Efficient Modulation for Band-Limited Channels


Abstract—This paper attempts to present a comprehensive tutorial survey of the development of efficient modulation techniques for band-limited channels, such as telephone channels. After a history of advances in commercial high-speed modems and a discussion of theoretical limits, it reviews efforts to optimize two-dimensional signal constellations and presents further elaborations of uncoded modulation. Its principal emphasis, however, is on coded modulation techniques, in which there is an explosion of current interest, both for research and for practical application. Both block-coded and trellis-coded modulation are covered, in a common framework. A few new techniques are presented.

I. HISTORICAL INTRODUCTION

Band-limited channels (as opposed to power-limited) are those on which the signal-to-noise ratio is high enough so that the channel can support a number of bits/Hz of bandwidth. The telephone channel (particularly the dedicated private line) has historically been the scene of the earliest application of the most efficient modulation techniques for band-limited channels. The reasons have to do with the commercial importance of such channels and with the fact that they can be modeled to first order as linear time-invariant channels, sharply band-limited between typically 300–3000 Hz, with high signal-to-noise ratios, typically 28 dB or greater. Their relatively low bandwidth permits a great deal of signal processing per transmission element, and therefore early application of the most advanced techniques, which have often then been applied several years later to broader-band channels (e.g., radio).

The earliest commercially important telephone-line modems appeared in the 1950's and used frequency shift keying to achieve speeds of 300 bits/s (Bell 103), or 1200 bits/s on private lines (Bell 202). The earliest commercially important synchronous modem was the Bell 201, introduced in about 1962, which used 4-phase modulation in a nominal 1200 Hz bandwidth to achieve 2400 bits/s on private lines. This remained the state of the art for most of the decade. (It was not unknown in this period to encounter users who thought that Nyquist or Shannon or someone else had proved that 2400 bits/s was about the maximum rate theoretically possible on phone lines.)

The first commercially important 4800 bit/s modem was the Milgo 4400/48, introduced in about 1967, which included a manually adjustable equalizer to allow use of a nominal 1600 Hz bandwidth in conjunction with 8-phase modulation to send 3 bits/Hz. The development of digital adaptive equalization soon allowed expansion of the nominal bandwidth to 2400 Hz, essentially the full telephone line bandwidth. Following a first generation of single-sideband 9600 bit/s modems in the late 1960's, which were only marginally successful, broad success was achieved by the Codex 9600C, introduced in 1971, which used quadrature amplitude modulation (QAM) with a 16-point signal constellation to send 4 bits/Hz in a nominal 2400 Hz bandwidth. (16-point QAM had been used at 4800 bits/s by ESE and ADS about 1970.) Modems with these characteristics remained the state of the art for another decade, and it was thought by many (including some of the present authors, who should have known better) that higher-speed modems would never be widely used commercially.

In 1980, a first generation of 14 400 bit/s modems was introduced by Paradyne (MP 14400), followed in 1981 by the Codex/ESE SP14.4 and by others. These modems simply extended 2400 Hz QAM modulation to 6 bits/Hz by using 64-point signal constellations, and by using advanced implementation techniques and exploiting the gradual upgrading of the telephone network, proved to operate successfully over a high percentage of circuits. In a second generation of 14.4 kbit/s modems that will begin appearing in 1984, coded QAM modulation is being introduced to provide greater performance margins. The principal focus of this paper will be on these new coded modulation techniques.

This evolution of high-speed modems to ever higher bit rates using successively more complicated modulation schemes is summarized in Table I, along with the designation and year of final adoption of CCITT international standards embodying these schemes. How far can this evolution go? History would suggest caution in stipulating any ultimate ceiling. However, without any dramatic general upgrading of the telephone network, we venture to say that 19.2 kbit/s is the maximum conceivable rate for a
TABLE I
Modem Milestones

<table>
<thead>
<tr>
<th>YEAR</th>
<th>MODEL</th>
<th>SPEED</th>
<th>BANDWIDTH</th>
<th>SYMBOL</th>
<th>COMMENTS</th>
<th>CCITT STANDARD</th>
</tr>
</thead>
<tbody>
<tr>
<td>1955</td>
<td>Bell 201</td>
<td>2400</td>
<td>1200 Hz</td>
<td>2</td>
<td>4-phase</td>
<td>V.25 (1961)</td>
</tr>
<tr>
<td>1957</td>
<td>Megs 4400</td>
<td>4800</td>
<td>1800 Hz</td>
<td>3</td>
<td>8-phase</td>
<td>V.27 (1972)</td>
</tr>
<tr>
<td>1971</td>
<td>Codex 9000C</td>
<td>9000</td>
<td>2400 Hz</td>
<td>4</td>
<td>16-QAM</td>
<td>V.29 (1976)</td>
</tr>
<tr>
<td>1980</td>
<td>Paradyne MP4400</td>
<td>14,400</td>
<td>2400 Hz</td>
<td>6</td>
<td>64-QAM</td>
<td></td>
</tr>
<tr>
<td>1981</td>
<td>Codex/ESS SP14.4</td>
<td>14,400</td>
<td>2400 Hz</td>
<td>6</td>
<td>64-QAM</td>
<td></td>
</tr>
<tr>
<td>1984</td>
<td>14,400</td>
<td>2400 Hz</td>
<td>6</td>
<td>128-QAM</td>
<td>8-state</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 1. Canonical QAM modulator.

Fig. 2. QAM channel model.

telephone-line modem for general use, even with all-out use of the most powerful coded modulation. We shall see.

II. CHANNEL MODEL AND BASIC LIMITS

A quadrature amplitude modulator can be used to generate any standard linear double-sideband modulated carrier signal (including forms of phase modulation and phase/amplitude modulation), which includes all types of modulation in general use in synchronous modems. A canonical QAM modulator is shown in Fig. 1.

Assuming that the only channel impairment is Gaussian noise and that the receiver achieves perfect carrier phase tracking, the simple model of Fig. 2 applies. Signals are sent in pairs \((x_1, y_1)\); the channel is essentially two-dimensional. We shall call such a pair a "signal point," imagined to lie on a two-dimensional plane. Signal points are sent at a regular rate of \(F \cdot \text{points/s} \) where \(F \) Hz is the nominal (Nyquist) bandwidth of the channel. A signal point is also called a "symbol," and the symbol interval is \(1/F_s\). The model indicates that the two signal point coordinates \((x_1, y_1)\) are independently transmitted over decoupled parallel channels and perturbed by Gaussian noise variables \((n_{x_1}, n_{y_1})\), each with zero mean and variance \(N\). Alternatively, we could regard the two-dimensional signal point as being perturbed by a two-dimensional Gaussian noise variable. If the average energy (the mean squared value) of each signal coordinate is \(S\), then the signal-to-noise ratio is \(S/N\).

The simplest method of digital signaling through such a system is to use one-dimensional pulse amplitude modulation (PAM) independently for each signal coordinate. (This is sometimes called narrow-sense QAM.) In PAM, to send \(m\) bits/dimension, each signal point coordinate takes on
one of $2^m$ equally likely equispaced levels, conventionally $\pm 1, \pm 3, \pm 5, \cdots, \pm (2^m-1)$. The average energy of each coordinate is then

$$S_m = (4^m-1)/3,$$

and it follows that

$$S_{m+1} = 4S_m + 1.$$

That is, it takes approximately (asymptotically, exactly) 4 times as much energy (or 6 dB) to send an additional 1 bit/dimension or 2 bits/symbol. The probability $P(E)$ that either $x_i$ or $y_i$ is in error is upperbounded and closely approximated by the probability that the two-dimensional Gaussian noise vector $(n_x, n_y)$ lies outside a circle of radius 1, which is easily calculated to be $P(E) = \exp(-1/2N)$. A noise variance $N$ of about 1/24 therefore yields an error probability per symbol in the range of about $6 \times 10^{-6}$.

The channel capacity of the Gaussian channel was calculated in Shannon's original papers [1]. Subject to an energy constraint $x^2 < S$, the capacity is

$$C = (1/2) \log_2 (1 + S/N) \text{ bits/dimension},$$

achieved when $x$ is a zero-mean Gaussian variable with variance $S$. Note that when $S$ becomes large with $N$ constant, it takes approximately (asymptotically, exactly) 4 times as much power to increase the capacity by an additional 1 bit/dimension. Therefore the ratio of bits/dimension achieved by PAM to channel capacity approaches 1 as $S$ becomes large. This fact was used in [2] to argue that coding has little to offer on highly band-limited channels.

We can make a more quantitative estimate of the potential gains from coding as follows. Using narrow-sense QAM (two-dimensional PAM) to send $m$ bits/dimension or $n = 2^m$ bits/symbol at an acceptable error rate (of the order of $10^{-3}$--$10^{-6}$) requires an average energy in each dimension of

$$S = (2^n - 1)/3$$

when $N = 1/24$, or $S/N = 8 \times 2^n$ for $n$ moderately large. If channel capacity could be achieved, we could send about $n' = \log_2(S/N)$ bits/symbol, or about $n + 3$ bits/symbol at the same signal-to-noise ratio. Thus, the potential gain is about 3 bits/symbol or, alternatively, about a factor of 8 (9 dB) of power savings.

Many authors (see, e.g., [3], [4]) regard the parameter $R_o$ as a better estimate than $C$ of the maximum rate that is practically achievable using coding. On the Gaussian channel, $R_o$ is [5]

$$R_o = (1/2) \log_2 (1 + S/2N) \text{ bits/dimension}.$$

It thus takes a factor of 2 (3 dB) more power to signal at $R_o$ than to signal at $C$. The maximum practical improvement obtainable by coding might therefore be estimated as of the order of 6 dB, or 2 bits/symbol (although the $R_o$ estimate is not universally accepted).

In what follows we shall show that simple coding techniques gain about 3 dB or 1 bit/symbol, while the most elaborate techniques described have theoretical gains of the order of 6 dB or about 2 bits/symbol. This is entirely consistent with the $R_o$ estimate given above, and suggests that little can be gained by seeking still more elaborate schemes. (In [6] the capacity of the telephone channel was estimated as of the order of 23 500 bits/s, roughly consistent with what we are saying here.)

### III. Uncoded Modulation Systems

Digital QAM signaling schemes are conventionally and usefully represented by two-dimensional constellations of all possible signal points. A $2^m$-point constellation can be used to send $n$ bits/symbol. A fair amount of effort has gone into finding "optimum" constellations. We shall shortly see that the payoff for this effort on purely Gaussian-noise channels is relatively slight, although the schemes found are helpful precursors for more elaborate schemes.

#### A. Rectangular Constellations

A brief flurry of theoretical papers in the early 1960's [7]--[10] developed two-dimensional signal constellations from various viewpoints. The most interesting for our present purposes are the family of constellations developed by Campopiano and Glazer [9], reproduced in Fig. 3. (We have taken the liberty of substituting a "cross constellation" for theirs at $n = 7$; the two are equally good.) For even integer numbers of bits/symbol, the constellations are simply representations of two independent PAM channels, so the constellations are square and have points drawn from the rectangular lattice of points with odd-integer coordinates. It takes about 6 dB more power to send 2 more bits/symbol, as expected. For odd integer numbers of bits/symbol, the constellations lie within an envelope in the form of a cross (and have hence come to be called "cross constellations") and the points are drawn from the same rectangular lattice (except for the 8-point constellation, where the outer points are put on the axes for symmetry and energy savings). With the figures scaled so that the minimum distance between any two points is equal to 2, the average signal energy in absolute terms and in dB is as given in Table II. We see that the "cross constellations" require about 3 dB more or less than the next lower or higher square constellation, respectively, as we would expect.

The Campopiano-Glazer construction can be generalized as follows: from an infinite array of points closely packed in a regular array or lattice, select a closely packed subset of $2^n$ points as a signal constellation. This important principle is at the root of much recent work. We shall explore applications of this principle, working up from the simpler to the more sophisticated.

When constellations are drawn from a regular lattice within some enclosing boundary, the following asymptotic
Fig. 3. Rectangular signal constellations (after Campopiano and Glazer [9]).

TABLE II
CAMPOPIANO–GLAZER CONSTELLATIONS

<table>
<thead>
<tr>
<th>No. Pts</th>
<th>S (dB)</th>
<th>D (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>2.0</td>
<td>4.0</td>
</tr>
<tr>
<td>8</td>
<td>6.6</td>
<td>10.3</td>
</tr>
<tr>
<td>16</td>
<td>13.0</td>
<td>19.7</td>
</tr>
<tr>
<td>32</td>
<td>21.0</td>
<td>32.2</td>
</tr>
<tr>
<td>64</td>
<td>35.7</td>
<td>54.2</td>
</tr>
<tr>
<td>128</td>
<td>62.7</td>
<td>93.2</td>
</tr>
<tr>
<td>256</td>
<td>170.7</td>
<td>223.3</td>
</tr>
</tbody>
</table>

Fig. 4. Cross constellation boundary.

Fig. 5. Improved rectangular constellations.

Comparisons between \( S \) and \( \hat{S} \) are given in Table II; the approximations are good ones. Furthermore, note that the cross is slightly more efficient than the square, by a factor of \( 31/32 \) or 0.14 dB (because it is more like a circle); this suggests that the cross would be the better shape even for \( n \) even, and indeed this is the case and can be easily achieved by taking alternate points from the next higher cross constellation, as shown in Fig. 5 for \( n = 4 \) and \( n = 6 \). The \( n = 4 \) cross constellation is as good as the conventional \( 4 \times 4 \) square constellation, and the \( n = 6 \) cross constellation is 0.1 dB better than the \( 8 \times 8 \) square constellation, about as predicted.

Of course, the best enclosing boundary would be a circle, the geometrical figure of least average energy for a given area. A circle with radius \( R \) has area \( \pi R^2 \) and average energy equal to \( R^2/2 \); setting \( R^2 = 4 \times 2^n \), we find that the average energy for a \( 2^n \)-point circular constellation ought to be about

\[
\hat{S} = (2/\pi)2^n \quad \text{(circle)}
\]

which would be only about \( \pi/3 \) or 0.20 dB better than the square, or 0.06 dB better than the cross. Fig. 5 also shows a
more circular constellation for $n = 6$, with the outer points moved to the axes as in Campopiano–Glazer's 8-point constellation; this constellation has been used in the Paradyne 14.4 kbit/s modem and, like the $n = 6$ cross, is about 0.1 dB better than the $n = 6$ square.

**B. Hexagonal Constellations**

As the densest lattice in two dimensions is the hexagonal lattice (try penny packing), constellations using points from a hexagonal lattice ought to be the most efficient. Indeed, the area of the hexagonal Voronoi region for a hexagonal lattice with minimum distance 2 is $2\sqrt{3} = 3.464$, or 0.866 the size of the square region, which according to our approximation principle should translate to a 0.6 dB gain for a hexagonal constellation over a rectangular one with the same boundary. (A hexagonal boundary, as suggested in [12], has an energy efficiency within 0.03 dB of the circular boundary, or 0.03 dB better than the cross.)

Fig. 6 shows the best hexagonal packings for $n = 2$ through 6. For $n \geq 4$, the predicted 0.6 dB gain is effectively obtained over the best rectangular packings. (Historical notes: suggestions that the hexagonal lattice would asymptotically be the best were made very early; see, e.g., [13]. The suboptimal $n = 3$ "double diamond" structure was actually used in a 4800 bit/s Hycom modem in the mid-1970's. There was a great deal of attention to $n = 4$ structures in the early 1970's because of their importance in 9600 bit/s modems; the rather strange-looking one shown here was apparently first discovered by Foschini et al. [14], and is still the best 16-point constellation known. The $n = 6$ suboptimal structure is used in the Codex/ESE SP14.4 modem.)

**IV. ELABORATIONS OF UNCODED MODULATION**

In this section we shall discuss further variants of un­coded modulation: constellations with nonuniform probabilities, higher-dimensional uncoded constellations, and constellations for nonintegral numbers of bits/symbol.

**A. Nonuniform Probabilities**

Attainment of the channel capacity bound requires that the signal points have a Gaussian probability distribution, whereas with all the constellations of the previous section it is implicit that points are to be used with equal probabilities. A uniform circular distribution of radius $R$ has average energy $S_c = \frac{R^2}{2}$ and entropy $H_c = \log_2 \pi R^2$; a
two-dimensional Gaussian distribution of variance $\sigma^2$ in each dimension has average energy $S_g = 2\sigma^2$ and entropy $H_g = \log_2 2\pi e \sigma^2$. Thus,

$$H_c = \log_2 2\pi \sigma^2 c$$
$$H_g = \log_2 2\pi e S_g$$

To yield the same entropy, the Gaussian distribution requires a factor of $e/2 = 1.36$ (or 1.33 dB) less average energy than the circular distribution.

Implementation of a constellation with nonuniform probabilities presents a number of practical problems. One possible way of achieving some of the potential gain is to divide the incoming data bits into words of nonuniform length according to a prefix code, and then to map the prefix code words into signal points drawn as before from a regular two-dimensional lattice. The probability associated with a prefix code word of length $t$ bits is then $2^{-t}$. For example, Fig. 7 gives a set of prefix code words and a mapping onto the hexagonal lattice that yields an average energy of $S = 7.02$ while transmitting an average of 4 bits/symbol, an improvement of close to 1 dB over the best $n = 4$ uniform code known. Of course, the fact that the number of data bits transmitted per unit time is a random variable leads to system problems (e.g., buffering, delay) that may outweigh any possible improvement in signal-to-noise margin.

### B. Higher-Dimensional Constellations

It is possible to achieve the same gain in another way by coding blocks of data into higher-dimensional constellations without going to the true block coding to be described in later sections. (By “true coding,” we refer to schemes in which the distance between sequences in a higher number of dimensions is greater than that between points in two dimensions.) We have already seen in Section III-A that a small (0.2 dB) gain is possible by going from one-dimensional PAM to two-dimensional QAM and choosing points on a two-dimensional rectangular lattice from within a circular rather than a square boundary. In the same way, by going to a higher number $N$ of dimensions and choosing points on an $N$-dimensional rectangular lattice from within an $N$-sphere rather than an $N$-cube, further modest savings are possible. Table III gives the energy savings possible in $N$ dimensions, based on the difference between average energy of an $N$-sphere versus an $N$-cube of the same volume. Note that as $N$ goes to infinity, the gain goes to $\pi e/6$ (by the Stirling approximation, $(n!)^{-1/n}$ goes to $e/n$), or 1.53 dB; the improvement over $N = 2$ goes to $e/2$ or 1.33 dB, as computed above. This is because for large $N$ the probabilities of points in any two dimensions become nonuniform and ultimately Gaussian. (It seems remarkable that a purely geometric fact like the asymptotic ratio of the second moment of an $N$-sphere to that of an $N$-cube can be derived from an information-theoretic entropy calculation in 2-space, but so it can.)

Implementation of such a scheme also involves added complexity that may outweigh the performance gain. To send $n$ bits/symbol in $N$ dimensions (assuming $N$ even), incoming bits must be grouped in blocks of $Nn/2$. Some sort of mapping must then be made into the $2^{Nn/2}$ $N$-dimensional vectors with odd-integer coordinates (assuming a rectangular lattice) which have least energy among all such vectors. This can rapidly become a huge task; and a corresponding inverse mapping must be made at the receiver. Compromises can be made to simplify the mapping, at the cost of some suboptimality in energy efficiency; e.g., the cross is an effective compromise between the square and the circle in two dimensions.

### C. Nonintegral Number of Bits/Symbol

It is sometimes desirable (as we shall see in Section VI) to transmit a nonintegral number of bits/symbol. Since in general an additional 1 bit/symbol costs about an additional 3 dB, it ought to be possible to send an additional 1/2 bit/symbol for about 1.5 dB. In this section we give a simple method that effectively achieves such performance. The method can be generalized to other simple binary fractions at the expected costs, but we shall omit the generalization here.

To send $n + 1/2$ bits/symbol, we proceed as follows. Use a signal constellation comprising $2^n$ "inner points" drawn from a regular grid, such as any of those of Section III, and an additional $2^n-1$ "outer points" drawn from the same grid and of as little average energy as possible, subject to whatever symmetry constraints may be imposed. Incoming bits are then grouped into blocks of $2n + 1$ bits and sent in two successive symbol intervals as follows. One bit in the block determines whether any outer point is to be used. If not, the remaining $2n$ bits are used, $n$ at a time, to select two inner points. If so, then one additional bit selects which of the two signals is to be an outer point, $n - 1$ bits select which outer point, and the remaining $n$ bits select which inner point for the other signal. (That is, at most one outer point is sent.) With random data, the average
energy is 3/4 the average energy of inner points plus 1/4 the average energy of outer points. Fig. 8 shows constellations of 24, 48, 96, and 192 points that can be used in such schemes for $4 \leq n \leq 7$; the average energy in all cases for $n + 1/2$ bits/symbol is approximately halfway between that needed for $n$ and that for $n + 1$ bits/symbol. Thus, these constellations are intermediate between the Campopiano/Glazer constellations in the same way that the cross constellations are intermediate between the squares. (In fact, it can be shown that the 2-dimensional cross constellations can be derived from 1-dimensional PAM constellations with “inner” and “outer” points in an analogous way.)

V. CODING FUNDAMENTALS

Heretofore we have been concerned with methods of mapping input bits to signal point constellations in two or more dimensions, where in higher dimensions the bits simply lie on the lattice that is the Cartesian product of two-dimensional rectangular lattices, so that the distance between points in $N$-space is no different from that in two dimensions. Now we shall begin to discuss methods of coding of sequences of signal points, where for the purposes of this paper we mean by coding (or “channel coding”) the introduction of interdependencies between sequences of signal points such that not all sequences are possible; as a consequence, perhaps surprisingly, the minimum distance $d_{\text{min}}$ in $N$-space between two possible sequences is greater than the minimum distance $d_o$ in 2-space between two signal points in the constellation from which signal points are drawn. Use of maximum likelihood sequence detection at the receiver yields a “coding gain” of a factor of $d_{\text{min}}^2/d_o^2$ in energy efficiency, less whatever additional energy is needed for signaling. (In practice, some of the “coding gain” may be lost due to there being a large number of sequences at distance $d_{\text{min}}$ from the correct sequence and therefore a large number of possibilities for error, called the “error coefficient” effect. We shall not be able to discuss the “error coefficient” much in this paper, but offer some general remarks at the end of Section VIII.)

Conventional coding techniques cannot be directly applied in conjunction with band-limited modulation techniques, at least with significant gain. (In 1970–1971, at least four companies prototyped conventional coding schemes for use in high-speed modems; two of the companies failed, and two shortly withdrew their products from the market.) In recent years, however, a number of effective coding techniques have been developed for such applications. The most important point to be made in this paper is that all of these coding schemes can be developed from a common conceptual principle. This principle was set forth clearly by Ungerboeck [15], who called it “mapping by set partitioning,” although its roots may perhaps be found elsewhere as well. We describe it in this section, and in succeeding sections then use it to develop all known and some new coding schemes, both block and trellis.

We shall consider only 2-dimensional constellations with points drawn from a 2-dimensional rectangular grid. (From research to date, we cannot find any advantage to starting with hexagonal grids when higher orders of coding are to be used.)

Such a constellation can be divided into two subsets by assigning alternate points to each subset; i.e., according to the pattern

```
.. . . . . . A B A B A

B A B A B ..
```

The resulting two subsets ($A$ and $B$) have the following properties.

(a) The points in each subset lie on a rectangular grid (rotated 45° with respect to the original grid).

(b) The minimum squared distance between points within a subset is twice the minimum squared distance $[d_o^2]$ between points in the original constellation.

Furthermore, because of the first property, the partitioning can be repeated to yield 4, 8, 16, ... subsets with similar properties, and in particular within-subset squared distances of 4, 8, 16, ... times $d_o^2$. Fig. 9 shows the 64-point square constellation divided into two subsets of 32 points, 4 subsets of 16 points, 8 subsets of 8 points, and
This nomenclature is different from that of Ungerboeck [15], who uses a more natural subscript notation that reflects the successive binary partitions; our nomenclature will be useful in the next section, where we shall use the fact (evident by inspection) that at both the 4-subset and 8-subset levels, the minimum squared distance between two \( A \) subsets, say, is \( 2d^2 \) times the Hamming distance between their subscripts: e.g., \( d^2(A_0, A_1) = 2d^2; \)
\( d^2(A_{00}, A_{01}) = 2d^2; \) and so forth.

These subsets may then be used to implement relatively simple but effective coding schemes, illustrated in general in Fig. 10. Certain incoming data bits are encoded in a binary encoder, resulting in a larger number of coded bits. The coded bits are then used to select which subsets are to be used for each symbol. The remaining incoming bits are not coded, but merely select points from the selected subsets, with the signal constellation chosen large enough to accommodate all incoming bits. The coding scheme is thus more or less decoupled from the choice of constellation, as long as it is of the rectangular grid type. The coding gain is effectively determined by the distance properties of the subsets combined with those of the binary code, regardless of the size of the constellation. On the other hand, the constellation size, boundary, symmetries, and other "uncoded" properties such as were investigated in earlier sections are more or less independent of the coded bits and are determined by the mapping of the remaining bits. This may be regarded as effectively decoupling "source coding" from "channel coding" and is important both conceptually and in implementation.

We now show how this general scheme can be applied to both block and trellis codes, with performance approaching the \( R_c \) estimate.

### VI. BLOCK CODES

In Section IV-B we saw what could be achieved by using points from an \( N \)-dimensional rectangular lattice and using an \( N \)-sphere, rather than an \( N \)-cube, as a boundary.

The rectangular lattice comprising all \( N \)-dimensional vectors with odd-integer coordinates is not the most densely packed for any \( N \) greater than 1; for example, for \( N = 2 \), the hexagonal lattice is 0.6 dB denser, as we have seen. Finding the densest lattice in \( N \) dimensions is an old and well-studied problem in the mathematical literature. Table IV gives the densest packings currently known for all \( N \) up to 24 and selected larger \( N \), with the improvement in packing density over the rectangular lattice given in absolute and in dB terms [16]. The dimensions \( N = 4, 8, 16, \) and 24 are locally particularly good and are known to be optimum in the sense of being the densest possible lattice packing in these dimensions.

A body of recent work [11], [12], [18]–[21] generalizes the Campopiano-Glazer construction to \( N \) dimensions by taking all points on the densest lattice in \( N \)-space that lie within an \( N \)-sphere, where the radius of the sphere is chosen just large enough to enclose \( 2^m \) points, to send \( m \) bits per dimension. These codes obtain a "coding gain" over PAM which is a combination of both the lattice packing density gain of Table IV ("channel coding") and the \( N \)-sphere/\( N \)-cube boundary gain of Table III ("source coding"). The resulting coding gains achievable for \( N = 2, \)

![Fig. 10. General coding scheme.](image-url)
4, 8, 16, 24, 32, 48 and 64 are shown in Table V. Because the number of near neighbors in these densely packed lattices becomes very large, the total number of error events (“error coefficient”) becomes large, which reduces the coding gain realized in practice. Also, the mapping of all $mN$ bit combinations to their corresponding signal points can be a monumental task, even if the number of symmetries and simplifications are cleverly exploited [20], [21].

We will now show that certain of these dense $N$-dimensional lattices can be constructed using 2-dimensional rectangular lattices, the subset partitioning idea, and simple binary block codes. In particular, we shall give constructions for $N = 4, 8, 16$ and 24 that form a natural sequence both in complexity and in nominal coding gain (respectively 1.5, 3.0, 4.5, and 6.0 dB, using the simplest implementations). (Cusack [21a] has recently shown how to construct dense 2$^{2}$-dimensional lattices from 2-dimensional lattices using Reed–Muller codes, for any $n$; for $N = 4, 8,$ and 16, the lattices obtained are the same as those we obtain here.)

To generate the optimum $N$-dimensional lattices for $N = 4, 8, 16,$ and 24, we shall use sequences of 2, 4, 8, and 12 points from the 2-dimensional rectangular lattice, partitioned as shown in Fig. 9, into 2, 4, 8, and 16 subsets, respectively.

The 4-dimensional lattice is generated by taking all sequences of two points in which both points come from the same subset, i.e., sequences of the form $(A, A)$ or $(B, B)$.

The 8-dimensional lattice consists of all sequences of four points in which all points are either $A$ points or $B$ points and further in which the 4 subset subscripts satisfy an overall parity check, $i_1 + i_2 + i_3 + i_4 = 0$; e.g., sequences of the form $(A_0, A_0, A_0, A_0), (B_0, B_1, B_0, B_1),$ and so forth. (In other words, the subscripts must be codewords in the (4, 3) single-parity-check block code, whose minimum Hamming distance between codewords is 2.)

The 16-dimensional lattice consists of all sequences of eight points in which all points are either $A$ points or $B$ points, and further, in which the 16 subset subscripts (each subset now having two subscripts) are codewords in the (16.11) extended Hamming code, whose minimum Hamming distance between codewords is 4.

The 24-dimensional lattice consists of all sequences of 12 points in which all points are either $A$ points or $B$ points; the 24 $(i, j)$ subscripts are codewords in the (24, 12) Golay code, known to have minimum Hamming distance 8, and further, in which the third subscripts $k$ are constrained to satisfy an overall parity check in the following way: if the sequence is of all $A$ points, then overall $k$ parity is even (an even number are equal to 1), while if the sequence is of all $B$ points, overall $k$ parity is odd.

If the minimum squared distance between points in the 2-dimensional constellation is $d_2^2$, then the minimum squared distance between points (sequences) in these higher-dimensional lattices can be shown to be $2d_2^2$, $4d_2^2$, $8d_2^2$, and $16d_2^2$, respectively, as follows.

a) A sequence of $A$ points and a sequence of $B$ points differ from each other by squared distance at least $d_2^2$ in every point and therefore by at least $2d_2^2$, $4d_2^2$, $8d_2^2$, and $12d_2^2$ in total. In fact, in the 24-dimensional case there is a distance of at least $5d_2^2$ in at least one symbol, so the minimum squared distance between $A$ sequences and $B$ sequences is at least $16d_2^2$. The proof depends on the properties of the Golay code as well as the particular partitioning shown in Fig. 9 and is in the Appendix.

b) Two different sequences with points all from the same sequence of subsets must differ in at least one point by the minimum within-subset squared distance, which is $2d_2^2$, $4d_2^2$, $8d_2^2$, or $16d_2^2$, respectively. This is all we need to establish $2d_2^2$ as the minimum squared distance between sequences in the 4-dimensional case.

c) For $N = 8, 16,$ and 24, the $(i)$ or $(i, j)$ subset subscripts are drawn from (4,3), (16,11), or (24,12) codes with minimum Hamming distances 2, 4, and 8, respectively. By the relation between subset Hamming distance $d_H$ and subset squared distance $d_S^2 = 2d_Hd_2^2$ given in Section V, two sequences with points drawn from subsets of the same type ($A$ or $B$) but different $i$ or $(i, j)$ subscripts must differ by squared distance at least $4d_2^2$, $8d_2^2$, or $16d_2^2$, respectively. This is all we need for the 8- and 16-dimensional cases.

d) For $N = 24$, two sequences of points from subsets of the same type and with the same $(i, j)$ subscripts but different $k$ subscripts must differ by at least $8d_2^2$ in at least two symbols because of the overall $k$ parity check, and the fact that the minimum squared distance between points of
the same type and with the same \((i, j)\) subscripts is \(8d^2\).
This concludes the 24-dimensional proof.

To send \(m\) bits/symbol using these lattices, we need to encode a block of \(mN\) bits into one of \(2^{mN}\) lattice points. To maximize coding gain, the \(2^{mN}\) lattice points of least energy should be chosen; however, implementation of the mappings from bits to points and vice versa becomes complex. Simpler methods will now be given, using the binary codes used to construct the lattices, and the constellations either of Fig. 3 (for \(N = 8\) and 24) or of Fig. 8, along with the method of sending half-integral numbers of bits/symbol given in Section IV-C (for \(N = 4\) and 16). The cost in coding gain is relatively small, ranging from a few tenths of a decibel for \(N = 4\) or 8, up to about 1 dB for \(N = 24\); it is upperbounded by the \(N\)-sphere/\(N\)-cube gain given in Table III.

Of the block of \(mN\) bits, we always use one bit to specify whether \(A\) or \(B\) points will be used. For \(N = 8, 16,\) and 24, a further set of bits is used as input to a binary block coder, which produces appropriate codewords to be used as subset subscript designators: 3 bits to produce 4 for \(N = 8, 11\) bits to produce 16 for \(N = 16,\) and \(12 + 11 = 23\) bits to produce \(24 + 12 = 36\) for \(N = 24\). Thus, a total of 1, 4, 12, or 24 incoming bits are used as in Fig. 10 to select the subsets, or \(\frac{1}{2}, 1, \frac{3}{2},\) or 2 bits/symbol, respectively.

The remaining bits are used to select points from the selected subsets. We use the rectangular constellations of Figs. 3 and 8 as follows: for \(N = 4,\) constellations of \(1.5 \times 2^m\) points as in Fig. 8, divided into two \(1.5 \times 2^{m-1}\) point subsets \(A\) and \(B;\) for \(N = 8,\) constellations of \(2^{m+1}\) points as in Fig. 3, divided into four \(2^{m-1}\)-point subsets; for \(N = 16,\) constellations of \(1.5 \times 2^{m+1}\) points as in Fig. 8, divided into eight \(1.5 \times 2^{m-2}\) point subsets; and for \(N = 24,\) constellations of \(2^{m+2}\) points as in Fig. 3, divided into 16 \(2^{m-2}\)-point subsets. (In all cases \(m\) must be large enough so that the subsets resulting from the partitioning have equal size and other desired properties, e.g., symmetries.) These can be used to send \(m - \frac{1}{2}, m - 1, m - 1 \frac{1}{2},\) or \(m - 2\) bits/symbol, where for \(N = 4\) and 16, the method of sending half-integral bits/symbol of Section IV-C may be used.

Since the minimum squared distance in \(N\)-space is 2, 4, 8, or 16 times the minimum squared distance for an uncoded \(2^m\)-point constellation, there is a distance gain of 3, 6, 9, or 12 dB, respectively. However, the expanded constellations required with coded modulation cost 1.5, 3, 4.5, and 6 dB, respectively, yielding a net coding gain of 1.5, 3, 4.5, and 6 dB for \(N = 4, 8, 16,\) and 24. The family relationship of this progression of codes is apparent.

Maximum likelihood sequence detection of the lattice point closest to a sequence of received points is easy for \(N = 4, 8,\) and 24. General methods are given in [22]. For \(N = 8,\) given four received points, assume first that \(A\) points were sent. Find the closest \(A\) point to each received point, and check subscript parity of the four subsets tentatively decided. If the parity check fails, change the least reliable decision to the next closest \(A\) point (which must be in the other \(A\) subset). This gives the best \(A\) sequence satisfying the parity-check constraint. Repeat, assuming that \(B\) points were sent, to get the best \(B\) sequence. Compare the best \(A\) and \(B\) sequences, and choose the better as the final decision.

Detection for \(N = 16\) or 24 is harder, involving either generalization of soft-decision decoding of the (16,11) or (24,12) block codes to perform error correction on tentative decision subscripts (as above); the method of changing the least reliable decision if an overall parity check fails was used in the 1950's as a soft-decision error-correction method for single-parity-check codes by Wagner, or exhaustive search of a neighborhood of the received sequence in \(N\)-space. Note that as its first step the decoder can always choose the closest point in each subset to each received point as representative of that subset, there being no reason to prefer any more distant point, and then proceed to determine the best sequence of subsets using those points (with their distances from the received point) as proxies for the corresponding subsets; the decoding task may thus be partitioned in the same way as coding is partitioned in Fig. 10.

VII. TRELLIS CODES

On power-limited channels (such as the satellite channel), convolutional coding techniques have more or less become the standard (although there are some who continue to champion block codes [23]). Generally, anything that can be achieved with a block code can be achieved with somewhat greater simplicity with a convolutional code. We have just seen that relatively simple \((N = 8)\) block codes can achieve of the order of 3 dB coding gain on band-limited channels, and relatively complex \((N = 24)\) block codes can achieve of the order of 6 dB. We shall now see that trellis codes can do the same, perhaps a bit more simply.

A. Ungerboeck Codes

For band-limited channels, the trellis codes came first, in the work of Ungerboeck [15]. In Ungerboeck's paper, to send \(n\) bits/symbol with two-dimensional modulation, a constellation of \(2^{m+1}\) points is used partitioned into 4 or 8 subsets. 1 or 2 incoming bits/symbol enter a rate-\(\frac{1}{2}\) or rate-\(\frac{3}{2}\) binary convolutional encoder, and the resulting 2 or 3 coded bits/symbol specify which subset is to be used. The remaining incoming bits specify which point from the selected subset is to be used.

The coding gain obtainable increases with the number \(M\) of states in the convolutional encoder. Ungerboeck's simplest scheme uses a 4-state encoder and achieves a nominal 3 dB coding gain (a factor of 4, or 6 dB, in increased sequence distance, less 3 dB due to use of the larger \(2^{m-1}\)-point constellation). His most complex scheme uses a 128-state encoder and gains 6 dB (the limit with 8 subsets and a \(2^{m-1}\)-point constellation since the within-subset distance is \(8d^2\) for a 9 dB gain, less the 3 dB due to the larger constellation). Table VI gives the coding gains obtained by
<table>
<thead>
<tr>
<th>States</th>
<th>Gain (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>2.0</td>
</tr>
<tr>
<td>8</td>
<td>3.0</td>
</tr>
<tr>
<td>16</td>
<td>3.0</td>
</tr>
<tr>
<td>32</td>
<td>3.5</td>
</tr>
<tr>
<td>64</td>
<td>4.0</td>
</tr>
</tbody>
</table>

Ungerboeck for these and intermediate numbers of states.

Decoding is assumed to be by the Viterbi algorithm [24] a maximum likelihood sequence estimation procedure for any trellis code. The complexity of such a decoder is roughly proportional to the number of encoder states. With these codes, each branch in the trellis corresponds to a subset rather than to an individual signal point; but if the first step in decoding is to determine the best signal point within each subset (the one closest to the received point), then that point and its metric (squared distance from the received point) can be used thereafter for that branch, and Viterbi decoding can proceed in a conventional manner. Fig. 11 gives trellises with branches labeled by subset for Ungerboeck's 4-state and 8-state codes.

(Note: the block codes of the previous section can be represented as trellises; Fig. 12 shows the trellises corresponding to the \( N = 4 \) and \( N = 8 \) codes. Viterbi decoding could therefore be used for them as well. It is interesting that the block code with 3 dB coding gain is also associated with a 4-state trellis, albeit decomposable into two parallel 2-state trellises. The \( N = 16 \) block code can similarly be associated with a 64-state trellis decomposable into two parallel 32-state trellises, and the \( N = 24 \) block code can be associated with a \( 2 \times 4096 \)-state trellis.)

8-state trellis codes with nominal 4 dB coding gain are in the process of being adopted as international CCITT standards for 9600 bit/s transmission over the switched (dial) telephone network [25] and potentially for 14.4 kbit/s transmission over private lines as well [26]. A slight variant [27] of the Ungerboeck scheme involving a nonlinear convolutional encoder is being used in these standards; with this variant, whose trellis is shown in Fig. 13, a rotation of a coded sequence is another coded sequence, so that differential coding techniques may be used. The distance properties and therefore coding gain of the variant are apparently identical to those of Ungerboeck's 8-state scheme.

B. Other Trellis Codes

The Ungerboeck codes seem to cover the range of possible coding gains with complexity of the order of what we might expect, and may therefore be taken as benchmarks of how much complexity is needed to achieve different coding gains in the 3-6 dB range. Can they be improved upon? From our research, the answer seems to be: yes, but not very much. In this section we shall describe two schemes that exhibit modest improvements and some new ideas: a 2-state code that has a nominal coding gain of almost 3 dB, and an 8-state trellis code with a coding gain...
The new idea in the 2-state scheme is to use a constraint of 4.5 dB. The new idea in the 8-state scheme is to use a 4-dimensional constellation rather than 2-dimensional as the basic constellation. (4-dimensional trellis codes have also been studied by Wilson [28] and Fang et al. [28a].)

1) 2-state Trellis Code: If a 2^n-point signal constellation is partitioned into two subsets of A points and B points, any channel coding is done using the (infinite) 2-state trellis. Down at the top of Fig. 14, it would appear at first glance that a 3 dB gain is obtained at no cost. This scheme sends n bits/symbol with no signal constellation expansion; further, any two sequences that start at one common node and end at another differ by a squared distance of at least 2d^2, because the paths differ by at least d^2 when they diverge and another d^2 when they merge, so the nominal coding gain is apparently a factor of 2 or 3 dB.

Of course, we cannot get something for nothing, and the fallacy in this scheme (called “catastrophic error propagation” in the convolutional coding literature) is that there are paths of infinite length starting from a common node that never remerge and have squared distance only d^2, e.g., any two paths of the form AXYZ... and BXYZ... (The “error coefficient” is infinite.)

One way of curing this problem is to terminate the trellis every b symbols by forcing it to a single node, illustrated in Fig. 14(b). In other words, at the bth symbol, the subset is constrained to be A or B, as necessary to reach the designated node. Only n - 1 bits can be used to determine the bth symbol, so there is a cost of 1 bit per b symbols of transmission capacity, but now a legitimate coding gain of 3 dB is obtained minus (1/b)×3 dB for the rate loss. The 8-space block code would operate in just this way if it used 2^n points, partitioned into A_0 and A_1 (see the top half of the trellis in Fig. 12); happily it is possible to insert a simple code made up of B points into the interstices of the A code lattice without compromising distance, and the additional bit involved in specifying A or B compensates for the bit lost at the fourth symbol, and allows a full 3 dB gain. This terminated trellis code may be regarded as a generalization of a single-parity-check block code.

Another way of gaining almost 3 dB while using a time-invariant trellis code is as follows. The signal constellation is modestly expanded to include (1 + p)2^n points, arranged on a rectangular grid and divided as usual into A and B subsets of (1 + p)2^{n-1} points each. The A and B subsets are further divided into (1 - p)2^{n-1} “inner” points and 2p “outer” points. Finally, sets A', A'', B', and B'' have 24 common points, 8 unique points; e.g., A' = 24 A inner points plus 8 A' outer points, S' = 31.3; S'' = 80; 5 = 43.5 (16.4 dB).

Fig. 14. 2-state trellises. (a) Infinite nonredundant 2-state trellis. (b) 2-state trellis terminated every 4 symbols. (c) Time-invariant 2-state trellis with A' ≠ A'', B' ≠ B''.

Fig. 15. 80-point signal constellation. 32-point subsets A', A'', B', B'' have 24 common points, 8 unique points; e.g., A' = 24 A inner points plus 8 A' outer points. S' = 31.3; S'' = 80; 5 = 43.5 (16.4 dB).
squared distance between two paths beginning and ending at common nodes remains $2d_0^2$ since the basic distance properties between $A$ subsets and $B$ subsets remain. But now, although there are still pairs of infinite sequences comprising only inner points that start from a common node and never accumulate distance of more than $d_0^2$, their probability is zero. The squared distance between any branch that uses an outer point and its counterpart branch must be at least $2d_0^2$ since the counterpart branch cannot use the same outer point and the distance between different points in the same subset is at least $2d_0^2$. For practical purposes, this means that a sequence containing such a branch cannot be confused with a sequence containing the counterpart branch. Their squared distance being at least $3d_0^2$, so that, whenever an outer point is sent, reconvergence to a common node is forced, as in the trellis termination method. Here, however, the reconvergence is probabilistic and happens on average every $1/p$ symbols, e.g., every 4 symbols if $p = 0.25$.

There is a slight reduction in power due to the increased constellation size; e.g., the average power using the constellation of Fig. 15 is 43.5 or 16.4 dB, versus 42 or 16.2 dB for the $8 \times 8$ constellation, or 41 (16.1 dB) for the Fig. 5 $n = 6$ cross constellation. Use of an integral approximation gives an estimate of additional power required of a factor of $1 + p^2$, or 1.0625 (0.26 dB) for $p = 0.25$. This can be made as small as desired by reducing $p$; the cost, however, is a greater average time to converge and a greater average number of near-neighbor sequences ("error coefficient") increasing inversely proportional to $p$; for this code the "error coefficient" is rather large and must be taken into account. Any $p$ greater than zero in principle avoids catastrophic error propagation; a $p$ of about 0.25 seems a good choice in practice.

2) 8-State Trellis Code: For the 8-state four-dimensional scheme, we shall use a two-dimensional rectangular grid divided into four subsets as before. The binary convolutional encoder for this scheme, however, operates on pairs of symbols rather than single symbols. An appropriate encoder is shown in Fig. 16. During each pair of symbol intervals, three bits enter the encoder and four coded bits are produced. The first two coded bits select the subset for the first symbol and the second two bits select the subset for the second symbol.

If the Hamming distance between two encoded sequences is $K$, then the squared distance between the mappings onto grid points is at least $Kd_0^2$. We now show that the minimum free Hamming distance of this convolutional code is 4. First note that the response of the encoder to a single 1 on any input line is a sequence with even weight, from which it follows that all encoded sequences have even weight and the minimum free distance is even. By inspection, it is easy to verify that there is no encoded sequence of weight 2, and a simultaneous 1 on all inputs yields an encoded sequence of weight 4. Thus, the squared distance between any two sequences corresponding to different encoded outputs is at least $4d_0^2$. If the encoded outputs are the same, but different elements are chosen from the same subsets, then again the squared distance is at least $4d_0^2$.

Suppose now that the signal constellation contains $2^n$ points. Then, over two symbol intervals, $2n-1$ bits enter the modem and one parity check is generated, giving $2n$ bits to select the two signal points. Since $n - \frac{1}{2}$ bits/symbol enter the modem, there is a loss of 1.5 dB due to the larger signal constellation and a gain of 6 dB in distance for a net nominal coding gain of 4.5 dB.

Section IV discussed encoding for a half-integer number of bits/symbol, and that method can be applied here. An alternative which is somewhat more attractive is to have an integer number $n$ of bits/symbol enter the modem, with three bits entering the convolutional encoder each pair of symbols and $2n-3$ bits entering a prefix-code source coder as described in Section IV. This both yields an integer number of bits/symbol and also gains some of the possible 1.33 dB for nonuniform probabilities.

VIII. CONCLUDING REMARKS

It has not been possible in this paper to cover a number of topics that are of importance in practice.

The only channel disturbance considered has been white Gaussian noise. Other disturbances are usually controlling on telephone channels. There is some accumulating experience that the coded modulation schemes are often more robust relative to un-coded schemes than would be predicted by Gaussian noise calculations against some important disturbances, such as nonlinear distortion and phase jitter, perhaps due to the memory inherent in coded modulation and sequence estimation over multiple symbols.

Because of the symmetries of attractive constellations, e.g., the 90° symmetry of most of our rectangular constellations, there may be an ambiguity in phase at the receiver. In general, there are two ways to handle this ambiguity with coded modulation. If the code is such that, on a sequence basis, every 90° rotation of a code sequence is another legitimate code sequence, then it will be possible by differential quadrantal coding to make end-to-end transmission transparent to 90° rotations. Alternatively, if 90° rotations do not give valid code sequences in general, then it will be possible eventually to detect this and to force receiver phase to a valid setting. The former technique is generally preferred. Of the codes we have discussed, the block codes generally are differentially codable.
and the trellis codes are generally not, although they can often be modified to be; e.g., the modification of Fig. 11

...section in Fig. 13.

...we have mostly used nominal coding gain as a

...of merit of coding schemes. In fact, error probabili-

...ties for coded systems on Gaussian channels are typically

...[24] of the form \( P(E) = K \exp(-E) \), where the exponent

...\( E \) is governed by the nominal coding gain and the "error coefficient" \( K \) is of the order of the number of coded sequences at minimum distance from an average transmitted sequence. In general, the error coefficient

...a) increases with the complexity of coding;

...b) can cost a significant fraction of a dB for coding schemes with moderate (3–4 dB) gain, for error probabilities in the \( 10^{-5} \)–\( 10^{-6} \) range;

...c) can become very large for schemes with large (6 dB) gain, such as the block codes with \( N = 24 \), or the most complex trellis codes; and

...d) is generally significantly larger for block codes than for trellis codes with comparable nominal coding gain.

Thus the error coefficient cannot be ignored in a more detailed assessment of coded systems.

VIII. Summary

On the band-limited channel, dense packing of 2-dimen-

...sional constellations with optimal (circular) boundaries

...leads less than 1 dB improvement over simple pulse ampli-

...modulation. Uncoded schemes in higher dimensions

...or, alternatively, source coding can gain somewhat more than 1 dB by using signal points with nonuniform probabilities. These gains pale by comparison with what can be obtained with (channel) coding, where relatively simple block or trellis codes easily yield coding gains of the order of 3 dB, or 1 bit/symbol. Relatively complex block and trellis codes have been constructed that yield the order of 6 dB, or 2 bits/symbol. Because this is as much gain as would be predicted using the \( R_s \) estimate and is only 3 dB below the capacity limit, it seems unlikely that further major improvements are possible. However, within the spectrum of performance of already known schemes, there will likely be some further embellishments that will reduce implementation complexity or have other desirable properties, such as the differentially coded variant of Ungerböck's 8-state trellis code that is likely to become an international standard.

APPENDIX

Proof That 24-Space Lattice Has \( d_{\min}^2 = 16d_0^2 \)

If the grid of triple-subscripted signal points illustrated in Fig. 9 is rotated 45° with a point \( A_{000} \) at the origin, and

...scaled so that the coordinates \((x, y)\) of all points are

...integers, then the following hold true.

...a) For \( A \) points, both \( x \) and \( y \) are even; for \( B \) points, both \( x \) and \( y \) are odd.

...b) For \( A \) points, \( i = 0 \) iff \( y = 0 \) mod 4, and \( j = 0 \) iff \( x = 0 \) mod 4; for \( B \) points, \( i = 0 \) iff \( y = 1 \) mod 4, and \( j = 0 \) iff \( x = 1 \) mod 4.

...c) For \( A \) points, \( x + y = 2i + 2j + 4k \) mod 8; for \( B \) points, \( x + y = 2i + 2j + 4k + 2 \) mod 8.

Two sequences with points from different groups differ by at least 1 in every one of the 24 \((x, y)\) coordinates. They cannot all differ only by 1, however, because of the following. The sum \( S \) of all coordinates satisfies

\[
S = 2w_i + 4w, \text{ modulo } 8, \quad \text{for } A \text{ sequences;}
\]

\[
2w_j + 4w, \text{ modulo } 8, \quad \text{for } B \text{ sequences}
\]

where \( w_i \) is the number of \((i, j)\) subscripts equal to 1, and \( w \) the number of \( k \) subscripts equal to 1. But since the \((i, j)\) subscripts form a Golay code word and all such words have weights equal to integer multiples of 4, and since \( w \) is even for \( A \) sequences and odd for \( B \) sequences by construction,

\[
S = 0 \text{ modulo } 8, \quad \text{for } A \text{ sequences;}
\]

\[
4 \text{ modulo } 8, \quad \text{for } B \text{ sequences.}
\]

Now suppose that there were a \( B \) sequence that differed from an \( A \) sequence by +1 in every coordinate, and let \( m \) be the number of coordinates in which the difference was +1. Since the sum \( S_A \) of the \( A \) coordinates is 0 mod 8 and the sum \( S_B \) of the \( B \) coordinates is 4 mod 8, and \( S_A - S_B = (m - (24 - m)) = 2m \) mod 8, it follows that \( 2m = 4 \text{ mod } 8 \), or \( m = 2 \text{ mod } 4 \). Now, the construction of the array is such that if a \( B \) point has an \( x \) coordinate 1 larger than the \( x \) coordinate of an \( A \) point, then the \( j \) subscript is the same, whereas if it is 1 smaller, then the \( j \) subscript is different; similarly a difference of +1 in \( y \) gives the same \( i \) subscript, while a difference of −1 gives the opposite one. Thus, \( m = w_i \). But, \( w_i = 0 \text{ mod } 4 \), so \( m = 0 \text{ mod } 4 \); contradiction. Hence, any \( B \) sequence must differ from every \( A \) sequence by at least 3 in one coordinate, Q.E.D.

This lattice and its distance properties were originally discovered by Leech [29].

REFERENCES


G. David Forney, Jr. (S'58-M'61-F'73) was born in New York, NY, on March 6, 1940. He received the B.S. degree in electrical engineering from Princeton University, Princeton, NJ, in 1961, and the M.S. and Sc.D. degrees in electrical engineering from the Massachusetts Institute of Technology, Cambridge, MA, in 1963 and 1965, respectively.

In 1965, he joined the Codex Corporation and became a Vice President and Director in 1970. Since 1982 he has been Vice President and Direc