types
0	0	0	0	0	0	0	0	0	(A, A)
			0	0	0	0	0	1	(B, B)
		1	0	0	0	0	1	0	(C, C)
			0	0	0	0	1	1	(D, D)
		8	0	1	0	0	0	0	(A, B)
			0	1	0	0	0	1	(B, A)
		9	0	1	0	0	1	0	(C, D)
			0	1	0	0	1	1	(D, C)
	1	2	0	0	0	1	0	0	(E, E)
			0	0	0	1	0	1	(F, F)
		3	0	0	0	1	1	0	(G, G)
			0	0	0	1	1	1	(H, H)
		10	0	1	0	1	0	0	(E, F)
			0	1	0	1	0	1	(F, E)
		11	0	1	0	1	1	0	(G, H)
			0	1	0	1	1	1	(H, G)
	2	4	0	0	1	0	0	0	(A, C)
			0	0	1	0	0	1	(B, D)
		5	0	0	1	0	1	0	(C, A)
			0	0	1	0	1	1	(D, B)
		12	0	1	1	0	0	0	(A, D)
			0	1	1	0	0	1	(B, C)
		13	0	1	1	0	1	0	(C, B)
			0	1	1	0	1	1	(D, A)
	3	6	0	0	1	1	0	0	(E, G)
			0	0	1	1	0	1	(F, H)
		7	0	0	1	1	1	0	(G, E)
			0	0	1	1	1	1	(H, F)
		14	0	1	1	1	0	0	(E, H)
			0	1	1	1	0	1	(F, G)
		15	0	1	1	1	1	0	(G, F)
			0	1	1	1	1	1	(H, E)
1	4	16	1	0	0	0	0	0	(A, E)
			1	0	0	0	0	1	(B, F)
		17	1	0	0	0	1	0	(C, G)
			1	0	0	0	1	1	(D, H)
		24	1	1	0	0	0	0	(A, F)
			1	1	0	0	0	1	(B, E)
		25	1	1	0	0	1	0	(C, H)
			1	1	0	0	1	1	(D, G)
	5	18	1	0	0	1	0	0	(E, C)
			1	0	0	1	0	1	(F, D)
		19	1	0	0	1	1	0	(G, A)
			1	0	0	1	1	1	(H, B)
		26	1	1	0	1	0	0	(E, D)
			1	1	0	1	0	1	(F, C)
		27	1	1	0	1	1	0	(G, B)
			1	1	0	1	1	1	(H, A)
	6	20	1	0	1	0	0	0	(A, G)
			1	0	1	0	0	1	(B, H)
		21	1	0	1	0	1	0	(C, E)
			1	0	1	0	1	1	(D, F)
		28	1	1	1	0	0	0	(A, H)
			1	1	1	0	0	1	(B, G)
		29	1	1	1	0	1	0	(C, F)
			1	1	1	0	1	1	(D, E)
	7	22	1	0	1	1	0	0	(E, A)
			1	0	1	1	0	1	(F, B)
		23	1	0	1	1	1	0	(G, C)
			1	0	1	1	1	1	(H, D)
		30	1	1	1	1	0	0	(E, B)
			1	1	1	1	0	1	(F, A)
		31	1	1	1	1	1	0	(G, D)
			1	1	1	1	1	1	(H, C)

lattice with MSED d_0^2 is partitioned into two 4D families $\cup_{i=0}^{3} i$ and $\cup_{i=4}^{7} i$ with MSED $2d_0^2$. Each 4D family is further partitioned into four 4D sublattices 0, 1, 2, 3, or 4, 5, 6, 7, with MSED $4d_0^2$. A second constituent 4D rectangular lattice formed by the third and fourth constituent 2D rectangular lattices is similarly partitioned.

Sixty-four 8D types are then formed, each corresponding to a pair of 4D sublattices and denoted as (0,0), (0,1),$\cdots$, and (7,7). Using the general principles described earlier, those 64 8D types may be grouped into 16 8D sublattices with MSED $4d_0^2$ and denoted as 0, 1,$\cdots$, and 15, as shown in Table II. Those 16 8D sublattices may be further grouped into two 8D families $\cup_{i=0}^{7} i$ and $\cup_{i=8}^{15} i$ with MSED $2d_0^2$.

The advantages of grouping multidimensional types into sublattices are clear in this example. Without that grouping, the 8D rectangular lattice with MSED d_0^2 might have been partitioned into 256 sublattices with MSED $4d_0^2$, each sublattice being an 8D subtype formed by concatenating 2D sublattices of Figs. 1 or 2, and denoted as (A, A, A, A), (A, A, A, B),$\cdots$, or (D, D, D, D). The task of constructing a trellis code using those 256 sublattices and the corresponding maximum-likelihood decoding would be extremely difficult if not impossible.

There are other ways to partition the 8D rectangular lattice into 16 8D sublattices to give the same distance properties as the earlier partitioning. However, the earlier partitioning has the special property that each 8D sublattice is invariant under each rotation corresponding to a phase ambiguity (90°, 180°, or 270°) of the lattice. With that property, the construction of rotationally invariant trellis codes using those 8D sublattices does not need to consider such rotations and is much simplified.

Based on the earlier partitioning of the 8D rectangular lattice, it is straightforward to show that the 16D rectangular lattice with MSED d_0^2 may be partitioned into two families with MSED $2d_0^2$, and each family may be further partitioned into 16 sublattices with MSED $4d_0^2$.

To partition a multidimensional rectangular lattice with MSED d_0^2 into sublattices with MSED greater than $4d_0^2$, a partitioning of each constituent 2D rectangular lattice into more than four 2D sublattices should be used. Table III gives a partitioning of the 4D rectangular lattice wit MSED d_0^2 into 32 sublattices with MSED $8d_0^2$, based o an eight-sublattice partitioning of the 2D rectangular lattic shown in Fig. 3 (note that the rectangular lattice in th figure is rotated by 45°).

The relationships between the sublattices, subfamilies and families of the 4D rectangular lattice shown in Table I or III with the densest 4D lattice D_4 are now exploited A translation of the 4D lattice D_4 may be defined as 4D sublattice 0 of Table I [3], [9]. With that definition of D_4, it is easy to see that each of the 4D sublattices, subfamilies, and families of Tables I or III may be interpreted as a translated, rotated, or scaled version of D_4. The partitioning of the 4D rectangular lattice may then be expressed as in Fig. 4. From that figure, we see that each time the intrasublattice (a D_4 lattice) MSED is doubled, the number of 4D sublattices is multiplied by four. Extensions to finer partitionings of the 4D rectangular lattice become obvious. Fig. 4 also says that a 4D lattice D_4 with MSED $2d_0^2$ may be partitioned into 16 sublattices with MSED $8d_0^2$, based on the eight-sublattice partitioning of its constituent 2D rectangular lattice shown in Fig. 3.

Similar relationships are exploited between the sublattices of the 8D rectangular lattice of Table II and the densest 8D lattice E_8. A translation of E_8 may be defined as 8D sublattice 0 of Table II [3], [6], [9]. With this definition of E_8, it is easy to see that other 8D sublattices of Table II are also translated or rotated versions of E_8. The partitioning of the 8D rectangular lattice of Table II may therefore be expressed as in Fig. 5, where it is also shown that each 8D family is an 8D lattice D_8 [9].

A finer partitioning of the 8D rectangular lattice with MSED d_0^2 into sublattices with MSED $8d_0^2$ will show that an 8D lattice E_8 with MSED $4d_0^2$ may be partitioned into 16 sublattices with MSED $8d_0^2$, based on the eight-sublattice partitioning of its constituent 2D rectangular lattice shown in Fig. 3. Each sublattice of E_8 can be shown to be another translated, rotated, or scaled version of E_8. The partitioning of E_8 is also shown in Fig. 5. From that figure, we see that each time the intrasublattice (an E_8 lattice) MSED is doubled, the number of 8D sublattices is multiplied by 16. Extensions to finer partitionings of the

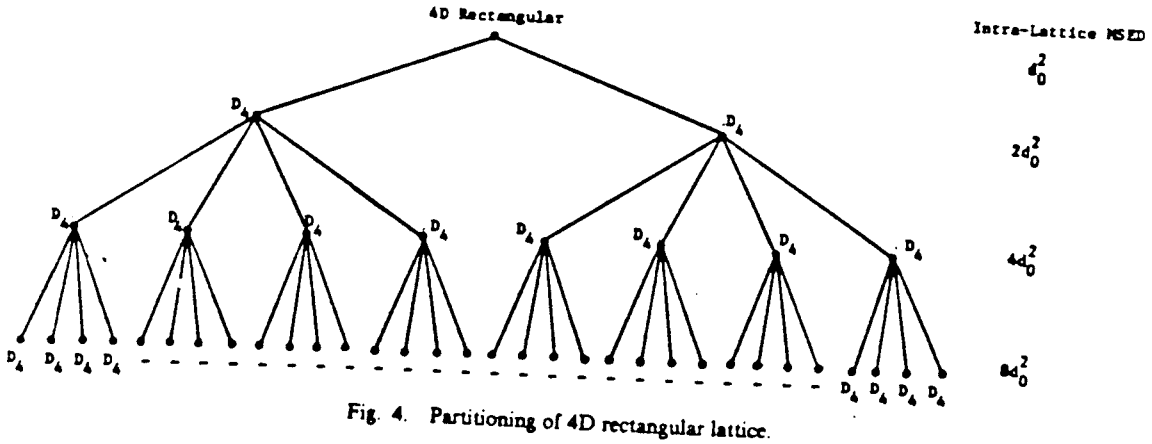


Fig. 4. Partitioning of 4D rectangular lattice.

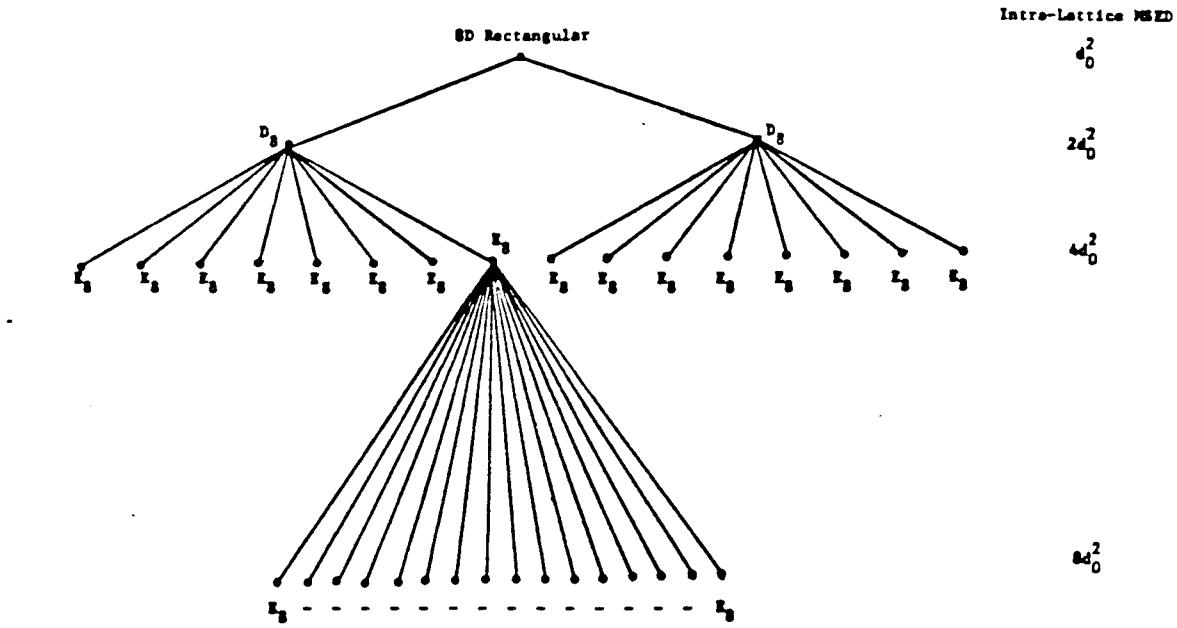


Fig. 5. Partitioning of 8D rectangular lattice.

8D rectangular lattice or the densest 8D lattice E_8 become obvious.

Finally, we define an 8D lattice DE_8 as the union of an E_8 and a rotated version of that E_8. The E_8 is 8D sublattice 0 of Table II. Furthermore, the rotated version of that E_8 is obtained by rotating the third and fourth constituent 2D points of each 8D point in that E_8 by 90° clockwise, which is 8D sublattice 1 of Table II. (Such a lattice has not to our knowledge appeared previously in the coding literature.) The MSED of the lattice DE_8 is thus $2d_0^2$. This lattice may be partitioned into two 8D families with MSED $4d_0^2$. Each family is a version of E_8 which may be further partitioned into 16 sublattices with MSED $8d_0^2$. We will use DE_8 and this 32-sublattice partitioning in a later coded-modulation scheme.

III. Construction of Multidimensional Constellations

To transmit Q information bits per signaling interval using a $2N$-dimensional trellis code of rate $m/m+1$, a $2N$-dimensional constellation of 2^{NQ+1} points is needed. In this section, we show how to construct such finite constellations from various infinite $2N$-dimensional lattices.

For small values of $NQ+1$ and a communication system where additive Gaussian noise is the only impairment, the constellation may be constructed to keep the average power as small as possible. However, for large values of $NQ+1$ or a communication system where other impairments such as linear or nonlinear distortion or phase jitter are also present, of which voiceband data transmission is an example, it is important to construct the constellation so that 1) the complicated mapping between the $NQ+1$ bits and the $2N$-dimensional constellation may be converted to N simple constituent 2D constellation mappings; 2) the size of the constituent 2D constellations is kept as small as possible; and 3) the peak-to-average power ratio is also kept as small as possible. The construction we give is similar to that used in [3] for transmitting a nonintegral number of bits per signaling interval in an uncoded scheme.

As we shall see, using this construction, the size of the constituent 2D constellations of a multidimensional constellation may be significantly smaller than that of a corresponding 2D trellis-coded modulation scheme. There is also essentially no penalty in peak-to-average power ratio from the use of a multidimensional constellation. With all of these desirable characteristics, the construction leads the way to practical applications of multidimensional trellis-coded modulation.

To appreciate the advantages of this construction, we will focus on the case where Q is equal to seven (unless otherwise specified). Such a number of bits per signaling interval may be used in voiceband modems at data rates higher than 14.4 kbit/s.

To construct a 4D constellation of 2^{15} points from the 4D rectangular lattice, we first construct its constituent 2D constellations. Fig. 1 shows a 192-point 2D constellation partitioned into four subsets A, B, C, and D. The 2D constellation includes the 128-point cross constellation (128-CR) located within the boundary shown, which is typically used in an uncoded scheme for transmitting seven bits per signaling interval. Those 128 points are called inner points. The 2D constellation also includes an outer group of 64 points, half as many as in the inner group. The outer points are selected as close to the origin as possible while satisfying the following two requirements. First, each subset A, B, C, or D has the same number of outer points. Second, if an outer point is rotated by 90°, 180°, or 270°, another outer point is obtained. The first requirement is necessary to convert the 4D constellation mapping into a pair of 2D constellation mappings. The second require-

ment preserves the symmetries of the lattice in the constellation. The two requirements should also be satisfied by the inner points, as 128-CR does.

The 4D constellation of 2^{15} points is then constructed by concatenating a pair of the 192-point 2D constellations, and excluding those 4D points whose corresponding pair of 2D points are both outer points. The average power of the 4D constellation may be determined as follows. For each constituent 192-point 2D constellation, the inner group is used three times as often as the outer group. The average power of the 4D constellation, which is also the average power of each constituent 192-point 2D constellation in this case, is thus $3/4$ times the average power of the inner points plus $1/4$ times the average power of the outer points. It is then straightforward to show that the average power of the 4D constellation is $28.0625d_0^2$. The peak power of the 4D constellation, which is also the peak power of the constituent 192-point 2D constellations, is $60.5d_0^2$. The peak-to-average power ratio of the 4D constellation is therefore 2.16, smaller than the peak-to-average power ratio, 2.33, of the 64-point 2D square constellation commonly used in an uncoded scheme for transmitting six bits per signaling interval.

The partitioning of the 4D rectangular lattice of Table I underlies the partitioning of this 4D constellation. From now on, for notational convenience, when we say that a constellation is of a certain type, we mean that the constellation is derived from a lattice of that type. For example, when we say a 4D rectangular constellation, we mean a 4D constellation derived from the 4D rectangular lattice. Furthermore, we will carry over the terminology used in the partitioning of a lattice, such as family, subfamily, type, and subtype, to the partitioning of a constellation derived from that lattice. The terminology "sublattice" will be replaced by "subset" to be consistent with previous work on partitioning 2D constellations.

In general, to transmit Q information bits per signaling interval using a rate-$m/m+1$ trellis code with a $2N$-dimensional rectangular constellation, where N is a power of two, the $2N$-dimensional constellation of 2^{NQ+1} points is constructed as follows. The first step is to obtain a constituent 2D rectangular constellation. The 2D constellation is divided into two groups, an inner group and an outer group. The number of points in the inner group is 2^Q, the same as that in the corresponding uncoded scheme. The number of points in the outer group is $1/N$ of that in the inner group. The inner group is selected first from the rectangular lattice so that the average power of the inner group is kept as small as possible. The outer group is selected from the rest of the rectangular lattice so that the average power of the outer group is minimized. The inner and outer groups must satisfy two requirements: first, each subset, obtained by partitioning the 2D constellation in accordance with the partitioning of the $2N$-dimensional constellation as described in the last section, has the same number of points in each group as other subsets; and second, each group is invariant under 90°, 180°, and 270° rotations. (Note that when a 2D rectangular constellation is partitioned into four subsets as in Fig. 1, satisfaction of the second requirement guarantees that the first requirement is also satisfied. However, this may not be the case when a 2D rectangular constellation is partitioned into more than four subsets.)

The $2N$-dimensional constellation of 2^{NQ+1} points is then constructed by concatenating N such 2D constellations, and excluding those $2N$-dimensional points corresponding to more than one 2D outer point. There are 2^{NQ} $2N$-dimensional points comprising only 2D inner points and 2^{NQ} $2N$-dimensional points comprising one 2D outer point. For each constituent 2D constellation, the inner group is used $2N-1$ times as often as the outer group. The average power of the $2N$-dimensional constellation, which is also the average power of each constituent 2D constellation in this case, is thus $(2N-1)/2N$ times the average power of the inner points plus $1/2N$ times the average power of the outer points.

Following the general principles described, an 8D rectangular constellation of 2^{29} points may be constructed from the 160-point 2D constellation shown in Fig. 2. The 2D constellation is partitioned into four subsets A, B, C, and D in accordance with the 16-subset partitioning of the 8D rectangular constellation (see Table II). The inner group is still 128-CR. The outer group has 32 points, only a quarter as many as in the inner group. The 8D constellation is formed by concatenating four such 160-point 2D constellations, and excluding those 8D points corresponding to more than one 2D outer point. The average power of this 8D constellation is $23.59375d_0^2$ with peak-to-average power ratio 2.14.

One advantage of using trellis coding with a multidimensional rectangular constellation instead of a 2D constellation becomes clear. It not only reduces the number of redundant bits but also reduces the size of the constituent 2D constellations. This is desirable especially when the size of the 2D constellation for the corresponding uncoded scheme, such as 128-CR, is already very large.

Using a multidimensional constellation has another advantage. It is easy to transmit a nonintegral number of information bits per signaling interval in such a constellation. Nonintegral numbers of bits per signaling interval poses a serious problem to any 2D modulation scheme. An unnecessarily large 2D constellation with a large peak-to-average power ratio is often required. This issue may be eliminated in a multidimensional constellation. For example, to transmit 7-1/4 information bits per signaling interval using a rate $m/m+1$ trellis code with an 8D rectangular constellation partitioned in accordance with Table II, the 8D constellation of 2^{30} points may be constructed from the 192-point 2D constellation of Fig. 1 as follows. A 4D constellation of 2^{15} points is first constructed from the 192-point 2D constellation of Fig. 1 as before. The 8D constellation of 2^{30} points is then formed by simply concatenating a pair of such 4D constellations. The average power and peak-to-average power ratio of this 8D constellation are $28.0625d_0^2$ and 2.16, respectively, the same as those of the constituent 4D constellations. The increase in

the average power of this 8D constellation from that of the previous 8D constellation of 2^{29} points is 0.75 dB, as one would expect based on the expectation that an additional information bit per signaling interval costs about an additional 3 dB of signal power [3]. Generalization of this example to other nonintegral numbers of bits or other multidimensional constellations will be reported in a companion paper.

In the remainder of this section, we shall briefly describe a few more multi-dimensional constellations to be used in the later coded modulation schemes. To partition a 4D rectangular constellation into 32 subsets as shown in Table III, each constituent 2D constellation is partitioned into eight subsets. Fig. 3 shows a 192-point 2D constellation with such a partitioning. The constellation happens to be the same as that of Fig. 1 except for a 45° rotation. A pair of such 2D constellations may be used to construct a 4D rectangular constellation of 2^{15} points with the same average power and peak-to-average power ratio as before.

A 16D rectangular constellation of 2^{57} points may be constructed from a 144-point 2D constellation with 128-CR as its inner group and only 16 points in its outer group. The average power of the 16D constellation can be shown to be $21.875d_0^2$ with peak-to-average power ratio 2.03.

In the case where a multidimensional constellation is derived from a nonrectangular lattice such as D_4, E_8, or DE_8, the construction of the nonrectangular constellation is slightly different from that of a rectangular constellation. The difference occurs because when the multidimensional nonrectangular constellation is constructed from its constituent 2D rectangular constellations, the concatenation of 2D points must be a valid point of the nonrectangular lattice.

With this difference, it can be shown that a 4D constellation D_4 of 2^{15} points may be constructed by concatenating a pair of 256-point 2D rectangular constellations. The 4D constellation has an average power $40.6875d_0^2$ and peak-to-average power ratio 1.93.

Similarly, a 320-point 2D rectangular constellation may be used to construct an 8D constellation E_8 of 2^{29} points with average power $47.133d_0^2$ and peak-to-average power ratio 2.17. The same 256-point 2D rectangular constellation used for the 4D constellation D_4 may be used to construct an 8D constellation DE_8 of 2^{29} points with the same average power and peak-to-average power ratio as for the 4D constellation D_4. Note that using an 8D constellation DE_8 rather than E_8 reduces both the size of the constituent 2D constellation and the average power of the 8D constellation.

IV. Trellis-Coded Modulation with Multidimensional Constellations

To send Q information bits per signaling interval using a rate-$m/m+1$ trellis code with a $2N$-dimensional constellation partitioned into 2^{m+1} subsets, m of the NQ information bits arriving in each block of N signaling intervals enter the trellis encoder, and the resulting $m+1$ coded bits specify which $2N$-dimensional subset is to be used. The remaining information bits specify which point from the selected $2N$-dimensional subset is to be transmitted.

In this section, we show how to construct rate-$m/m+1$ trellis codes with various partitionings of $2N$-dimensional constellations and how to select a point from a $2N$-dimensional subset; or, more generally, how to map the $NQ+1$ trellis-encoded or nontrellis-encoded bits into a $2N$-dimensional constellation. The $2N$-dimensional constellation mapping is converted to N constituent 2D constellation mappings with the assistance of a bit converter and a block encoder.

A number of trellis codes have been constructed and evaluated in terms of coding gain, error coefficient, transparency to phase ambiguities, size of constituent 2D constellations, peak-to-average power ratio, and complexity. In particular, a 16-state code with a 4D rectangular constellation partitioned as in Table I is shown in Section IV-A. Section IV-B shows a 64-state code with an 8D rectangular constellation partitioned as in Table II. These two codes are optimal in the sense that, given the constellation and its partitioning, each code achieves both the largest possible coding gain and the smallest possible error coefficient with the smallest number of states. Both codes are transparent to all phase ambiguities (90°, 180°, and 270°) of their constellations.

To increase the coding gain further using a 4D rectangular constellation, a finer partitioning of the 4D constellation as shown in Table III should be used. Section IV-C shows a 64-state code with this partitioning of the 4D rectangular constellation. This is the smallest number of states that can be used to realize the larger coding gain promised by the finer partitioning of the 4D constellation. This 64-state code is also transparent to all phase ambiguities of the constellation.

In Section IV-D, we extend our study to other codes with 4D, 8D, and 16D rectangular constellations, D_4, E_8, and DE_8. Again, as in the previous section, we will focus on the case where the number Q of information bits transmitted per signaling interval is seven unless otherwise specified. The seven information bits arriving in the current signaling interval n are denoted as $I1_n$, $I2_n$, $\cdots$, and $I7_n$.

A. 16-State Code with 4D Rectangular Constellation

A rate-2/3, 16-state code with a 4D rectangular constellation of 2^{15} points is shown in Fig. 6. The 4D constellation is constructed from the 192-point 2D constellation of Fig. 1 as in the last section and is partitioned into eight subsets as in Table I. The three output bits $Y0_n$, $I1_n$, and $I2'_n$ of the trellis encoder are associated with the 4D subsets in accordance with Table I.

If we denote the current and next states of the trellis encoder as $W1_pW2_pW3_pW4_p$, $p=n$ and $n+2$, the trellis diagram is as shown in Fig. 7. The association of 4D subsets with the state transitions of Fig. 7 satisfies the following three requirements: 1) the 4D subsets associated

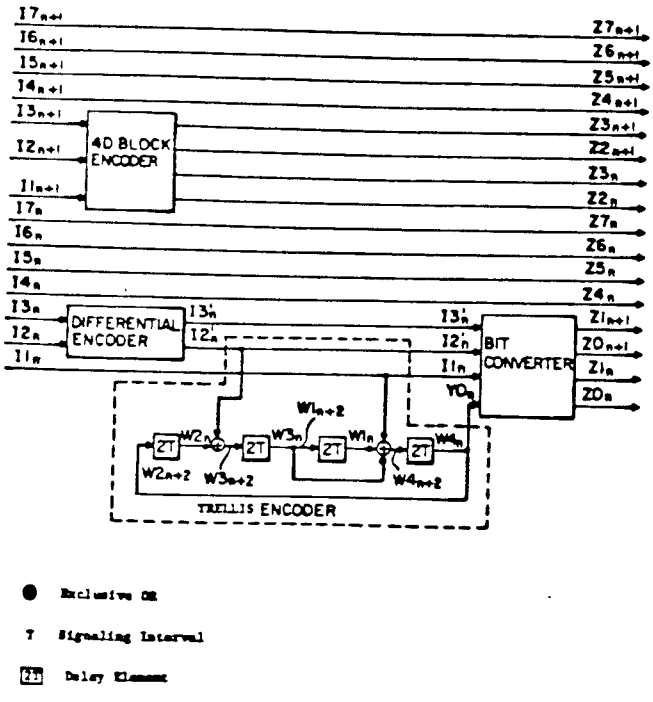


Fig. 6. 16-state code with 4D rectangular constellation.

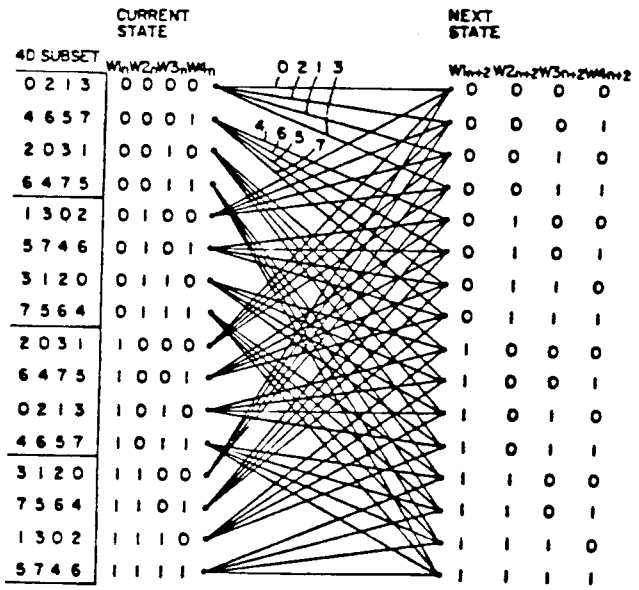


Fig. 7. Trellis diagram of 16-state code of Fig. 6.

with the transitions leading from a state are different from each other and belong to the same 4D family $U_{i-0}^{3}i$ or $U_{i-4}^{7}i$ (see Section II), and likewise for the 4D subsets associated with the transitions leading to a state; 2) the MSED between two allowed sequences of 4D subsets corresponding to two distinct trellis paths is larger than $4d_0^2$, which is the MSED of each 4D subset; and 3) a one-to-one function F that maps each state of the trellis encoder into another state may be defined so that the following statement is valid Denote X as the 4D subset associated with the transition from a current state i to a next state j, and Y as the 4D subset obtained when X is rotated by 90° clockwise. Then Y is associated with the transition from the current state $F(i)$ to the next state $F(j)$. The function F for this code is

$$F\colon W1_pW2_pW3_pW4_p \rightarrow \overline{W1}_pW2_p\overline{W3}_pW4_p$$

where an overbar denotes inversion.

That the second requirement above is satisfied may be seen as follows. Referring to Fig. 7, for each current state $W1_nW2_nW3_nW4_n$, the four possible next states are $W3_nW4_nX1X2$, where $X1X2 = 00, 01, 10$, or 11. All transitions originating from even-numbered states (states with $W4_n$ equal to zero) are associated with 4D subsets from the first 4D family $U_{i-0}^{3}i$, while all transitions originating from odd-numbered states (states with $W4_n$ equal to one) are associated with 4D subsets from the second 4D family $U_{i-4}^{7}i$. Furthermore, if Y is the 4D subset associated with the transition from a current state $W1_nW2_nW3_nW4_n$ to an even-numbered (or odd-numbered) next state, then Y is also the 4D subset associated with the transition from the current state $W1_nW2_n\overline{W3_n}W4_n$ to an odd-numbered (or even-numbered) next state.

The first requirement guarantees that the MSED between any two allowed sequences of 4D points is $4d_0^2$. The coding gain of the code over the uncoded 128-CR therefore is

$$10\log_{10}\left(\frac{4d_0^2}{28.0625d_0^2}\Big/\frac{d_0^2}{20.5d_0^2}\right) = 4.66 \text{ dB},$$

where $28.0625d_0^2$ is the average power of the 4D constellation as determined in the last section, and $20.5d_0^2$ is the average power of 128-CR. This is also the largest possible coding gain that can be achieved with the partitioning of the 4D rectangular constellation of Table I. This coding gain may be viewed as the combination of a gain of 6.02 dB from the trellis code if the 4D constellation were not expanded from 2^{14} to 2^{15} points, and a loss of 1.36 dB due to that expansion. The expansion loss is less than the 3 dB loss of a 2D rate-$m/m+1$ trellis code, as promised by the use of a 4D constellation.

The second requirement above eliminates MSED error events which differ in more than one 4D point from a given sequence of 4D points. The error coefficient of the code is thus minimized to 24 per 4D point (equivalent to 12 per 2D point), which is the number of nearest neighbors to any point in the same 4D subset (a D_4 lattice). Taking into account the boundary effect of the finite constellation would reduce this value.

The third requirement guarantees that the code can be made transparent to all phase ambiguities (90°, 180°, and 270°) of the constellation. Since the same 4D subset is obtained when a 4D subset is rotated by 180°, the construction of the trellis code needs to take into account only 90° rotation. The 270° rotation is then taken care of automatically.

We now show how to map the three trellis-encoded bits and the remaining 12 non-trellis-encoded information bits into the 4D constellation. Referring to Table I, after using the three trellis-encoded bits to specify a 4D subset, a

fourth nontrellis-encoded information bit $I3'_n$ is used to specify a 4D type within the 4D subset. To make the scheme transparent to all phase ambiguities of the constellation, the association of the three trellis-encoded bits $Y0_n$, $I1_n$, and $I2'_n$, and the fourth uncoded bit $I3'_n$, with 4D types is done as follows. For each pattern $Y0_n I1_n I2'_n I3'_n$, denote X as the associated 4D type. Denote $X1$, $X2$, and $X3$ as the 4D types obtained when the 4D type X is rotated by 90°, 180°, and 270° clockwise, respectively. Denote $S3_1S2_1$, $S3_2S2_2$, and $S3_3S2_3$ as the bit pairs obtained when the bit pair $I3'_nI2'_n$ is advanced by one, two, and three positions, respectively, in a circular sequence 00, 01, 10, 11. Then the 4D types associated with the bit patterns $Y0_nI1_nS2_1S3_1$, $Y0_nI1_nS2_2S3_2$, and $Y0_nI1_nS2_3S3_3$ are $X1$, $X2$, and $X3$, respectively.

As a first step of the 4D constellation mapping, a bit converter converts the four bits $Y0_n$, $I1_n$, $I2'_n$, and $I3'_n$ into two pairs of selection bits, $Z0_nZ1_n$ and $Z0_{n+1}Z1_{n+1}$, which are used to select the pair of 2D subsets corresponding to the 4D type. With the correspondence between the bit pair $Z0_pZ1_p$ and 2D subsets A, B, C, and D as shown in Table IV, the operation of the bit converter is shown in Table I.

TABLE IV
CORRESPONDENCE BETWEEN $Z0_pZ1_p$ AND FOUR 2D SUBSETS

2D Subset	$Z0_pZ1_p$
A	00
B	01
C	10
D	11

A 4D block encoder then takes three of the remaining eleven uncoded information bits, $I1_{n+1}$, $I2_{n+1}$, and $I3_{n+1}$, and generates two pairs of selection bits, $Z2_nZ3_n$ and $Z2_{n+1}Z3_{n+1}$ in accordance with Table V. Each of the bit pairs can assume any of the values 00, 01, or 10, but they cannot both assume the value 10. The first pair $Z2_nZ3_n$ will be used to select the inner group or outer group of the first selected 2D subset, and likewise for the second pair with respect to the second 2D subset. The inner group is organized into two halves. If the bit pair is 00, one-half of the inner group is selected; if the bit pair is 01, the other half of the inner group is selected; otherwise the outer group is selected.

TABLE V
4D BLOCK ENCODER

$I1_{n+1}$	$I2_{n+1}$	$I3_{n+1}$	$Z2_n$	$Z3_n$	$Z2_{n+1}$	$Z3_{n+1}$
0	0	0	0	0	0	0
0	0	1	0	0	0	1
0	1	0	0	0	1	0
0	1	1	0	1	1	0
1	0	0	1	0	0	0
1	0	1	1	0	0	1
1	1	0	0	1	0	0
1	1	1	0	1	0	1

There are 16 2D points in the outer group or in either half of the inner group of a 2D subset, and eight uncoded information bits remain for selecting from among those 2D points. Those eight bits are taken in two groups of four bits each and are renamed as $Z4_pZ5_pZ6_pZ7_p$, $p = n$ and $n+1$. The first group $Z4_nZ5_nZ6_nZ7_n$ is used to select a 2D point from the previously selected outer group or the selected half of the inner group of the first 2D subset, and likewise for the second group $Z4_{n+1}Z5_{n+1}Z6_{n+1}Z7_{n+1}$.

To make the scheme transparent to all phase ambiguities of the constellation, the association of the bit group $Z2_pZ3_pZ4_pZ5_pZ6_pZ7_p$, $p = n$ or $n+1$, with 2D points should be such that the same bit pattern of the group is associated with each of the four 2D points which can be obtained from each other through 90° rotations. In Fig. 1, each point in subset A is associated with a bit pattern of $Z2_pZ3_pZ4_pZ5_pZ6_pZ7_p$. The bit pattern associated with points in other subsets can be obtained by following the above rule.

To summarize, the bit converter and the 4D block encoder take the three trellis-encoded bits and the 12 remaining uncoded information bits and produce two groups of eight selection bits each, $Z2_pZ3_pZ4_pZ5_pZ6_pZ7_pZ0_pZ1_p$, $p = n$ and $n+1$. The first group is then used to address a 2D mapping table to obtain a first 2D point. The table may be constructed from Fig. 1 and Table IV. The second group addresses the same 2D mapping table to obtain the second 2D point. The 4D point corresponding to the pair of 2D points is the one selected for transmission.

That the scheme is transparent to all the phase ambiguities of the constellation may be seen as follows. If we translate a sequence of the bit pairs $I3'_nI2'_n$ appearing at the inputs of the trellis encoder and the bit converter all by the same number of positions, one, two, or three, in a circular sequence, 00, 01, 10, 11, then the sequence of 2D points produced by the 4D constellation mapping procedure will be rotated by 90°, 180°, and 270° clockwise, respectively. Therefore, a differential encoder of the form

$$I3'_nI2'_n = (I3'_{n-2}I2'_{n-2} + I3_nI2_n) \bmod 100_{\text{base 2}}$$

in Fig. 6 and a corresponding differential decoder of the form

$$I3_nI2_n = (I3'_nI2'_n - I3'_{n-2}I2'_{n-2}) \bmod 100_{\text{base 2}}$$

at the output of the trellis decoder will remove all the phase ambiguities of the constellation.

Two general principles in constructing a trellis code with a multidimensional constellation may be extracted here. The first principle says that the intersubset MSED of the multidimensional subsets associated with transitions originating from each state of the trellis encoder should be kept as large as possible, and likewise for the multidimensional subsets associated with transitions leading to each state. This principle is also used in [1] for constructing a 2D trellis code.

The second principle says that for each of those phase ambiguities of the constellation such that the multidimensional subsets are not invariant under the corresponding

rotations, it should be possible to define a one-to-one function F which maps each state of the trellis encoder into another state so that the following statement is valid. Let X be the multidimensional subset associated with the transition from a current state i to a next state j. Let Y be the multidimensional subset obtained when X is rotated by a number of degrees corresponding to that phase ambiguity. Then Y is the multidimensional subset associated with the transition from the current state $F(i)$ to the next state $F(j)$. The second principle is also used in [2] for constructing a rotationally invariant 2D trellis code.

B. 64-State Code with 8D Rectangular Constellation

A rate-3/4, 64-state code with an 8D rectangular constellation of 2^{29} points is shown in Fig. 8. The 8D constellation is constructed from the 160-point 2D constellation of Fig. 2 as in the last section and is partitioned into sixteen subsets as in Table II. The association of the four trellis-encoded bits $Y0_n$, $I1_n$, $I2_n$, and $I3_n$ with the 8D subsets is also shown in Table II.

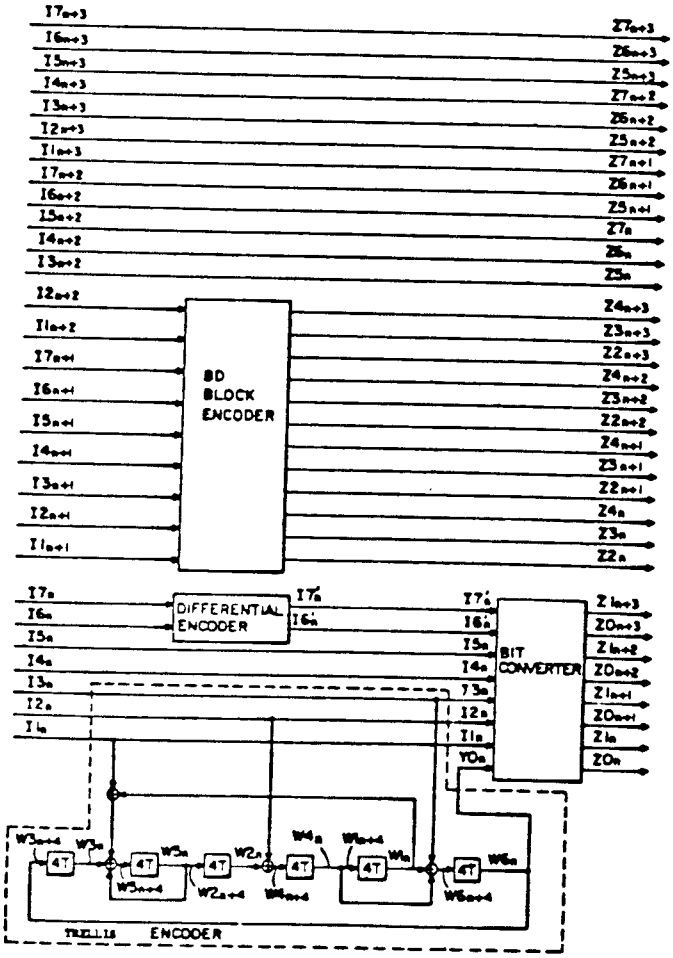


Fig. 8. 64-state code with 8D rectangular constellation.

The association of the 8D subsets with the state transitions of the trellis code satisfies the first general principle described at the end of the last subsection. Therefore the MSED between two allowed sequences of 8D points is $4d_0^2$, and the coding gain over the uncoded 128-CR is

$$10\log_{10}\left(\frac{4d_0^2}{23.59375d_0^2}\Big/\frac{d_0^2}{20.5d_0^2}\right) = 5.41 \text{ dB}$$

where $23.59375d_0^2$ is the average power of the 8D constellation as determined in the last section. This is also the largest possible coding gain that can be achieved with the partitioning of the 8D rectangular constellation of Table II. This coding gain may be segregated into two parts, a gain of 6.02 dB from the trellis code if the 8D constellation were not expanded from 2^{28} to 2^{29} points, and a loss of 0.61 dB due to that expansion. This expansion loss is less than the 1.36 dB loss for the 4D code in the last subsection.

There is no need to consider any of the phase ambiguities (90°, 180°, and 270°) of the 8D rectangular constellation in the construction of a rotationally invariant trellis code with the partitioning of the constellation of Table II because each 8D subset is invariant under rotations corresponding to those phase ambiguities.

The association of the 8D subsets with the state transitions also meets the following requirement. The MSED between two valid sequences of 8D subsets corresponding to two distinct trellis paths is larger than $4d_0^2$, which is the MSED of each 8D subset. The error coefficient of the code is thus minimized to 240 per 8D point (equivalent to 60 per 2D point), which is the number of nearest neighbors to any point in the same 8D subset (an E_8 lattice).

Referring to Fig. 8, after an 8D subset is selected by the four trellis-encoded bits, another four nontrellis-encoded information bits $I4_n$, $I5_n$, $I6'_n$, and $I7'_n$ are used to select an 8D subtype within the 8D subset (see Section II). To make the scheme transparent to all phase ambiguities of the constellation, the association of the four uncoded information bits with the 8D subtypes meets the following requirement. For each 8D subset, let X be the 8D subtype associated with a bit pattern of $I4_nI5_nI6'_nI7'_n$. Let $X1$, $X2$, and $X3$ be the 8D subtypes obtained when X is rotated by 90°, 180°, and 270° clockwise, respectively. Denote $S6_1S7_1$, $S6_2S7_2$, and $S6_3S7_3$ as the bit pairs obtained when the bit pair $I6'_nI7'_n$ is advanced by one, two, and three positions, respectively, in a circular sequence 00, 01, 10, 11. Then the 8D subtypes associated with the bit patterns $I4_nI5_nS6_1S7_1$, $I4_nI5_nS6_2S7_2$, and $I4_nI5_nS6_3S7_3$ are $X1$, $X2$, and $X3$, respectively. The 8D subtype selection procedure shown in Fig. 9 meets the above requirement.

To map the four trellis-encoded bits and the 25 remaining nontrellis-encoded information bits into the 8D constellation, a bit converter and an 8D block encoder are used to convert those 29 bits into four groups of eight selection bits each, $Z2_pZ3_pZ4_pZ5_pZ6_pZ7_pZ0_pZ1_p$, $p = n$, $n+1$, $n+2$, and $n+3$. Each group is then used to address the same 2D mapping table to obtain a 2D point. The 8D point corresponding to those four 2D points is the one selected for transmission. The 2D mapping table may be constructed from Fig. 2 and Table IV. In Fig. 2, the same six-bit value is associated with each of the four points

STEP 1:
USE $Y0_n$ $I1_n$ $I2_n$ $I3_n$ TO OBTAIN AN 8D SUBTYPE

$Y0_n$	$I1_n$	$I2_n$	$I3_n$	8D SUBTYPE
0	0	0	0	(A A A A)
0	0	0	1	(A A C C)
0	0	1	0	(A A A B)
0	0	1	1	(A A C D)
0	1	0	0	(A C A C)
0	1	0	1	(A C C B)
0	1	1	0	(A C A D)
0	1	1	1	(A C C A)
1	0	0	0	(A A A C)
1	0	0	1	(A A C B)
1	0	1	0	(A A A D)
1	0	1	1	(A A C A)
1	1	0	0	(A C A A)
1	1	0	1	(A C C C)
1	1	1	0	(A C A B)
1	1	1	1	(A C C D)

STEP 2:
ROTATE THE {3RD & 4TH; 2ND & 4TH; 2ND & 3RD} 2D SUBSETS OF THE 8D SUBTYPE BY 180° IF $I4_nI5_n$ = {01; 10; 11}

STEP 3:
ROTATE ALL FOUR 2D SUBSETS OF THE 8D SUBTYPE OBTAINED IN STEP 2 BY {90°; 180°; 270°} CLOCKWISE IF $I6'_nI7'_n$ = {01; 10; 11}

Fig. 9. 8D subtype selection procedure for 64-state code of Fig. 8.

which can be obtained from each other through 90° rotations.

The table for the bit converter can be obtained from Fig. 9 and Table IV. The operation of the 8D block encoder is shown in Fig. 10. It takes nine uncoded information bits and generates four groups of three selection bits each. Each bit group can assume any of the values 000, 001, 010, 011, and 100, but at most one of the bit groups can assume the value 100. Each bit group is used to select the inner or outer group of points of a 2D subset corresponding to the previously selected 8D subtype. If the value of the bit group is 000, 001, 010, or 011, a quarter of the inner group of points is selected; if it is 100, the outer group of points is selected.

The differential encoder in Fig. 8 has the form

$$I6'_nI7'_n = (I6'_{n-4}I7'_{n-4} + I6_nI7_n) \bmod 100_{base\,2}.$$

It is straightforward to show that the scheme in Fig. 8 is transparent to all phase ambiguities of the constellation.

C. 64-State Code with 4D Rectangular Constellation

To increase the coding gain further using a 4D rectangular constellation, a finer partitioning of the 4D constellation as shown in Table III should be used. A rate-4/5, 64-state code with a 4D rectangular constellation of 2^{15} points is shown in Fig. 11. The 4D constellation is constructed from the 192-point 2D constellation of Fig. 3 as in the last section. The association of the five trellis-encoded bits $Y0_n$, $I1_n$, $I2_n$, $I3'_n$, and $I4'_n$, and a sixth nontrellis-encoded information bit, $I5_n$, with the 4D types is shown in Table III. The 4D constellation mapping is converted into a pair of constituent 2D constellation mappings in a manner similar to that in the previous two subsections.

The MSED between two allowed sequences of 4D points of this code may be derived from the trellis diagram of the code. The trellis diagram satisfies the first general principle described in Section IV-A. The MSED of the code therefore is at least $4d_0^2$. For each current state $W1_nW2_nW3_nW4_nW5_nW6_n$, the sixteen possible next states are $W5_nW6_nX1X2X3X4$ where $X1$, $X2$, $X3$, and $X4$ are binary variables. All the transitions originating from even-numbered states (states with $W6_n$ equal to 0) are associated with 4D subsets from 4D family 0 (see Table III). All the transitions originating from odd-numbered states (states with $W6_n$ equal to one) are associated with 4D subsets from 4D family 1. Denote Y as the 4D subset associated with the transition from a current state $W1_nW2_nW3_nW4_nW5_nW6_n$ to a next state $W5_nW6_nX1X2X3X4$. Then Y and the three 4D subsets associated with the transitions from the current state $W1_nW2_nW3_nW4_nW5_nW6_n$ to the three next states $W5_nW6_nX5X6X3X4$, where $X5$ and $X6$ are binary variables and $X5X6$ is not equal to $X1X2$, all belong to the same 4D subfamily (see Table III). Furthermore, if Y is the 4D subset associated with the transition from a current state $W1_nW2_nW3_nW4_nW5_nW6_n$ to an even-numbered (or odd-numbered) next state, then for each of the four values of a bit pair $X1X2$, Y is also the 4D subset associated with the transition from the current state $W1_nW2_nX1X2\overline{W5_n}$ $W6_n$ to an odd-numbered (or even-numbered) next state.

$I1_{n+1}I2_{n+1}I3_{n+1}$	$I2_nI3_nI4_n$	$I2_{n+1}I3_{n+1}I4_{n+1}$	$I2_{n+2}I3_{n+2}I4_{n+2}$	$I2_{n+3}I3_{n+3}I4_{n+3}$
0 x x	0 $I2_{n+1}I3_{n+1}$	0 $I4_{n+1}I5_{n+1}$	0 $I6_{n+1}I7_{n+1}$	0 $I1_{n+2}I2_{n+2}$
1 0 0	1 0 0	0 $I4_{n+1}I5_{n+1}$	0 $I6_{n+1}I7_{n+1}$	0 $I1_{n+2}I2_{n+2}$
1 0 1	0 $I4_{n+1}I5_{n+1}$	1 0 0	0 $I6_{n+1}I7_{n+1}$	0 $I1_{n+2}I2_{n+2}$
1 1 0	0 $I4_{n+1}I5_{n+1}$	0 $I6_{n+1}I7_{n+1}$	1 0 0	0 $I1_{n+2}I2_{n+2}$
1 1 1	0 $I4_{n+1}I5_{n+1}$	0 $I6_{n+1}I7_{n+1}$	0 $I1_{n+2}I2_{n+2}$	1 0 0

Fig. 10. 8D block encoder.

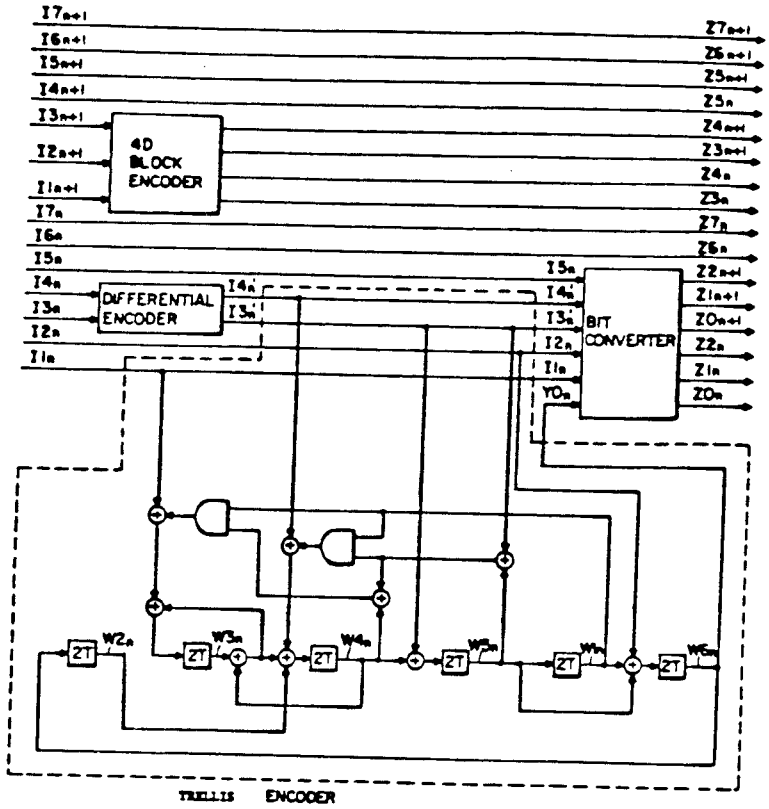


Fig. 11. 64-state code with 4D rectangular constellation.

The above statements together guarantee that the MSED of the code is at least $5d_0^2$. Since there exists an error event with a squared Euclidean distance $5d_0^2$, the coding gain of this code over the uncoded 128-CR is 5.63 dB. This coding gain may be segregated into two parts, a gain of 6.99 dB from the trellis code if the 4D constellation were not expanded from 2^{14} to 2^{15} points, and a loss of 1.36 dB due to that expansion. With some straightforward calculation, the error coefficient of the code may be shown to be 144 per 4D point, equivalent to 72 per 2D point.

Since the 4D subsets are not invariant under any of the rotations corresponding to the three phase ambiguities of the constellation, three one-to-one functions $F1$, $F2$, and $F3$, which map each state of the trellis encoder into another state and correspond to 90°, 180°, and 270° rotations, respectively, are needed as required by the second general principle of Section IV-A. Those three functions are defined as follows:

$$F1: W1_pW2_pW3_pW4_pW5_pW6_p \rightarrow \overline{W1}_pW2_pX3W4_pX5W6_p$$

$$F2: W1_pW2_pW3_pW4_pW5_pW6_p \rightarrow W1_pW2_p\overline{W3}_pW4_pW5_pW6_p$$

$$F3: W1_pW2_pW3_pW4_pW5_pW6_p \rightarrow \overline{W1}_pW2_pY3W4_pY5W6_p$$

where

$$X3X5 = (W3_pW5_p + 01) \bmod_{\text{base 2}}$$

and

$$Y3Y5 = (W3_pW5_p + 11) \bmod_{\text{base 2}}.$$

The differential encoder of Fig. 11 is the same as that of Fig. 6 except that its input and output bit pairs are now named as $I4_nI3_n$ and $I4'_nI3'_n$ rather than $I3_nI2_n$ and $I3'_nI2'_n$, respectively.

D. Extensions

The number of states of the 16-state 4D code of Section IV-A or 64-state 8D code of Section IV-B may be reduced without sacrificing the coding gain, but the error coefficient is increased. For example, a 32-state code with the same partitioning of the 8D rectangular constellation as in Section IV-B can provide 5.41 dB coding gain, the same as the 64-state code of that subsection, but its error coefficient is 124 per 2D point, compared to 60 for the 64-state code. The smallest number of states required for a code with the partitioning of the 4D rectangular constellation of Section IV-A to provide 4.66 dB coding gain is eight, the same as the number of 4D subsets of that partitioning. The smallest number of states required for a code with the partitioning of the 8D rectangular constellation of Section IV-B to provide 5.41 dB coding gain is 16, again the same as the number of 8D subsets of the partitioning.

Given the partitioning of the 4D or 8D rectangular constellation of Section IV-A or -B, it is impossible to increase the coding gain further or reduce the error coefficient of the 16-state or 64-state code of those two subsections. With a finer partitioning of the 4D or 8D rectangular constellation based on the same partitioning of the constituent 2D constellations, however, it becomes possible to reduce the error coefficient further by increasing the

number of states. For example, a 32-state code with the 4D rectangular constellation partitioned into 16 subsets, each subset corresponding to a 4D type of Table I, can provide 4.66 dB coding gain, the same as that of the 16-state code of Section IV-A, but the error coefficient is four per 2D point, less than the 12 of the 16-state code and the same as that for the uncoded 128-CR. Note that this kind of partitioning of a multidimensional lattice may be easily derived from a coarser partitioning of the same lattice obtained by following the principles described in Section II.

To increase further the coding gain of the 16-state or 64-state code of Sections IV-A or -B, a finer partitioning of the 4D or 8D constellation based on a finer partitioning of the constituent 2D constellations should be used. The 64-state code of Section IV-C is an example. Note that the number of states of that code cannot be reduced without sacrificing the coding gain. On the other hand, the coding gain can be further increased, or the error coefficient reduced, using the same partitioning of the 4D rectangular constellation as in Section IV-C, if the number of states is increased beyond 64. The upper limit on the coding gain in this case is 7.67 dB, and the lower limit on the error coefficient in the case of 7.67 dB coding gain is 12 per 2D point.

With a 16D rectangular constellation of 2^{57} points constructed from a 144-point 2D constellation and partitioned into 32 subsets as in the previous sections, it can be shown that a 32-state code can be constructed to achieve a MSED of $4d_0^2$. The coding gain of this code over the uncoded 128-CR is 5.74 dB. This is also the largest possible coding gain that can be achieved with the 32-subset partitioning of the 16D rectangular constellation. This coding gain may be segregated into two parts, a gain of 6.02 dB from the trellis code if the 16D constellation were not expanded from 2^{56} to 2^{57} points, and a loss of 0.28 dB due to that expansion. This expansion loss is only 0.33 dB less than that for the 8D code of Section IV-B. The error coefficient of this code is larger than 1000 per 2D point. This number can be reduced to 412 if the number of states is increased to 128. The lower limit on the error coefficient using the 32-subset partitioning of the 16D constellation is 284 per 2D point, which is 1/8 the number (2272) of nearest neighbors to any point in the same 16D subset.

Trellis codes using an 8D constellation E_8 of 2^{29} points constructed from a 320-point 2D constellation and partitioned into 16 subsets as in the previous sections are described next. Although the partitioning of E_8 is such that there is no 8D family in between the constellation and the 8D subsets (see Fig. 5), we group the 16 8D subsets into two equal-sized 8D families in an arbitrary manner. The intrafamily MSED is therefore the same as the MSED of the constellation. Now note that the distance relationship between an 8D family and its eight subsets of the constellation E_8 is identical to that between an 8D family and its eight subsets of the 8D rectangular constellation partitioned in accordance with Table II. Since the construction of a trellis code with that partitioning of the 8D rectangular constellation does not depend on the interfamily MSED, all the trellis codes constructed for the 8D rectangular constellation can be used for the constellation E_8.

In particular, the rate-3/4, 64-state code of Section IV-B can be used with E_8 to achieve a MSED of $8d_0^2$, which is the MSED of each 8D subset. Since the average power of the constellation E_8 is $47.133d_0^2$, the coding gain of this code over the uncoded 128-CR is 5.41 dB. This is also the largest possible coding gain that can be achieved with the 16-subset partitioning of E_8. This coding gain may be segregated into three parts: a gain of 3.04 dB from the use of E_8 [3], another gain of 3.01 dB from the trellis code if the constellation E_8 were not expanded from 2^{28} to 2^{29} points, and a loss of 0.64 dB due to that expansion. The error coefficient of this code is the minimum, 240 per 8D point (or 60 per 2D point), which is the number of nearest neighbors to any point in the same 8D subset (an E_8 lattice). Note that both the coding gain and error coefficient of this code are the same as those of the 64-state code with the 8D rectangular constellation of Section IV-B.

The fact that the large MSED of E_8 does not help a trellis code achieve additional coding gain suggests the consideration of the 8D lattice DE_8. The MSED of the lattice DE_8 is only $2d_0^2$. An 8D constellation DE_8 of 2^{29} points can be constructed from a 256-point 2D constellation with an average power $40.6875d_0^2$, as shown in Section III. Both the size of the constituent 2D constellation and the average power of the constellation DE_8 are less than those of an 8D constellation E_8 of equal size. The constellation DE_8 is partitioned into two families of 16 subsets each as in Section II. The intrafamily and intrasubset MSED's are $4d_0^2$ and $8d_0^2$, respectively, which are the same as those of the partitioning of E_8 described above. A rate-4/5, 32-state code can be constructed with the constellation DE_8 to achieve an MSED of $8d_0^2$. The coding gain of this code over the uncoded 128-CR is therefore 6.05 dB. This is also the largest possible coding gain that can be achieved with the 32 subsets of the constellation DE_8. With some calculation, this coding gain may be segregated into three parts: a gain of 0.93 dB from the use of DE_8, another gain of 6.02 dB from the trellis code if the constellation DE_8 were not expanded from 2^{28} to 2^{29} points, and a loss of 0.9 dB due to that expansion. Comparing this segregation of coding gain with that for the constellation E_8, one can see that although the lattice DE_8 is a poor lattice for block-coded modulation, it is a good lattice for trellis-coded modulation. The error coefficient of this code is more than 500 per 2D point. This number can be reduced to 316 and 124 if the number of states is increased to 64 and 128, respectively. Further improvement is certainly possible by further increasing the number of states.

A recent report [16] gives trellis-coded modulation schemes based on a partitioning of the 4D lattice D_4, the most complex using a 64-state trellis code. Similar schemes may be constructed using the techniques of this paper, with a 4D constellation D_4 of 2^{15} points constructed from

number of states. For example, a 32-state code with the 4D rectangular constellation partitioned into 16 subsets, each subset corresponding to a 4D type of Table I, can provide 4.66 dB coding gain, the same as that of the 16-state code of Section IV-A, but the error coefficient is four per 2D point, less than the 12 of the 16-state code and the same as that for the uncoded 128-CR. Note that this kind of partitioning of a multidimensional lattice may be easily derived from a coarser partitioning of the same lattice obtained by following the principles described in Section II.

To increase further the coding gain of the 16-state or 64-state code of Sections IV-A or -B, a finer partitioning of the 4D or 8D constellation based on a finer partitioning of the constituent 2D constellations should be used. The 64-state code of Section IV-C is an example. Note that the number of states of that code cannot be reduced without sacrificing the coding gain. On the other hand, the coding gain can be further increased, or the error coefficient reduced, using the same partitioning of the 4D rectangular constellation as in Section IV-C, if the number of states is increased beyond 64. The upper limit on the coding gain in this case is 7.67 dB, and the lower limit on the error coefficient in the case of 7.67 dB coding gain is 12 per 2D point.

With a 16D rectangular constellation of 2^{57} points constructed from a 144-point 2D constellation and partitioned into 32 subsets as in the previous sections, it can be shown that a 32-state code can be constructed to achieve a MSED of $4d_0^2$. The coding gain of this code over the uncoded 128-CR is 5.74 dB. This is also the largest possible coding gain that can be achieved with the 32-subset partitioning of the 16D rectangular constellation. This coding gain may be segregated into two parts, a gain of 6.02 dB from the trellis code if the 16D constellation were not expanded from 2^{56} to 2^{57} points, and a loss of 0.28 dB due to that expansion. This expansion loss is only 0.33 dB less than that for the 8D code of Section IV-B. The error coefficient of this code is larger than 1000 per 2D point. This number can be reduced to 412 if the number of states is increased to 128. The lower limit on the error coefficient using the 32-subset partitioning of the 16D constellation is 284 per 2D point, which is 1/8 the number (2272) of nearest neighbors to any point in the same 16D subset.

Trellis codes using an 8D constellation E_8 of 2^{29} points constructed from a 320-point 2D constellation and partitioned into 16 subsets as in the previous sections are described next. Although the partitioning of E_8 is such that there is no 8D family in between the constellation and the 8D subsets (see Fig. 5), we group the 16 8D subsets into two equal-sized 8D families in an arbitrary manner. The intrafamily MSED is therefore the same as the MSED of the constellation. Now note that the distance relationship between an 8D family and its eight subsets of the constellation E_8 is identical to that between an 8D family and its eight subsets of the 8D rectangular constellation partitioned in accordance with Table II. Since the construction of a trellis code with that partitioning of the 8D rectangular constellation does not depend on the interfamily MSED, all the trellis codes constructed for the 8D rectangular constellation can be used for the constellation E_8.

In particular, the rate-3/4, 64-state code of Section IV-B can be used with E_8 to achieve a MSED of $8d_0^2$, which is the MSED of each 8D subset. Since the average power of the constellation E_8 is $47.133d_0^2$, the coding gain of this code over the uncoded 128-CR is 5.41 dB. This is also the largest possible coding gain that can be achieved with the 16-subset partitioning of E_8. This coding gain may be segregated into three parts: a gain of 3.04 dB from the use of E_8 [3], another gain of 3.01 dB from the trellis code if the constellation E_8 were not expanded from 2^{28} to 2^{29} points, and a loss of 0.64 dB due to that expansion. The error coefficient of this code is the minimum, 240 per 8D point (or 60 per 2D point), which is the number of nearest neighbors to any point in the same 8D subset (an E_8 lattice). Note that both the coding gain and error coefficient of this code are the same as those of the 64-state code with the 8D rectangular constellation of Section IV-B.

The fact that the large MSED of E_8 does not help a trellis code achieve additional coding gain suggests the consideration of the 8D lattice DE_8. The MSED of the lattice DE_8 is only $2d_0^2$. An 8D constellation DE_8 of 2^{29} points can be constructed from a 256-point 2D constellation with an average power $40.6875d_0^2$, as shown in Section III. Both the size of the constituent 2D constellation and the average power of the constellation DE_8 are less than those of an 8D constellation E_8 of equal size. The constellation DE_8 is partitioned into two families of 16 subsets each as in Section II. The intrafamily and intrasubset MSED's are $4d_0^2$ and $8d_0^2$, respectively, which are the same as those of the partitioning of E_8 described above. A rate-4/5, 32-state code can be constructed with the constellation DE_8 to achieve an MSED of $8d_0^2$. The coding gain of this code over the uncoded 128-CR is therefore 6.05 dB. This is also the largest possible coding gain that can be achieved with the 32 subsets of the constellation DE_8. With some calculation, this coding gain may be segregated into three parts: a gain of 0.93 dB from the use of DE_8, another gain of 6.02 dB from the trellis code if the constellation DE_8 were not expanded from 2^{28} to 2^{29} points, and a loss of 0.9 dB due to that expansion. Comparing this segregation of coding gain with that for the constellation E_8, one can see that although the lattice DE_8 is a poor lattice for block-coded modulation, it is a good lattice for trellis-coded modulation. The error coefficient of this code is more than 500 per 2D point. This number can be reduced to 316 and 124 if the number of states is increased to 64 and 128, respectively. Further improvement is certainly possible by further increasing the number of states.

A recent report [16] gives trellis-coded modulation schemes based on a partitioning of the 4D lattice D_4, the most complex using a 64-state trellis code. Similar schemes may be constructed using the techniques of this paper, with a 4D constellation D_4 of 2^{15} points constructed from

a 256-point 2D constellation and partitioned into sixteen subsets as in the previous sections. A 64-state code can be shown to achieve a coding gain of 6.05 dB over the uncoded 128-CR. This is also the largest possible coding gain that can be achieved with the 16-subset partitioning of the constellation D_4. The coding gain may be segregated into three parts: a gain of 1.64 dB from the use of D_4 [3], another gain of 6.02 dB from the trellis code if the constellation D_4 were not expanded from 2^{14} to 2^{15} points, and a loss of 1.61 dB due to that expansion. The error coefficient of this code is, however, quite large, about ten times as large as that of the 64-state code with the 4D rectangular constellation of Section IV-C. It may be interesting to note here that the 4D rectangular lattice may be characterized as the union of D_4 and a rotated version of D_4 (see Fig. 4), which is similar to the relationship between the lattices DE_8 and E_8 and is another motivation for us to consider the lattice DE_8.

V. Decoder

A conventional maximum-likelihood decoding algorithm such as the Viterbi algorithm is used as the decoder [17]. As a preliminary step, the decoder must determine the point in each of the multidimensional subsets which is closest to the received point, and calculate its associated metric (the squared Euclidean distance between the two points). Because of the way in which a multidimensional constellation is partitioned, the closest point in each multidimensional subset and its associated metric may be obtained as follows. Each received $2N$-dimensional point is divided into a pair of N-dimensional points. The closest point in each $2N$-dimensional subset and its associated metric are found based on the point in each of the N-dimensional subsets which is closest to the corresponding received N-dimensional point and its associated metric. The N-dimensional subsets are those subsets which are used in Section II to construct the $2N$-dimensional subsets. The foregoing process may be used iteratively to obtain the closest point in each $2N$-dimensional subset and its associated metric based on the closest point in each of the basic 2D subsets and its associated metric.

As an example, we show how to determine the closest point in each of the 16 subsets of the 8D rectangular constellation of Section IV-B and its associated metric (see Fig. 12). First, for each of the four received 2D points of a received 8D point, the decoder determines the closest 2D point in each of the four 2D subsets of the 160-point 2D constellation of Fig. 2, and calculates its associated metric. These metrics are called 2D subset metrics. Because there are only 40 2D points in each of the four 2D subsets, this step is quite easy, being no more complex than that required for a 2D code.

Next, the decoder determines the 4D point in each of the 16 4D types (see Table I) which is closest to the first received 4D point (the 4D point corresponding to the first and second 2D points of the received 8D point), and calculates its associated metric. These metrics are called 4D type metrics. The 4D type metric for a 4D type is

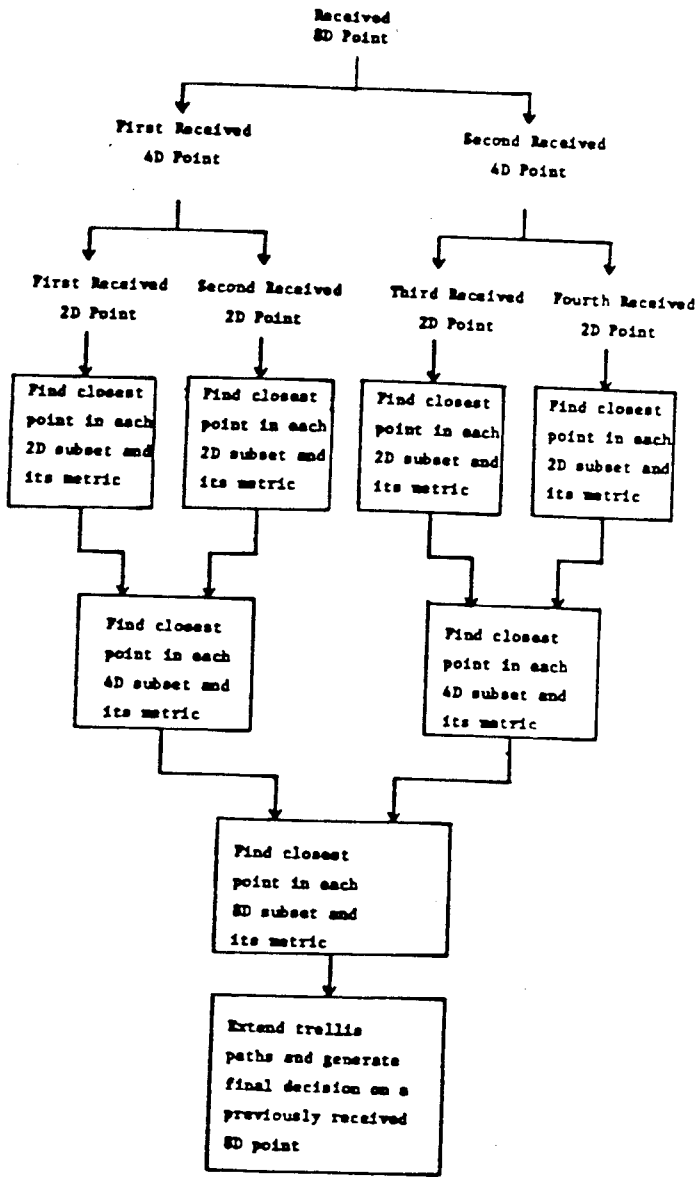


Fig. 12. Viterbi decoding algorithm for 64-state code of Fig. 8.

obtained merely by adding the two 2D subset metrics for the pair of 2D subsets corresponding to that 4D type. The decoder then compares the two 4D type metrics for the pair of 4D types within each 4D subset (see Table I). The smaller 4D type metric becomes the 4D subset metric associated with that 4D subset, and the 4D point associated with the smaller 4D type metric is the 4D point in that 4D subset which is closest to the first received 4D point. The same process is repeated for the second received 4D point.

The decoder then determines the closest 8D point in each of the 64 8D types (see Table II) and calculates its associated metric. These metrics are called 8D type metrics. The 8D type metric for an 8D type is obtained by adding the two 4D subset metrics for the pair of 4D subsets corresponding to that 8D type. Finally, the decoder compares the four 8D type metrics corresponding to the four 8D types within each 8D subset (see Table II). The smallest 8D type metric becomes the 8D subset metric

associated with that 8D subset, and the 8D point associated with the smallest 8D type metric is the closest 8D point in that 8D subset. These 8D subset metrics are then used to extend the trellis paths and generate final decisions on the transmitted 8D points in the usual way.

The final decision on a transmitted 8D point obtained from the foregoing procedure may not be a valid point of the 8D constellation of Section IV-B because more than one of the four 2D points of the decision may come from the outer group of the 2D constellation of Fig. 2. When this happens, a modification must be made in the procedure to arrive at a valid 8D point. This type of error is caused by the boundary effect of the finite constellation. Since we did not consider that effect in our calculation of the error coefficient in the last section, this type of error is already included in the error coefficient.

Because of the construction of multidimensional constellations described in Section III, at the receiver the mapping from each decoded multidimensional point back to the information bits is simplified. Again, we will use the code of Section IV-B to illustrate this inverse mapping procedure. Each of the four 2D points corresponding to the final decision on a transmitted 8D point is first mapped back to eight Z bits (see Fig. 2 and Table IV). Then performing the inverse conversions corresponding to the bit converter and 8D block encoder (see Figs. 9 and 10) followed by a differential decoding operation produces the desired 28 information bits. The mapping from a 2D point back to the eight Z bits requires a table of only 160×8 bits. The inverse conversion corresponding to the bit converter requires a table of only 256×7 bits. The inverse conversion corresponding to the 8D block encoder may be done with a short procedure which does not even require a table.

VI. Comparisons and Conclusion

To compare trellis codes using various partitionings of multidimensional or 2D constellations, a way to measure code complexity is needed. With the simplified multidimensional constellation mapping as described in Section IV and the simplified decoding method as described in Section V, a good measure of code complexity is the ratio of the total number of allowed state transitions of a code to the number of signaling intervals corresponding to the dimensionality of its constellation. For example, for the eight-state 2D code used in the CCITT standards V.32 and V.33, this ratio is 32 [2], [4], [5]. For the 16-state 4D code of Section IV-A, this ratio is also 32.

The reason for using this ratio to measure code complexity is that to update the path metric of each state of a code in the Viterbi decoding algorithm, the path metrics associated with all the transitions leading to that state need to be calculated and compared. This must be done only once every block of N signaling intervals, where $2N$ is the number of dimensions in the constellation. The updating of the path metrics dominates the code complexity. (The last statement is valid at least for the codes to be compared in Table VI. A refined measure of code complexity, which takes into account the calculation used to obtain multidimensional subset metrics, may be considered for codes with a small number of states relative to their number of dimensions.)

Table VI lists the characteristics of some of the multidimensional trellis codes that have been studied, along with the characteristics of some 2D trellis codes from [1], [2]. (Some of the 2D code characteristics have not been published previously.) The error coefficients for the 64- and 128-state 2D codes are due to [18]. The codes in Table VI are listed by increasing complexity and, for codes of equal complexity, by increasing number of dimensions. Although some characteristics shown in Table VI are for the case where the number of information bits transmitted per 2D signaling interval is equal to seven, the conclusions drawn below may be generally applied to other transmission rates.

The principal conclusion is that for the same (modest) complexity (i.e., complexity less than or equal to that of the 32-state 2D code), trellis-coded modulation with multi-

TABLE VI
Code Comparison

Scheme	Lattice	Number of Points in Constituent 2D Constellation[a]	Number of States of Trellis Code	Number of Multi-Dimensional Subsets	Relative Code Complexity	Coding Gain[a,b] (dB)	Error Coefficient (per 2D Point)	Peak-to-Average Power Ratio[a]
1	2D Rectangular	256	8	8	1	4.01	16	1.93
2	4D Rectangular	192	16	8	1	4.66	12	2.16
3	2D Rectangular	256	32	8	4	4.80	16	1.93
4	8D Rectangular	160	64	16	4	5.41	60	2.14
5	E_8	320	64	16	4	5.41	60	2.17
6	2D Rectangular	256	64	8	8	5.47	56	1.93
7	D_4	256	64	16	8	6.05	828	1.93
8	DE_8	256	64	32	8	6.05	316	1.93
9	16D Rectangular	144	128	32	8	5.74	412	2.03
10	2D Rectangular	256	128	8	16	6.05	344	1.93
11	4D Rectangular	192	64	32	16	5.63	72	2.16
12	DE_8	256	128	32	16	6.05	124	1.93

[a] For transmitting seven information bits per 2D signaling interval.
[b] As compared to the uncoded 128-CR.

dimensional rectangular constellations is superior to using 2D constellations. Using a multidimensional rectangular constellation not only improves the performance in terms of both the coding gain and error coefficient, but also reduces the size of the constituent 2D constellations. A smaller 2D constellation implies better performance when the received signal contains signal-dependent noise. Examples of this conclusion are Scheme 1 versus Scheme 2 and Scheme 3 versus Scheme 4 of Table VI. The better performance of Schemes 2 and 4 is not obtained at the cost of increasing the peak-to-average power ratio of the constellations to an unacceptably large value. This is also true for all other multidimensional trellis codes appearing in this paper. It is important to keep this ratio small because the transmission medium may distort the transmitted signal nonlinearly.

Trellis codes with the densest 8D constellation E_8 do not seem to have any advantage over codes using the 8D rectangular constellation. There is, however, a disadvantage associated with the constellation E_8, namely that the constituent 2D constellation of E_8 is twice as large as that for the 8D rectangular constellation, as may be seen by comparing Schemes 4 and 5.

Using a rectangular constellation with more than eight dimensions seems to yield diminishing returns, as may be seen by comparing Scheme 4 and 9. The size of the constituent 2D constellations of Scheme 9 is 9/10 times that of Scheme 4, which is not much smaller. The coding gain of Scheme 9 is only 0.33 dB more than that of Scheme 4, but the error coefficient of Scheme 9 is about seven times that of Scheme 4. Even worse, the complexity of Scheme 9 is twice as much as that of Scheme 4.

For higher complexity, using a multidimensional rectangular constellation has less to offer. Schemes 10 and 11 have about the same performance, comparing both the coding gain and error coefficient. The advantage of a smaller constituent 2D constellation for the multidimensional rectangular constellation is, however, preserved.

The advantage of a smaller constituent 2D constellation for the multidimensional constellation disappears when the densest 4D constellation D_4 or the 8D constellation DE_8 is used. Trellis codes with the constellation DE_8 are only slightly better than their 2D correspondents, as may be seen by comparing Schemes 6 and 8, or Schemes 10 and 12. DE_8 shows, however, an important concept. That is, while it is desirable that the multidimensional subsets used in trellis-coded modulation be dense lattices, it is not necessarily desirable that the overall multidimensional constellation be as dense. Multidimensional rectangular constellations are the extreme case of the above statement. We have not found that trellis codes using the densest 4D constellation D_4 have any advantage.

From another viewpoint, using a multidimensional constellation instead of a 2D constellation with a trellis code reduces the code complexity while maintaining the same performance in terms of both the coding gain and error coefficient. Examples of this conclusion are Scheme 2 versus Scheme 3, Scheme 4 versus Scheme 6, and Scheme 8 versus Scheme 10.

Generally speaking, it is easier to construct a trellis code that is transparent to all phase ambiguities of the constellation using a multidimensional instead of a 2D constellation. The reason is that some or all of the phase ambiguities of a multidimensional constellation may be removed by a careful partitioning of the multidimensional constellation into subsets as described in Section II, without the involvement of the trellis code. Examples are Schemes 2, 4, 5, 8, 9, and 12. Scheme 11 is also transparent to all phase ambiguities of its constellation as demonstrated in Section IV-C. All phase ambiguities of Scheme 11 are, however, removed with the involvement of the trellis code.

These multidimensional trellis codes may be generalized to rectangular constellations with dimensions other than 4, 8, or 16, to other multidimensional constellations with rectangular or nonrectangular constituent 2D constellations, and to trellis codes of rate other than $m/m+1$.

Acknowledgment

The author is indebted to Dr. G. D. Forney, Jr., for his encouragement and guidance in preparing this paper. The author also wishes to thank Dr. S. U. Qureshi and Prof. R. G. Gallager for their helpful discussions. Dr. V. Eyuboglu, Mr. M. E. Huntzinger, and Mr. M. Gilbride provided useful help in doing real-time experiments with some codes.

References

[1] G. Ungerboeck, "Channel coding with multilevel/phase signals," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 55–67, Jan. 1982.

[2] L. F. Wei, "Rotationally invariant convolutional channel coding with expanded signal space—Part I: 180 degrees and Part II: Nonlinear codes," *IEEE J. Select. Areas Commun.*, vol. SAC-2, pp. 659–686, Sept. 1984.

[3] G. D. Forney, Jr., *et al.*, "Efficient modulation for band-limited channels," *IEEE J. Select. Areas Commun.*, vol. SAC-2, pp. 632–647, Sept. 1984.

[4] CCITT Study Group XVII, "Recommendation V.32 for a family of 2-wire, duplex modems operating at data signalling rates of up to 9600 bit/s for use on the general switched telephone network and on leased telephone-type circuits," Document AP VIII-43-E, May 1984.

[5] CCITT Study Group XVII, "Draft recommendation V.33 for 14400 bits per second modem standardized for use on point-to-point 4-wire leased telephone-type circuits," Circular 12, COM XVII/YS, Geneva, Switzerland, May 17, 1985.

[6] J. D. Brownlie and E. L. Cusack, "Duplex transmission at 4800 and 9600 bit/s on the general switched telephone network and the use of channel coding with a partitioned signal constellation," in *Proc. Zurich Int. Sem. Digital Commun.*, Mar. 1984.

[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, Mar 1982.

[8] ——, "A fast encoding method for lattice codes and quantizers," *IEEE Trans. Inform. Theory*, vol. IT-29, pp. 820–824, Nov. 1983.

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

[10] N. J. A. Sloane, "Tables of sphere packings and spherical codes," *IEEE Trans. Inform. Theory*, vol. IT-27, pp. 327–338, May 1981.

[11] A. Gersho and V. B. Lawrence, "Multidimensional signal constellations for voiceband data transmission," *IEEE J. Select. Areas Commun.*, vol. SAC-2, pp. 687–702, Sept. 1984.

[12] E. L. Cusack, "Error control codes for QAM signaling," *Electron. Lett.*, vol 20, pp. 62–63, 1984.

[13] R. Fang and W. Lee, "Four-dimensionally coded PSK systems for combatting effects of severe ISI and CCI," in *Proc. IEEE Globecom Conv. Rec.*, pp. 30.4.1–30.4.7, 1983.

[14] S. G. Wilson *et al.*, "Four-dimensional modulation and coding: An alternate to frequency-reuse," in *Proc. IEEE ICC Conv. Rec.*, pp. 919–923, 1984.

[15] A. R. Calderbank and N. J. A. Sloane, "Four-dimensional modulation with an eight-state trellis code," *AT&T Tech. J.*, vol. 64, pp. 1005–1018, May–June 1985.

[16] ——, "New trellis codes based on lattices and cosets," *IEEE Trans. Inform. Theory*, vol. IT-33, pp. 177–195, Mar. 1987.

[17] G. D. Forney, Jr., "The Viterbi algorithm," *Proc. IEEE*, vol. 61, pp. 268–278, Mar. 1973.

[18] V. Eyuboglu, private communication.

AN EIGHT DIMENSIONAL 64-STATE TRELLIS CODE FOR TRANSMITTING 4 BITS PER 2D SYMBOL

S.A. Tretter
Electrical Engineering Department
University of Maryland
College Park, MD 20742

ABSTRACT

An eight-dimensional, 64-state, 90 degree rotationally invariant trellis code for transmitting 4 bits per baud over a bandlimited channel is described. The 2D constellation contains 20 points. The code achieves a 5.23 dB coding gain over the uncoded 4×4 QAM constellation and a 1.23 dB gain over the standard CCITT V32 trellis code. Simulation results are presented that verify these coding gains. Simulation results showing symbol error probability vs. SNR and trellis depth are also presented.

I. INTRODUCTION

This paper describes an eight-dimensional, 64-state, 90 degree rotationally invariant trellis code for transmitting four bits per baud over a bandlimited channel with QAM modulation. It is based on the work of Wei [1]. The code achieves a 5.23 dB coding gain over the uncoded 4×4 QAM constellation and a 1.23 dB gain over the standard two-dimensional CCITT V32 trellis code when soft decision Viterbi decoding is used. The 8D encoder operates on blocks of 16 consecutive data bits taken four per baud over four bauds. A systematic convolutional encoder adds one check bit to the 16 data bits and the resulting 17 bits are mapped into an 8D point consisting of a sequence of four 2D signal points each selected from a 20-point 2D constellation. The V32 code selects signal points from the 32-point cross 2D constellation. The fact that a 20-point 2D constellation is used each baud for the 8D code rather than the 32-point constellation makes this 8D code more immune to channel impairments such as phase jitter and nonlinearities. Furthermore, when the received baseband signal point is used for equalizer updating and carrier tracking, the 20 point constellation gives more reliable decision directed performance.

The 8D signal constellation and its theoretical coding gain when combined with the 64-state trellis encoder are presented in Section II. The 8D encoder is described in Section III and the decoder in Section IV. The results of simulations to measure symbol error probability vs. SNR and trellis depth are discussed in Section V.

II. CONSTELLATION FOR THE 8D CODE AND ITS THEORETICAL PERFORMANCE

The 2^{17} 8D constellation points are selected from an 8D rectangular lattice. Each 8D point consists of a sequence of four 2D points which are selected from a 2D rectangular lattice and each 2D point is transmitted by QAM modulation. The 2D constellation is formed by first selecting 16 points from a 2D rectangular lattice corresponding to a standard uncoded constellation with 90 degree rotational symmetry for transmitting 4 bits per symbol. These points are called the inner points. In this case, the 4×4 rectangular grid of points shown inside the dotted lines in Fig. 1 was selected. Next, four outer points selected from the rectangular lattice to preserve symmetry and have minimum energy are added to the inner points resulting in the 20-point constellation shown in Fig. 1. These 20 points are partitioned into the four subsets designated A, B, C, and D in Fig. 1. Each subset contains five points and the minimum Euclidean distance between points within a subset is $d_f = 4$. Under a 90 degree clockwise rotation, subset A becomes C, C becomes B, B becomes D, and D becomes A. Also notice that the three bits assigned to each point are invariant to 90 degree rotations. This is part of the reason the code is transparent to rotations.

Next, pairs of 2D subsets are concatenated to form the eight 4D subsets listed in Table 1. These subsets were formed so that the minimum Euclidean distance between 4D points within a subset is 4 just as in the 2D case.

Finally, pairs of 4D subsets are concatenated to form the 16 8D subsets shown in Table 2. The integers under the column labelled "8D Types" refer to the 4D subsets in Table 1. Each 8D type contains four sequences of 2D subsets so each 8D subset contains 16 distinct sequences of four 2D subsets, each of which is called an 8D subtype. The minimum Euclidean distances between 8D subtypes within an 8D subset is again 4. Also, the 8D subsets are invariant of 90 degree rotations.

8D constellation points are selected by the encoder so that either a sequence of four 2D inner points or a sequence with one outer point and three inner points is selected. The outer point can be in any of the four positions. This results in 2^{17} 8D constellation points. Also, the probability of selecting an inner point is $P_{in} = 7/8$ and the probability of an outer point is $P_{out} = 1/8$. Thus, the average signal power transmitted per baud is

$$S = P_{in}S_{in} + P_{out}S_{out} = 12$$

Wei [1] shows that the minimum free Euclidean distance for the encoder described in Section III is equal to the minimum distance between points within one of the 2D subsets which is 4. At high signal-to-noise ratios (SNR), the coding gain is

$$G = 10\log_{10}\frac{(d_f^2/S)_{\text{coded}}}{(d^2/S)_{\text{uncoded}}} = 5.2287dB$$

III. THE 8D ENCODER

A block diagram of the 8D encoder is shown in Fig. 2. It accepts blocks of 16 successive data bits taken four bits per baud. These are labelled I1n, I2n,$\cdots$,I4n+3. The subscripts $n, n+1, n+2$, and $n+3$ indicate the four successive bauds. The first three bits (I1n, I2n, I3n) drive a systematic, rate 3/4, 64 state, feedback convolutional encoder. The encoder generates the one redundant bit labelled YOn. The bits I3n+1 and I2n+1 are differentially

encoded to make the code transparent to 90 degree rotations. If the pairs are considered to be two bit numbers, the differential encoding rule is

$$(I3n+1', I2n+1') = (I3n+1, I2n+1) +$$
$$(I3n-3', I2n-3') \bmod 4$$

The eight bits Y0n,···,I3n+1' are mapped into the eight bits Z0n, Z1n,···,Z1n+3 by the BIT CONVERTER which can be implemented most easily as a 256 byte lookup table. The four bits Y0n, I1n, I2n, and I3n are used to select the 8D subset according to Table 2. The remaining four bits select the subtype within the subset. Within each subset, there are four groups of four subtypes that are rotationally invariant to 90 degree rotations. I3n and I3n+1 select one of these four groups. I2n+1' and I3n+1' select the subtype within the selected group. The differential encoding described above makes the code transparent to 90 degree rotations. A specific algorithm for selecting the sequence of four 2D subsets for an 8D subtype is shown in Table 3. Each pair (Z1, Z0) specifies a 2D subset according to the rule

Z1	Z0	2D Subset
0	0	A
0	1	C
1	0	B
1	1	D

The remaining nine input bits determine the specific 8D point within the 8D subtype. The 8D BLOCK ENCODER shown in Fig. 2 maps these 9 bits into the 12 Z bits (Z2n+i, Z3n+i, Z4n+i) for i=0,1,2, and 3 using the rules displayed Table 4. These triplets of Z bits specify the sequence of four 2D constellation points corresponding to the sequence of 2D subsets of the 8D subtype selected. The mapping of a triplet to a 2D point is shown in Fig. 1. Notice that Z2n+i determines whether the 2D point is an inner or outer point.

I1n, I2n, and I3n direct the convolutional encoder through a sequence of states. Each state consists of the six W bits shown in Fig. 2. Each state can make a transition to eight distinct next states and each next state can be reached from eight distinct previous states. A table showing the state transitions with their 8D subset assignments can be found in [1].

IV. THE 8D DECODER

The decoder is presented with received 8D symbols consisting of blocks of four successive 2D symbols. Each 2D symbol consists of an x and y component represented to the accuracy of the receiver's A/D converter. The decoder implements the soft decision Viterbi algorithm.

The encoder assigns an 8D subset to each branch in the trellis, so there are parallel paths. Thus, the first step in the decoding process is to quantize the received 8D symbol to the nearest point in each of the 16 8D subsets. This can be efficiently accomplished by going from 2D to 4D to 8D decisions. The decoder first quantizes each received 2D symbol to the closest constellation point in each of the 2D subsets A, B, C, and D. The corresponding squared Euclidean distances and Z bits are saved. The next step is to combine pairs of 2D decisions into 4D decisions. Using Table 1, the best 4D point for each of the eight 4D subsets is found for the first and second pairs of received 2D points and the corresponding Z bits and squared distances are saved. Finally, these pairs of 4D points are combined to find the best 8D point in each of the 8D subsets listed in Table 2 and the corresponding Z bits and squared distances are saved.

Next the path metrics to each of the 64 states are updated using the standard soft decision Viterbi decoding algorithm. The state with the smallest metric is found and the trellis memory is traced back to its beginning to find the decoded 8D symbols. The decoded Z bits are transformed back into information bits by inverses of the bit converter and block encoder.

V. SIMULATION RESULTS

A FORTRAN simulation was used to evaluate the performance of this 8D code in the presence of additive Gaussian noise. The 2D CCITT V32 code and 4×4 QAM uncoded constellation were also simulated. Complete programs and a more detailed report are available from the author.

Simulation results for the 8D code with a trellis depth of 32 8D symbols, the V32 code with a trellis depth of 32 2D symbols, and the 4×4 QAM constellation are shown in Fig. 3. The expected coding gains were achieved at high SNR. As expected, the gains diminished as SNR decreased.

Simulations were also performed to measure the performance as a function of the trellis depth. The required trellis depth is an important parameter to know for a practical decoder implementation. The results are shown in Fig. 4. Essentially all the expected coding gain was achieved with a trellis depth of 15 8D symbols.

REFERENCES

1. Lee-Fang Wei, "Trellis-Coded Modulation with Multidimensional Constellations," IEEE Trans. on Information Theory, Vol. IT-33, No. 4, July 1987, pp. 483-501.

ACKNOWLEDGEMENTS

This work was supported by Penril DataComm, 207 Perry Parkway, Gaithersburg, MD 20877.

4D Subset	4D Types
0	(A,A), (B,B)
1	(C,C), (D,D)
2	(A,B), (B,A)
3	(C,D), (D,C)
4	(A,C), (B,D)
5	(C,B), (D,A)
6	(A,D), (B,C)
7	(C,A), (D,B)

Table 1
Partitioning of 4D Constellation
into 8 Subsets

8D Subset	$Y0_n$	$I1_n$	$I2_n$	$I3_n$	8D Types
0	0	0	0	0	(0,0),(1,1),(2,2),(3,3)
1	0	0	0	1	(0,1),(1,0),(2,3),(3,2)
2	0	0	1	0	(0,2),(1,3),(2,0),(3,1)
3	0	0	1	1	(0,3),(1,2),(2,1),(3,0)
4	0	1	0	0	(4,4),(5,5),(6,6),(7,7)
5	0	1	0	1	(4,5),(5,4),(6,7),(7,6)
6	0	1	1	0	(4,6),(5,7),(6,4),(7,5)
7	0	1	1	1	(4,7),(5,6),(6,5),(7,4)
8	1	0	0	0	(0,4),(1,5),(2,6),(3,7)
9	1	0	0	1	(0,5),(1,4),(2,7),(3,6)
10	1	0	1	0	(0,6),(1,7),(2,4),(3,5)
11	1	0	1	1	(0,7),(1,6),(2,5),(3,4)
12	1	1	0	0	(4,0),(5,1),(6,2),(7,3)
13	1	1	0	1	(4,1),(5,0),(6,3),(7,2)
14	1	1	1	0	(4,2),(5,3),(6,0),(7,1)
15	1	1	1	1	(4,3),(5,2),(6,1),(7,0)

Table 2
Partitioning of 8D Constellation into
16 Subsets

Note: The integers under the column labelled 8D Types refer to the 4D subsets defined in Table 1.

STEP 1

Use Y0n, I1n, I2n, and I3n to select an 8D subtype from this table

$Y0_n$	$I1_n$	$I2_n$	$I3_n$	8D Subtypes
0	0	0	0	(AAAA)
0	0	0	1	(AACC)
0	0	1	0	(AAAB)
0	0	1	1	(AACD)
0	1	0	0	(ACAC)
0	1	0	1	(ACCB)
0	1	1	0	(ACAD)
0	1	1	1	(ACCA)
1	0	0	0	(AAAC)
1	0	0	1	(AACB)
1	0	1	0	(AAAD)
1	0	1	1	(AACA)
1	1	0	0	(ACAA)
1	1	0	1	(ACCC)
1	1	1	0	(ACAB)
1	1	1	1	(ACCD)

STEP 2

Rotate the 3rd & 4th / 2nd & 4th 2D subsets of the 8D subtype by 180 degrees if (I4n, I1n+1) = 2nd & 3rd

0	1
1	0
1	1

STEP 3

Rotate all four 2D subsets of the 8D subtype obtained in STEP 2 by

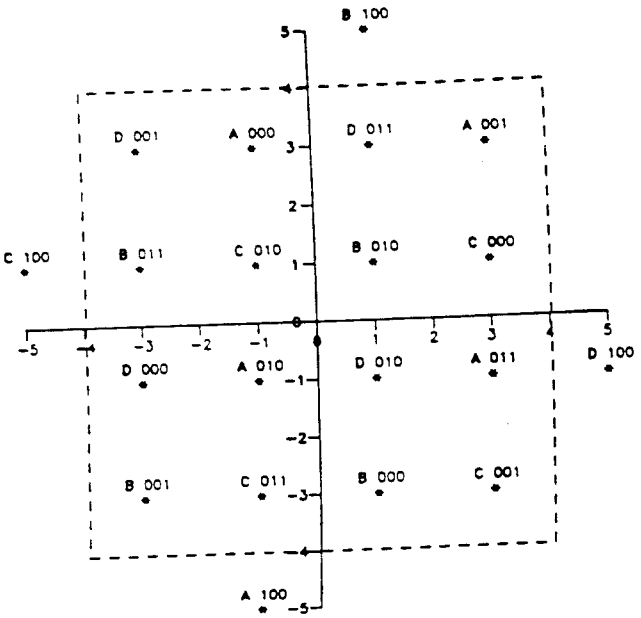
90 / 180 degrees clockwise if (I2n+1′, I3n+1′) = / 270

0	1
1	0
1	1

Table 3
8D Subtype Selection Procedure

	I4n+1	I1n+2	I2n+2	Z2n	Z3n	Z4n	Z2n+1	Z3n+1	Z4n+1	Z2n+2	Z3n+2	Z4n+2	Z2n+3	Z3n+3	Z4n+3
4 inner pts →	0	x	x	0	I1n+2	I2n+2	0	I3n+2	I4n+2	0	I1n+3	I2n+3	0	I3n+3	I4n+3
	1	0	0	1	0	0	0	I3n+2	I4n+2	0	I1n+3	I2n+3	0	I3n+3	I4n+3
1 cute-	1	0	1	0	I3n+2	I4n+2	1	0	0	0	I1n+3	I2n+3	0	I3n+3	I4n+3
3 inner pts	1	1	0	0	I3n+2	I4n+2	0	I1n+3	I2n+3	1	0	0	0	I3n+3	I4n+3
	1	1	1	0	I3n+2	I4n+2	0	I1n+3	I2n+3	0	I3n+3	I4n+3	1	0	0

Table 4
8D BLOCK ENCODER



Note Points are labelled by subset and (Z2,Z3,Z4)

Fig. 1. 20-Point 2D Signal Constellation for 8D Code

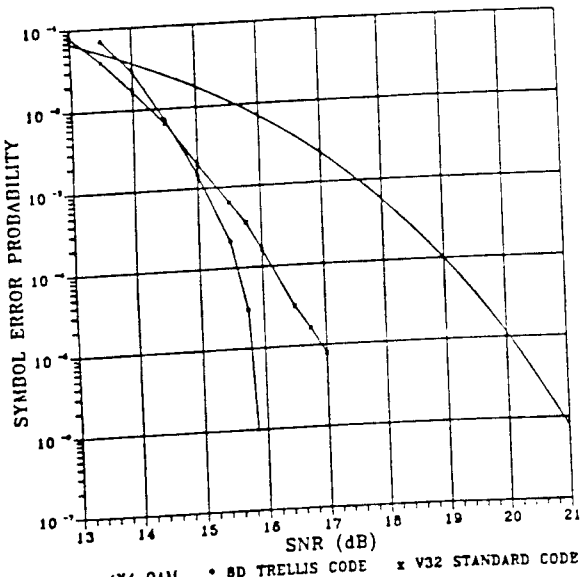


Fig. 3. Simulation Results for 8D Code, V32 Code, and 4x4 QAM

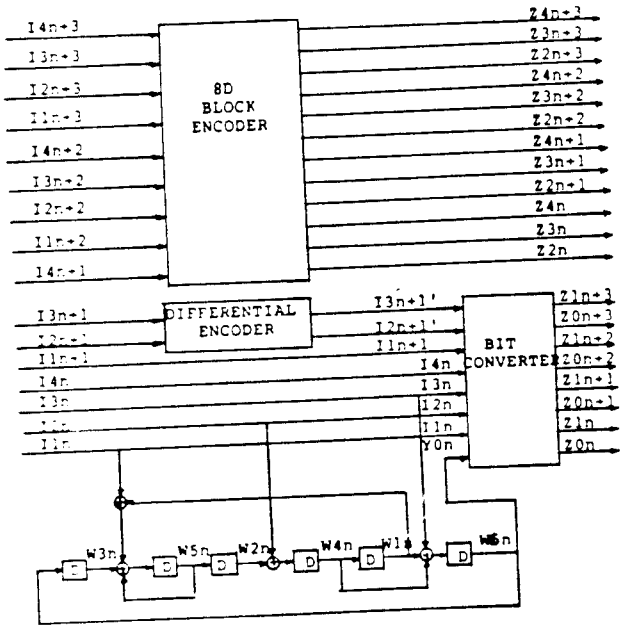


Fig. 2. 64-State 8D Encoder.

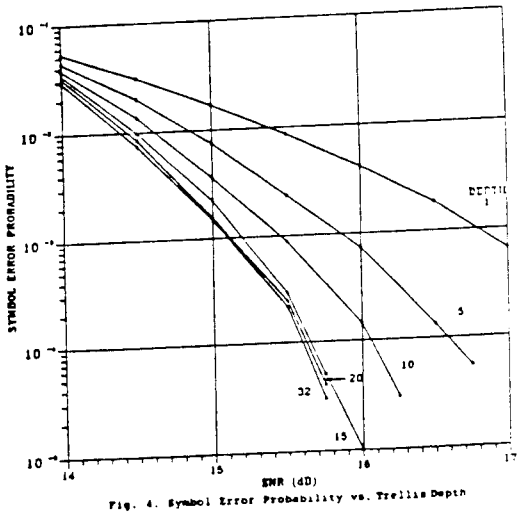


Fig. 4. Symbol Error Probability vs. Trellis Depth

Recent Key Papers on Multi-Dimensional and Trellis Codes

1. G.D. Forney, Jr., "Coset Codes I: Introduction and Geometrical Classification," IEEE Trans. on Information Theory, Vol IT-34, No. 5, Sept. 1988, pp. 1123-1151.
2. G.D. Forney, Jr., "Coset Codes II: Binary Lattices and Related Codes," IEEE Trans. on Information Theory, Vol 34, No. 5, Sept. 1988, pp. 1152-1187.
3. G.D. Forney, Jr. and L.F. Wei, "Multidimensional Constellations I: Introduction, Figures of Merit, and Generalized Cross Constellations," IEEE Journal on Selected Areas of Communications, Vol SAC-7, August 1989.
4. G.D. Forney, Jr., "Multidimensional Constellations II: Voronoi Constellations," IEEE Journal on Selected Areas of Communications, Vol SAC-7, August 1989.
5. G.D. Forney, Jr., "Trellis Shaping," Submitted to IEEE Trans. on Information Theory, July 1990. Presented in part at IEEE Info. Theory Workshop, Ithaca, N.Y., June 1989.
6. M.V. Eyuboglu and G.D. Forney, Jr., "Trellis Precoding," Submitted to IEEE Trans. Info. Theory 1990. Presented at IEEE Symposium on Info. Theory, San Diego, Jan. 1990
7. G.D. Forney, Jr. "Coded Modulation for Band-Limited Channels," IEEE Info. Theory Society Newsletter, Dec. 1990