DRA

A Final Report

on Grant No. NAG3-141

FOUR-DIMENSIONAL MODULATION AND CODING: AN ALTERNATE TO FREQUENCY-REUSE

Submitted to:

National Aeronautics and Space Administration
Lewis Research Center
21000 Brookpark Road
Cleveland, OH 44135

Attention: Dr. E. F. Miller

Submitted by:

S. G. Wilson
Associate Professor

H. A. Sleeper
Graduate Research Assistant

Communication Systems Laboratory
Department of Electrical Engineering
RESEARCH LABORATORIES FOR THE ENGINEERING SCIENCES
SCHOOL OF ENGINEERING AND APPLIED SCIENCE
UNIVERSITY OF VIRGINIA
CHARLOTTESVILLE, VIRGINIA

(NASA-CR-174453) FOUR-DIMENSIONAL MODULATION AND CODING: AN ALTERNATE TO FREQUENCY-REUSE Final Report (Virginia Univ.) 60 p HC A04/MF A01 CSCL 20N

N83-36300

Unclas
G3/32 15158

Report No. UVA/528200/EE83/107

Copy No. ______

September 1983

PREFACE

This report constitutes the final report to NASA Lewis Research Center under NAG 3-141, though it is not a summary of technical work performed. Other reports filed under this contract are:

1. "Channel Effects on Continuous-Phase Modulation, A Simulation Study," R. G. Harkness and S. G. Wilson, UVA/528200/EE82/101, June 1982.
2. "Convolutional Coding Combined with Continuous Phase Modulation," S. V. Pizzo and S. G. Wilson, UVA/528200/EE82/102, November 1982.
3. "An Improved Algorithm for Evaluating Trellis Phase Codes," M. G. Mulligan and S. G. Wilson, UVA/528200/EE82/103, November 1982.
4. "Rate 3/4 Convolutional Coding of 16-PSK: Code Design and Performance Study," S. G. Wilson, H. A. Sleeper, P. J. Schottler, and M. T Lyons, UVA/528200/EE82/105.
5. "Rate 2/3 Convolutional Coding of CPFSK," M. G. Mulligan and S. G. Wilson, UVA/528200/EE83/107.

TABLE OF CONTENTS

ABSTRACT

We discuss four-dimensional modulation as a means of improving communication efficiency on the band-limited Gaussian channel, with the four dimensions of signal space constituted by phase-orthogonal carriers ($\cos \omega_c t$ and $\sin \omega_c t$) simultaneously on space-orthogonal electromagnetic waves. "Frequency reuse" techniques use such polarization orthogonality to reuse the same frequency slot, but the modulation is not treated as four-dimensional, rather a product of 2-D modulations, e.g. QPSK.

It is well known that, higher-dimensionality signalling affords possible improvements in the power-bandwidth sense, [1-3]. We build upon this work to describe 4-D modulations based upon subsets of lattice-packings in 4-D, which afford simplification of encoding and decoding. Sets of up to 1024 signals are constructed in 4-D, providing a (Nyquist) spectral efficiency of up to 10 bps/Hz. Energy gains over the reuse technique are in the 1-3 dB range at equal bandwidth. Finally, trellis codes onto 4-D modulation sets are investigated as a means of further improving the power/bandwidth tradeoff. We focus upon codes with up to 4 states for R = 2, 3, and 4 bits/symbol interval.

1. INTRODUCTION

"Frequency-reuse" is a technique which utilizes two spatially-orthogonal electric field polarizations for communicating on the same carrier frequency to double the apparent spectral capacity of a satellite communications system. Provided the two fields can be kept orthogonal (admittedly a problem on some channels due to depolarization) then the spectrum efficiency is twice that of a non-reuse strategy, and the energy efficiency is exactly that of a single channel at the same E_b/N_o level. A typical application would perform quadrature phase shift keying (QPSK) on each polarization providing a theoretical spectral efficiency of 4 bps/Hz, with probability of bit error given by

$$P_b = Q\left(\left(\frac{2E_b}{N_o}\right)^{1/2}\right) \tag{1}$$

as for antipodal signalling.

Viewed more broadly, this signalling method may be treated as a special case of four-dimensional modulation, with two phase-orthogonal dimensions residing in each of two space-orthogonal directions.

The transmitted signal may be represented as

$$S_i(t) = \vec{u}_V(a_i \cos \omega_c t + b_i \sin \omega_c t) + \vec{u}_H(c_i \cos \omega_c t + d_i \sin \omega_c t) \tag{2}$$

where $\vec{u}_V$ and $\vec{u}_H$ denote unit vectors in the so-called "vertical" and "horizontal" orientations. Letting the orthonormal basis set be

$$\phi_o(t) = \vec{u}_V \sqrt{\frac{2}{T}} \cos \omega_c t$$

ORIGINAL PAGE IS
OF POOR QUALITY

$$\phi_1(t) = \vec{u}_V \sqrt{\frac{2}{T}} \sin \omega_c t$$

$$\phi_2(t) = \vec{u}_H \sqrt{\frac{2}{T}} \cos \omega_c t \qquad (3)$$

$$\phi_3(t) = \vec{u}_H \sqrt{\frac{2}{T}} \sin \omega_c t ,$$

we obtain a signal space representation of the i^{th} signal as the vector $\sqrt{T/2}\ (a_i, b_i, c_i, d_i)$. In this context, QPSK with frequency reuse provides a 16-ary constellation in 4-D with signals of the normalized form (±1, ±1, ±1, ±1), i.e. the vertices of a 4-cube centered at the origin. Because of the usual association of each of the four bits with ±1 modulation on a fixed dimension, minimum bit error probability detection can be achieved simply by sign detection in each coordinate position.

Figure 1 illustrates the block diagram of the modulator with the 2-D/reuse and 4-D perspectives. The hardware differences are surprisingly minor, indeed a system using polarization reuse already employs the required RF components to perform the more general 4-D modulation. Demodulation is likewise similar. The 4-D receiver employs quadrature carrier demodulation on each polarization, followed by matched filtering and decision making. Here lies the principal difference; the 2-D receiver utilizes two separate 2-D decision rules, while the general case uses a 4-D rule. For general constellations, this decision rule can be rather unwieldy, but in the case of 4-D lattice-based constellations, simple procedures are available.

The 4-D signal design problem is to locate M points in R^4 so that for a given minimum Euclidean distance between signals, the average (or peak) energy is minimized. More formally, letting $\bar{s}_i$ denote signal locations and $||\cdot||$ the usual norm, the problem is

$$\text{minimize } \frac{1}{M} \sum_{i=1}^{M} ||\bar{s}_i||^2$$

subject to

$$||s_i - s_j|| \geq d_{min} , \quad i \neq j$$

This is the classical sphere packing problem for which ample previous work has been done. We illustrate by discussing known results in 2D and 3D which are more easily perceived. In two-dimensions the best arrangement for large M places signal points on vertices of equilateral triangles which tesselate the plane. This is sometimes referred to as a hexagonal lattice, as the decoding regions are regular hexagons centered at each signal point. For finite M in 2-D, references [4] and [5] provide optimal constellations and certain symmetric constellations. As an example, the optimum M=16 constellation in 2-D has the arrangement shown in Figure 2a, while Figure 2b illustrates the standard 16-QASK design, which may be visualized as a Cartesian product of 1-D 4-level AM.

The optimum design is about 0.5 dB more efficient in use of energy (average), slightly more under a peak energy constraint, with both having the same spectral efficiency. This example points to the (slight) superiority of joint 2-D design rather than a standard iterated

1-D modulation. Of course, the optimal constellation is more complicated to implement, especially in the receiver detection circuitry.

Other interesting results are known in three dimensions [6]. For large M, the best packing is to place signals at centers of rhombic dodecahedra, regular polyhedra which have 12 faces and butt against 12 other signals. The centers or signal points lie on a face-centered cubic lattice. In the special case of M=8, we have a natural design using the 8 vertices of a 3-D cube. This design is again a product of 1-D antipodal modulation. Intuition suggests this might be the optimal arrangement of 8 points on a 3-D sphere, but a construction using tetrahedra, one inverted and "pushed through" the other (known as the antiprism) [6] provides a better distribution of points, by about 0.5 dB under peak and average energy constraints.

These examples indicate rather miniscule gains over a simple "product of 1-D" approach, but in general the gains are better, particularly for larger M. We have selected examples where the simple approach leads naturally to efficient constructions. In addition, the jointly-coded approach offers more flexibility. If we want M=16 points in 3-D the simple product designs such as 4-level AM x 4-ary QPSK give a 3-D constellation substantially poorer than the best placement of 16 signals on a sphere.

What we seek are 4-D signal constructions for M=8 through 1024 points which have superior energy efficiency to that obtained in a 2-D modulation-with-frequency-reuse approach. We shall concentrate on designs based upon 4-D lattices [7] as "fast" decoding algorithms for the Gaussian channel exist. The articles of Conway and Sloane [7-9] provide much of the groundwork in characterizing lattices in four dimensions and their packing properties.

Of primary interest is the lattice designated D_4, consisting of the points (x_1, x_2, x_3, x_4) whose integer coordinates have an even sum. For infinite lattices, D_4 has the property that the packing density is largest, i.e. 4-D spheres (decoding regions of a certain size) are packed most densely in 4-space, which provides an optimal signal design for the Gaussian channel, at least for large M. Edge effects, that is truncation of the lattice to obtain a number of points equalling a power of two, compromise this optimality somewhat, but D_4 provides a basis for investigation.

Previous related work may be found on 4-D modulation in the work of Welti and Lee [2] and of Zetterberg and Brandstom, [3,8]. Welti and Lee analyze several classes of codes for M ranging beyond a thousand and tabulate the energy versus bandwidth performance of the best codes. The Welti/Lee codes are essentially subsets of D_4, or translations of D_4, although the terminology is not used. Zetterberg and Brandstom concentrate on quaternion groups as constructions for 4-D codes and arrive at comparable performance for a smaller number of codes. These codes also have the property that signal vectors lie on a 4-D sphere (equal-energy), whereas the Welti/Lee codes are allowed to consume all of 4-space within a sphere. This equal-energy constraint is a significant penalty as M becomes large in the same way M-ary PSK becomes less efficient than M-ary amplitude/phase modulation in 2-D.

We next give a brief discussion of 4-D lattices and the cases of interest, prior to describing specific signal constellations for modulation and trellis codes built upon them.

2. FOUR-DIMENSIONAL LATTICES

An n-dimensional lattice is a regular set of points in m-dimensional space defined by

$$\bar{s} = u_1 \bar{a}_1 + \ldots u_n \bar{a}_n \tag{5}$$

where $\bar{s}$ is a m-dimensional column vector, u_i are integers and $\bar{a}_i$ are n linearly independent column vectors in R^m. Note $m \geq n$. The vectors $\bar{a}_i$ are a basis for the lattice in an integer-coefficient expansion.

Given such a lattice L, the dual lattice L* consists of all points $\bar{y}$ spanned by $\bar{a}_1$, $\bar{a}_2$, ... $\bar{a}_n$ such that $\bar{s} \bullet \bar{y}$ is integer-valued. Two lattices A and B are equivalent if their points may be mapped 1-1 by a coordinate rotation and scaling.

Cases of Interest

a) Z_4 is the set of all four-tuples with integer coordinates, and is dubbed the "integer lattice." We may define the bases as follows:

$$a_1^T = (1\ 0\ 0\ 0)$$
$$a_2^T = (0\ 1\ 0\ 0)$$
$$a_3^T = (0\ 0\ 1\ 0)$$
$$a_4^T = (0\ 0\ 0\ 1)$$

The minimum distance between points in this lattice is $d_{min} = 1$ as is seen by enumeration, and the "kissing number" is 8 (the kissing number τ is that number adjacent lattice points located at distance d_{min}).

b) D_4 is the set of all integer-valued 4-vectors with an <u>even</u> sum. As such it may be viewed as a punctured version of Z_4 where vectors with odd-sum are removed, and it is obvious that $d_{min} = \sqrt{2}$ by virtue of this puncturing. (We shall be careful to normalize for energy and distance later.)

A basis for D_4 is defined as (note [2] utilizes a different basis)

$$a_1^T = (2, 0, 0, 0)$$
$$a_2^T = (0, 2, 0, 0)$$
$$a_3^T = (1, 1, 1, 1)$$
$$a_4^T = (1, 1, 1, -1)$$

For D_4 the kissing number τ is 24 and D_4 represents the densest lattice packing for four-dimensions in the sense that among all lattice packings the largest number of unit radius spheres can be placed per unit volume.

D_4^*, the dual lattice of D_4, is best defined as $Z_4 \cup \{Z_4 + (1/2, 1/2, 1/2, 1/2)\}$, that is form the union of Z_4 and a translate of Z_4. As defined, $d_{min} = 1$, but it is known that D_4^* is equivalent to D_4 as defined above.

c) A_4 is formed by the set of all 5-dimensional integer vectors whose sum is zero, e.g. (3, -1, 0, -1, -1), (2, 0, -2, 0, 0), etc. Geometrically the lattice may be viewed as a hyperplane through Z_5 with the plane cutting the origin so $\sum_1^5 x_i = 0$. Since all the intersected points lie in a 4-D space we may assign the points to have 4-D coordinates to construct a signal constellation. For A_4 the kissing number τ is 20 and $d_{min} = \sqrt{2}$.

Decoding of Lattice Codes

Lattice constellations are of special interest due to their fast decoding procedures. Given a received vector $\bar{r} = (r_1, \ldots r_4)$ the task is to locate the closest point in the lattice for maximum likelihood decoding on the Gaussian channel. For the above lattices we describe simple procedures decoding [7]:

Z_4: Round-off each r_i to the nearest integer and adopt this integer vector as the codeword. This amounts to simple quantization of each signal coordinate independently.

D_4: Round-off $\bar{r}$ as above to produce an integer vector; if its sum is even, adopt it; if not, round the "worst" r_i the other way; the integer vector will then have an even sum.

D_4*: Repeat the algorithm for D_4 with offsets of $\bar{r}_o = (0, 0, 0, 0)$, $\bar{r}_o = (1/2, 1/2, 1/2, 1/2)$, $\bar{r}_o = (0, 0, 0, 1)$ or $\bar{r}_o = (1/2, 1/2, 1/2, -1/2)$, then pick the best among these four winners

A_4: The reader is referred to [7], pages 230-231, for a simple discussion of the procedure; in general this is a more complicated procedure than the preceding. Decoding can be done with 5-D or 4-D coordinates.

The above methods presume an infinite lattice with no attention to the fact that signal constellations are finite sets. Assuming the constellation is a full lattice out through some hypershell, then we decode as above and check the shell radius; if it does not exceed that for the constellation in use, the decoded point is accepted. If the decoded point is outside the constellation, we must re-decode to the nearest constellation point using some special rule.

We will also be interested in decoding constellations which are translated versions of a root lattice, say by $\bar{s}_o$. It is obvious that merely subtracting this vector from $\bar{r}$, then performing normal lattice decoding is optimal.

3. LATTICE SIGNAL DESIGNS IN 4-D: ASYMPTOTIC COMPARISONS

For 4-D lattices, it is known that D_4 (or its dual D_4^*) provide the densest packing of unit spheres per unit of 4-D volume. This suggests that D_4 will produce optimal signal constellations for the additive Gaussian channel since decoding regions for this problem are spheres. When the number of signals M selected from concentric shells becomes large, the ratio of average energy expended to squared minimum distance is $2\sqrt{M}/3\pi$ [2], and since the kissing number is 24 and there are $\log_2 M$ bits per signal, the error probability is given by

$$P[\varepsilon] \sim 24\, Q\left(\frac{d_{min}}{\sqrt{2N_o}}\right) = 24\, Q\left(\left(\frac{2E_b}{N_o}\left(\frac{\log_2 M}{M^{1/2}}\right)\left(\frac{1}{.81}\right)\right)^{1/2}\right) = 24\, Q\left(\left(\frac{2\bar{E}\ (.81)}{N_o\ M^{1/2}}\right)^{1/2}\right) \quad (6)$$

where E_b is the energy per bit and E is the energy per symbol.

For M = 64, the performance given by this asymptotic expression is

$$P[\varepsilon] \sim 24\, Q\left(\left(\frac{2E_b}{N_o}(.93)\right)^{1/2}\right) \quad (7)$$

which is asymptotically only 0.3 dB less efficient than QPSK transmission, but with 6 bits/4 dimensions rather than 4 bits/4 dimensions, i.e. 50% better spectral efficiency. At M = 1024, the expression gives

$$P[\varepsilon] \sim 24\, Q\left(\left(\frac{2E_b}{N_o}(0.39)\right)^{1/2}\right) \quad (8)$$

or 4.1 dB worse than QPSK, but with 2.5 times the spectral efficiency.

The packing density for the integer lattice, Z_4, is only half that of D_4 [8], while that of A_4 is rather close to that of D_4, namely 89%. To interpret this we say that within a large volume of

ORIGINAL PAGE IS
OF POOR QUALITY

R^4, if 100 unit radius spheres can be packed for Z_4, then 200 can be using the D_4 arrangement, and 179 can with A_4.

Stated in another way, suppose we wish M signals in Z_4, D_4, or A_4. The peak energy requirement to include this many signals is $E_p \sim .32\ M^{\frac{1}{2}}$ for D_4, $0.45\ M^{\frac{1}{2}}$ for Z_4 and $.34\ M^{\frac{1}{2}}$ for A_4. This projects a 1.5 dB advantage in peak energy for D_4 over Z_4 at equal M.

It is also known that in R^4 the peak-to-average energy ratio is 3/2 in the limit of a large number of points uniformly distributed within a hypersphere. This holds independently of the lattice so the relative efficiencies above hold for both peak and average energy comparisons.

Another more constructive comparison is provided by enumerating the lattice points and calculating $\bar{E}$, the average symbol energy, divided by d^2_{min}. This ratio is essentially the signal-to-noise ratio and can be related easily to $P[\varepsilon]$. This ratio is shown for Z_4, A_4 and D_4 in Figure 3. Points plotted correspond to those M with fully-populated shells, but these are typically not powers of 2. For a given M, we wish to achieve a certain d_{min} for the smallest possible $\bar{E}$, so D_4 is superior. For a given M, it appears that Z_4 requires about 1.5 dB additional energy, while A_4 requires about 0.2 - 0.3 dB higher energy, relative to D_4. Or at a given $\bar{E}/d^2$ ratio, D_4 can convey twice as many symbols as can Z_4. These are obviously consistent with packing theory described above.

Based on these asymptotic results, it is clear that D_4 is the proper construction for "large M," while A_4 is a close second. The slightly more complicated decoding for A_4 also penalizes it. It is possible however that edge effects may become significant for smaller M whereby the shell structure of the various lattices is a natural for certain small M. Also, we are interested in convenient values of M, perhaps not easily obtained with all lattices.

4. MODULATION SETS IN 4-D

We now describe explicit designs for $M=2^n$ in 4-D and evaluate these on both averag› and peak energy basis versus bandwidth. For all cases we define bandwidth in the Nyquist-sense, which says that (theoretically) a 4-D modulation as described can transmit $\log_2 M$ bits per symbol with a carrier signal bandlimited to a total bandwidth of $1/T_s$ where T_s is the 4-D symbol rate (note all basis functions are orthogonal and have the same spectral density). Since the symbol rate $R_s = 1/T_s$ is $R/\log_2 M$, we have that $B = R/\log_2 M$. The spectral efficiency is $R/B = \log_2 M$ bps/Hz. As an example, with $M = 64$ points in 4-D, $R/B = 6$ bps/Hz. This represents a lower bound on bandwidth actually, as attainment of the Nyquist limit, without any partial-response coding, necessitates unrealizable pulse shapes or transmission filters. We also note that the spectral efficiency depends only on M and not upon the constellation, whereas the energy efficiency does depend on signal placement.

Given a constellation of M points in 4-D, we let d_{min} be the minimum Euclidean distance between any pair of points. Let $\bar{E}$ be the average energy expended in transmitting one symbol. In general we can write

$$\bar{E} = k\, d_{min}^2 \qquad (9)$$

where k is a parameter of the design.

For a maximum likelihood deceiver, the asymptotic performance will be

$$P[\varepsilon] \sim N\,Q\left(\left(\frac{d_{min}^2}{2N_o}\right)^{1/2}\right)$$

$$= N\,Q\left(\frac{E}{2kN_o}\right)^{1/2} = N\,Q\left(\left(\frac{\bar{E}_b \log_2 M}{2kN_o}\right)^{1/2}\right) \quad (10)$$

where $\bar{E}_b$ is the average energy <u>per bit</u> and $N_o/2$ is the two-sided noise spectral density, with Q(y) being the one-sided Gaussian tail integral. N in (10) is a small constant reflecting the number of minimum distance pairs, but in comparing energy efficiency, only the argument of the Q-function is of interest.

As an example, we find that for for the M = 64 design given below, $\bar{E} = 1.688\ d^2$, giving

$$P[\varepsilon] \sim N\,Q\left(\left(\frac{3\bar{E}_b}{1.688\ N_o}\right)^{1/2}\right) \quad (11)$$

We may also represent P[ε] in terms of peak energy if such constraints are more important; the development is as above except we must write $E_p = k_2 d^2$ where $k_2 \geq k$ above.

Next we describe the performance of the iterated 2-D approach as a "4-D" construction for comparison purposes.

4.1 <u>Modulation in 4-D Using Product of 2-D Modulation</u>

The traditional frequency-reuse viewpoint is to perform 2-D modulation on each polarization, each independent of the other. This affords a certain simplicity and flexibility but as we show is inferior to the general 4-D modulation. We consider the types of 2-D modulation shown in Figure 4, all rectangular grid designs. These constellations are all subsets of Z_2 and are admittedly not optimum in 2-D, but have simple decoding regions and are commonly seen in applications literature. With each constellation we list the asymptotic error probability versus E_b/N_o (average), as well as the peak-to-average energy ratio.

When used in product fashion to achieve 4-D modulation, we shall plot such cases so that $E_{4D} = 2E_{2-D}$ and the number of signals is M^2. For example, 16-QASK in 2-D forms a 256-ary modulation in 4-D.

Figure 5 plots the energy versus spectrum performance of these 2-D product designs for M = 16, 64, 256 and 1024. We tabulate the energy efficiency relative to that of antipodal signalling (an M = 16 design formed by ±1 modulation on each basis function, or QPSK on each polarization).

4.2 <u>4-D Constellations with $M = 2^n$</u>

In practical digital transmission we are interested in sets whose size is a power of 2, so that exactly $\log_2 M$ bits are conveyed per symbol. Unfortunately the lattice shell populations do not in all cases match this requirement. Of course we can simply delete points from a bigger constellation until we reach a power-of-two, but this generally leaves a lack of symmetry and complicates decoding.

To search for desirable sets, we first used a computer to enumerate shells and cumulative counts through various shells for the lattices D_4, A_4 and Z_4. These results are tabulated in the Appendix. For each lattice, different offset vectors were added to move the origin within the lattice. This has the effect of changing shell counts and perhaps allows us to hit upon a good design.

To illustrate the use of these tables, we consider Table A1. The lattice, when no offset vector is applied, has 1 point at the origin, 8 points in the first shell of norm 1, 24 in the next shell, etc. In cumulative terms, there are 33 points through the first 3 shells. By simply deleting the origin we are left with 32 points in 4-D whose average energy is (33/32)(1.697) = 1.75. The figures of merit for modulation designs in $\bar{E}/d^2_{min}$ which in this case is 1.75 since $d_{min} = 1$. (We shall achieve a design from D_4 however with a smaller ratio). Also, we may observe a M = 64 point design by removing the origin and a 128 point design by removing the first two shells. Their respective $\bar{E}/d^2$ figures are 2.37 and 3.75. With offset vector of (0.5,0.5,0,0) we find an M = 16 design with $\bar{E}/d^2 = 1.5$, but again this will be inferior to the D_4 design. An improved M = 128 construction with full shells gives $\bar{E}/d^2 = 3.375$. With an offset of (0.5,0.5,0.5,0) we attain a M = 8 design with $\bar{E}/d^2 = 0.75$. To summarize, the best Z_4 designs found are listed in Table I.

It is of interest to compare the Z_4 designs with those cf Z_2 products of M = 16, 64, 256, and 1024. The respective values of $\bar{E}/d^2$ are 1.0, 03.0, 5.0, and 10.0 and comparison with the results of Table I shows little improvement, in fact M = 1024 is slightly worse in Z_4. If compared on a peak-energy comparison, the comparison swings in favor of Z_4 since by design we are keeping all signal points inside 4-D spheres. Nonetheless, the perfcrmance improvements with the Z_4 lattice are not substantial.

D_4 is the lattice of special interest based on mere consideration of packing density. With zero offset however, the shell populations do not readily match 2^n. Thus we repeated the enumeration procedure for

D_4 under different offsets with results tabulated in Appendix B. With zero offset, the "Class I" codes of Welti and Lee emerge for M = 25, 49, and 145 points, though the D_4 lattice terminology was not used in their earlier work.

In certain notable cases, fully-populated shells give convenient totals. Specifically with (1,0,0,0) offset applied to D_4, we then have the set of integer-vectors with odd sums, and the five shells with smallest radii contain exactly 256 points. Likewise, with an offset of (0.5,0.5,0,0) applied to D_4, we find 64 points in its first five shells (the radii are now different). Both of these designs have earlier been listed by Welti and Lee.

In other cases, we have studied the shell populations to find attractive combinations. These are listed in Table II. In general, the $\bar{E}/d^2$ and E_p/d^2 ratios are significantly smaller than those found for Z_4, as expected from the earlier discussion. For M = 64, the saving in average energy is 10 log (2.37/1.69) ≈ 1.5 dB, and the saving in peak energy is 1.25 dB. Compared to the use of 8x8 reuse (still M = 64), the respective savings are 2.5 dB and 2.5 dB.

16-ary designs which outerform the 4-D hypercube are difficult to find. Two which do so by 0.6 dB on an average energy basis, but not on a peak energy basis, are a design having a 2-8-6 shell structure and one with a 4-8-4 structure. The former is obtained with an offset of (0, 0.5, 0.5, 0.5) while the second is with (   ). The outer shell is partly-populated for both.

Comparison of these 4-D constructions with the product of 2D case is provided in Figure 5. We plot average energy efficiency relative to antipodal, versus Nyquist bandwidth, as described earlier in this section. Several observations may be made. First, there is a 32-point D_4 design having the same energy efficiency as QPSK/reuse, yet 25% greater spectral efficiency. The same comparison can be made between a 16x16 reuse strategy and 1024 points from D_4: the energy efficiency is virtually the same, but spectral efficiency is 25% greater. Viewed at a fixed spectral efficiency, we see gains in average energy of 1.5 - 2.5 dB for M = 64 up to 1024 while gains are less for smaller M. The energy gains are slightly better if peak energy is compared: at M = 256 the gain is another 1.3 dB in favor of the D_4 constellation.

Finally, we remark that the D_4 approach can provide a greater amount of communications flexibility than does the 2-D with reuse approach. As an example, M = 32 points in 4-D is conveniently attained from D_4, but a 2-D/reuse strategy to achieve the same throughput would necessitate a 4x8 design. Unless the power allocated to each polarization is made unequal, the performance is limited to that of the 8-ary polarization, about 3 dB worse than that of the 4-ary channel. For such cases the preferences for 4-D modulation is even more clear, saving roughly 3dB in average energy.

5. R_o, THE RANDOM CODING EXPONENT AND CUT OFF RATE

The parameter R_o for a modulation scheme is a measure of that modulation's utility as a code alphabet. Massey [11] and others have argued that when coding is contemplated, modulations ought to be designed by maximizing R_o instead of a more familiar optimization of bit error probability. For the ensemble of rate R convolutional codes it may be shown that the average symbol error probability is bounded by

$$\overline{P[\varepsilon]} < C_R\, 2^{-KR_o/R} \quad \text{for } R < R_o \tag{12}$$

where K is the constraint length and C_R is a constant independent of K. Thus maximizing R_o minimizes $P[\varepsilon]$ for a given rate. Also, R_o has the significance that sequential decoders have finite mean computation per decoded bit if $R < R_o$.

For the additive Gaussian channel [12]

$$R_o \stackrel{\Delta}{=} -\log_2\left[\frac{1}{M^2}\sum_i\sum_j e^{-d_{ij}^2 E_s/4N_o}\right] \frac{\text{bits}}{\text{symbol}} \tag{13}$$

where d_{ij} is the distance between signals i and j under a normalization where average energy $E_s = 1$. From (13), R_o tends to $\log_2 M$ bits/symbol as E_s/N_o increases.

We have numerically evaluated R_o for the 16-ary- 64-ary, and 256-ary constellations from D_4 described in the previous section, and results are shown in Figure 6 versus average energy per 4-D symbol. Note all curves reach a high SNR asymptote of $\log_2 M$ bit/symbol, while at low E/N_o, the curves coalesce, indicative of the expected result that large alphabets are no better than small ones for poor SNR. We also observe a key result for coding: to achieve a certain R_o of n bits per symbol, it is roughly sufficient to use a code alphabet having $2(2^n)$ symbols, i.e. doubling the set needed to communicate n bits in uncoded manner.

Figure 7 also plots R_o for two product of 2-D modulations, having 8x8 = 64 and 16x16 = 256 points. We earlier saw the power efficiency of these designs from an uncoded point of view. It is interesting that the differences in R_o are rather minor; in the region of the knee of the curve, where coded communication systems normally seek to operate, the 2-D product designs are about 0.5-1 dB less efficient. They have the same high SNR and low SNR asymptote however. This would seem to suggest that random coding arguments don't provide a strong preference for use of 4-D modulation over simpler 2-D products. If peak energy comparison are made, the 4-D approach becomes about 1 dB better still. We remark however that R_o considerations are not entirely reflective of the ability to produce good codes, especially for simple codes.

6. TRELLIS CODES FOR 4-D MODULATION

The 4-D modulation sets previously described may be used as a signal alphabet for trellis codes as a means of further enhancing the energy efficiency. Such codes can be optimally decoded with the Viterbi algorithm, although the trellis size must kept manageably small.

The theme of this work follows that of Ungerboeck [13], which proposed convolutional coding onto a signal set twice as l s needed for uncoded transmission, yet having the same dimensionali in this way, we may increase the minimum distance between coded sequences, while not expanding bandwidth. An example is mapping three information bits per interval onto a 16-ary modulation in 2-D, e.g. 16-QASK.

In the case of 2-D codes, the modulation symbols were assigned to trellis branches using a heuristic set partitioning concept, [13], which intuitively leads to good codes without resort to brute-force test of all possible codes of a given complexity. We apply this same methodology here with 4-D modulation, although the set partitioning is less obvious.

The R_o discussion of the previous section suggests that doubling the modulation set is roughly sufficient to optimize the error exponent for the random ensemble of codes, and we use this as a guideline. For example, if we seek to efficiently encode R = 4 bits/interval, we should consider the 32-ary 4-D constellation as a signal set. The bandwidth would be the same as uncoded 16-ary in 4-D, but with energy gain dependent on trellis complexity. It may be that use of a 48-ary or 64-ary base, provides better performance due to special features of set partitioning. The use of the larger alphabet does not substantially complicate the modulator/demodulator beyond a 32-ary . se; receiver complexity is largely determined by the trellis size.

We have made preliminary investigations of code design for small (less than 4-state) trellis codes having R = 2, 3, and 4 b s/interval, and begin with the simplest case to illustrate.

Suppose we seek a 2-state code with R = 2 bit/interval. The trellis diagram is shown in Figure 7a, with 4 branches per state. We consider assigning symbols from an 8-ary set to the eight branches as labelled. Now consider pairs of sequences which split at time n = 0 and remerge at some later time. The one-step merges have $d_1^2 = 4$ because of antipodality. The two step-merges, of which there are several types, also have $d_2^2 = 4$ since two units of d^2 accrue on each interval. The average energy expended per interval, $\bar{E}$, is 1. Thus $d_{min}^2 = 4\bar{E}$ and asymptotically

$$P[\varepsilon] \sim N\,Q\left(\left(\frac{d_{min}^2}{2N_o}\right)^{1/2}\right) = N\,Q\left(\left(\frac{2\bar{E}}{N_o}\right)^{1/2}\right) \qquad (14)$$

$$= N\,Q\left(\left(\frac{4\bar{E}_b}{N_o}\right)^{1/2}\right)$$

Now to evaluate this design, we can compare with an uncoded means of transmitting 2 bits/interval in 4-D. Though not the best way, we could use binary PSK on each polarization, or QPSK on a single polarization. Each has

$$P[\varepsilon] \sim N\,Q\left(\left(2\bar{E}_b/N_o\right)^{1/2}\right) \qquad (15)$$

showing a 3 dB gain for the coded case, with no change in bandwidth.

ORIGINAL PAGE IS
OF POOR QUALITY

Another comparison is against uncoded 8-ary, recognizing the proposition that the modulation set used for coding could transmit 3 bits per interval rather than 2. From section 4

$$P[\varepsilon] \sim N\, Q\left(\left(\frac{3E_b}{N_o}\right)^{1/2}\right) \tag{16}$$

with a spectral efficiency of 3 bps/Hz. Relative to this case, the coded design gains 10 log (4/3) = 1.2 dB in return for a 50% increase in bandwidth. Viewed in this light, the 2-state coding design is not very attractive relative to uncoded 8-ary signalling.

Now consider use of a 4-state trellis as shown in Figure 7b. With the same rate, R = 2, we have the option of splitting the four branches per state into 4-sets-of-1 or 2-sets-of-2. The latter doesn't buy any gain over 2-state because the one-step merges still are possible and have $d_1^2 = 4$. Thus only the 4-by-1 strategy has potential for improvement. It turns out however that no assignment of the 8 signals to these 16 branches can improve the 2-step distance beyond 4.

Next, suppose we allow use of a 16-ary modulation, via the hypercube vertices. We may conveniently carve this set into 4 sets of 4 as listed in Figure 7b. The intraset squared distance is at least 8, while the interset distance is at least 4. By assigning sets as shown to the trellis, the 2-step squared distance is now $d^2 = 12$, but recalling $\bar{E} = 4$, $d_{min}^2 = 3\bar{E} = 6\bar{E}_b$ and

$$P[\varepsilon] \sim N\, Q\left(\left(\frac{3\bar{E}_b}{N_o}\right)^{1/2}\right) \tag{17}$$

Thus performance is actually worse (asymptotically) than the earlier code, pointing out the subtle interactions of trellis structure, coding rate, and modulation set.

Hand calculations show that 8-state codes do gain over the 2-state case, but further optimization is required for these larger codes.

Next consider the R = 3 case, with 2 states to begin. The trellis is illustrated in Figure 8a. As a first cut, use the 16-ary set formed by the hypercube and divide into 4 sets of 4 as in Figure 7b. The one step merge distance is 8, while the two-step distance is at least 12. Thus $d^2_{min} = 2\bar{E} = 6\bar{E}_b$ since we have 3 bits/interval. Asymptotically

$$P[\varepsilon] \sim N\, Q\left(\left(3E_b/N_o\right)^{1/2}\right) \tag{18}$$

1.8 dB better than the QPSK with reuse strategy. Unfortunately, this energy efficiency is the same as for uncoded 8-ary with exactly the same bandwidth. Thus the 2-state code presented is of no practical use.

As a next case, assume a 4-state trellis with R = 3 and use the hypercube set as before, except split the 16 signals into 8 sets of antipodal pairs, e.g. 1111 and -1-1-1-1. The one-step squared-distance is now 16, while the two-step merges are at least distance 12. Thus $d^2_{min} = 3\bar{E} = 9\bar{E}_b$ and

$$P[\varepsilon] = N\, Q\left(\left(4.5\ E_b/N_o\right)^{1/2}\right)$$

Compared to QPSK with reuse, or uncoded 16-ary, we have a gain of 10 log (4.5/2) = 3.6 dB with a bandwidth which is 33% greater. We may also compare at the same bandwidth with uncoded 8-ary: the coded 16-ary case has a gain of 10 log (4.5/3) = 1.8 dB. This code is relatively easily decoded, since pairs of paths entering each state are antipodal; once selection between these is made, the receiver must arbitrate between the remaining four paths. We also note that since the modulation is QPSK/reuse, the modem equipment is rather simple.

The 8-state extension (not shown) of this case has a $d^2_{min} = 4\bar{E}$, yielding a 4.8 dB gain over uncoded 16-ary, again with a 33% bandwidth expansion.

We finally address R = 4 bits/symbol coding. We begin with a 2-state case, and 32-ary modulation. We may split the 32-ary set into 4 sets of 8 as shown in Figure 9a. The intraset d^2 is 6 and the interset d^2 is 2, so that $d^2_{min} = 4$. Since $\bar{E} = 3$, $d^2 = (4/3)\bar{E} = (16/3)\bar{E}_b$, and

$$P[\varepsilon] \sim N\,Q\left(\left(\frac{2.67\;E_b}{N_o}\right)^{1/2}\right) \tag{19}$$

This represents a 1.3 dB gain over 16-ary with the same bandwidth.

If the sets are further partitioned into 8 sets of 4 and the trellis splits the 16 branches as 4 sets of 4, then a $d^2_{min} = 6$ can be attained with 4 states (Figure 9b). For this case

$$P[\varepsilon] \sim N\,Q\left(\left(\frac{4E_b}{N_o}\right)^{1/2}\right) \tag{20}$$

giving a 3 dB gain over uncoded 16-ary having the same bandwidth.

To summarize the code study thus far, it appears that coding is most beneficial in D_4 for higher throughput cases, e.g. R ≥ 3 bits/interval, relative to uncoded counterparts. Further investigations are presently being made to extend these results to (1) higher rates, e.g. R = 5 and 6 bits/interval, and (2) larger trellises.

7. CONCLUSION

Four-dimensional modulation provides a means of improving the power and/or bandwidth utilization of satellite channel, relative to a polarization reuse strategy. 4-D lattices are known to have superior packing density as a basis for signal design, and we have provided explicit constructions for 8, 16, 32, ... 1024 signals in 4-D. The most efficient are subsets of the lattice D_4, or translates thereof. Typically, about 1.5 to 3 dB gain may be had at equal bandwidth over a polarization reuse strategy, or for fixed power, about 25% less bandwidth may be consumed.

Trellis codes have been studied as a means of further extending the power/bandwidth tradeoff. Thus far codes for R = 3 and 4 bits/interval with four state or less have been shown to provide attractive gains relative to cases using polarization reuse.

We remark that the designs presented here in general require that amplifiers be utilized which are linear up to the maximum power required by the constellation. This seems unavoidable for attaining high spectral efficiency, although continuous-phase-modulation is an attractive alternative.

8. ACKNOWLEDGEMENTS

The authors wish to thank Daniel McGrady and Nina Srinath for assistance in preparation of this work.

ORIGINAL PAGE IS
OF POOR QUALITY

BIBLIOGRAPHY

1. D. Slepian, "Group Codes for the Gaussian Channel," BSTJ, 1960, pp. 575-602.

2. G. R. Welti and J. S. Lee, "Digital Transmission with Coherent Four-Dimensional Modulation," IEEE Trans. on Information Theory, July 1974.

3. L. Zetterberg and H. Brandstrom, "Codes for Combined Phase and Amplitude Modulated Signals in Four-Dimensional Space," IEEE Trans. on Communication, September 1977.

4. C. M. Thomas et al, "Digital Amplitude - Phase Keying with M-ary Alphabets," IEEE Trans. on Communications, February 1974.

5. G. J. Foschini, R. D. Gitlin and S. B. Weinstein, "Optimization of Two-Dimensional Signal Constellation in Presence of Gaussian Noise," IEEE Trans. of Communications, January 1974.

6. H. S. M. Coxeter, "The Problem of Packing ..... on the Sphere," Transactions, New York Academy of Science, Vol. 24, No. 3, January 1962.

7. J. H. Conway and N. J. A. Sloane, "Fast Quantizing and Decoding Algorithms for Lattice Quantizers and Codes," IEEE Trans. on Information Theory, March 1982.

8. J. H. Conway and N. J. A. Sloane, "Voronoi Regions of Lattices .....," IEEE Trans. on Information Theory, March 1982.

9. N.J.A. Sloane, "Tables of Sphere Packing and Spherical Codes," IEEE Trans. on Information Theory, May 1981.

10. H. Brandstrom, "Combined Amplitude and Phase Modulation based on Complex Polytopes," Tech. Report 123, Electrical Engineering, Royal Institute of Technology, Stockholm, December 1977.

11. J. L. Massey, "Coding and Modulation in Digital Communications," Proceedings Int'l Zurich Seminar on Dig. Communications, March 1974.

12. J. M. Wozencraft and I. M. Jacobs, Principles of Communication Engineering, Wiley, 1967.

13. G. Ungerboeck, "Channel Coding With Multilevel/Phase Signals," IEEE Trans. on Information Theory, January, 1982.

Table I

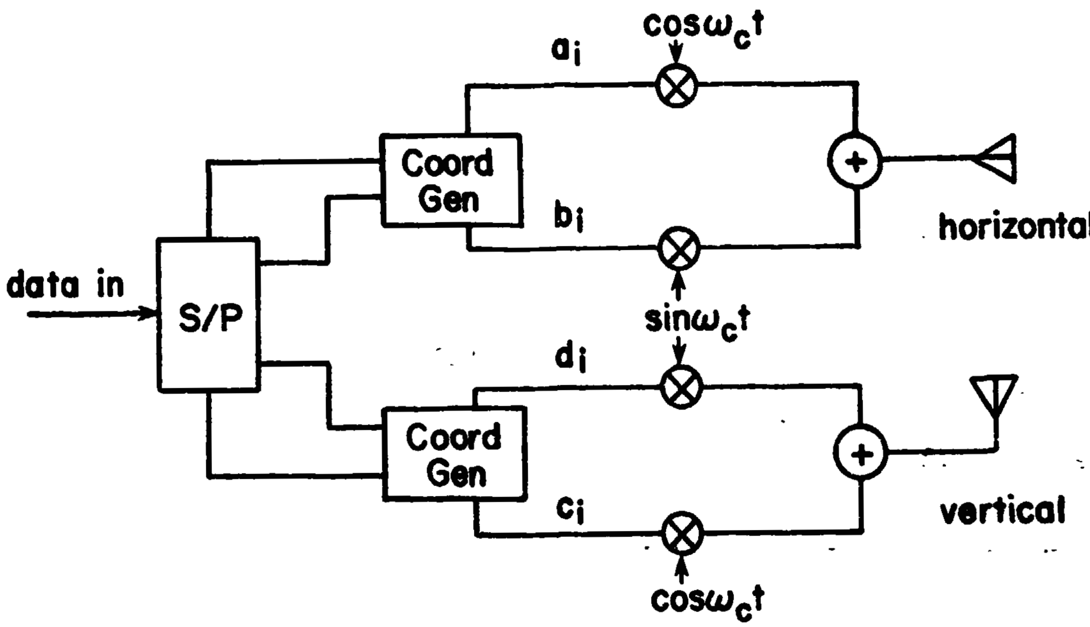
Parameters of Best Z_4 Designs

$\underline{M}$	$\bar{E}/d^2$	E_p/d^2	offset[1)]	Comments
8	0.75	0.75	(0.5,0.5,0.5,0)	full
16	1.0	1.00	(0.5,0.5,0.5,0.5)	single-shell
32	1.75	2.00	(0,0,0,0)	remove origin
64	2.37	3.00	(0,0,0,0)	remove origin
128	3.375	4.50	(0.5,0.5,0,0)	full
256	5.00	6.75	(0.5,0.5,0.5,0)	remove first shell
512	6.75	9.00	(0.5,0.5,0.5,0.5)	full
1024	10.83	14.75	(0.5,0.5,0.5,0)	remove first and fourth shells

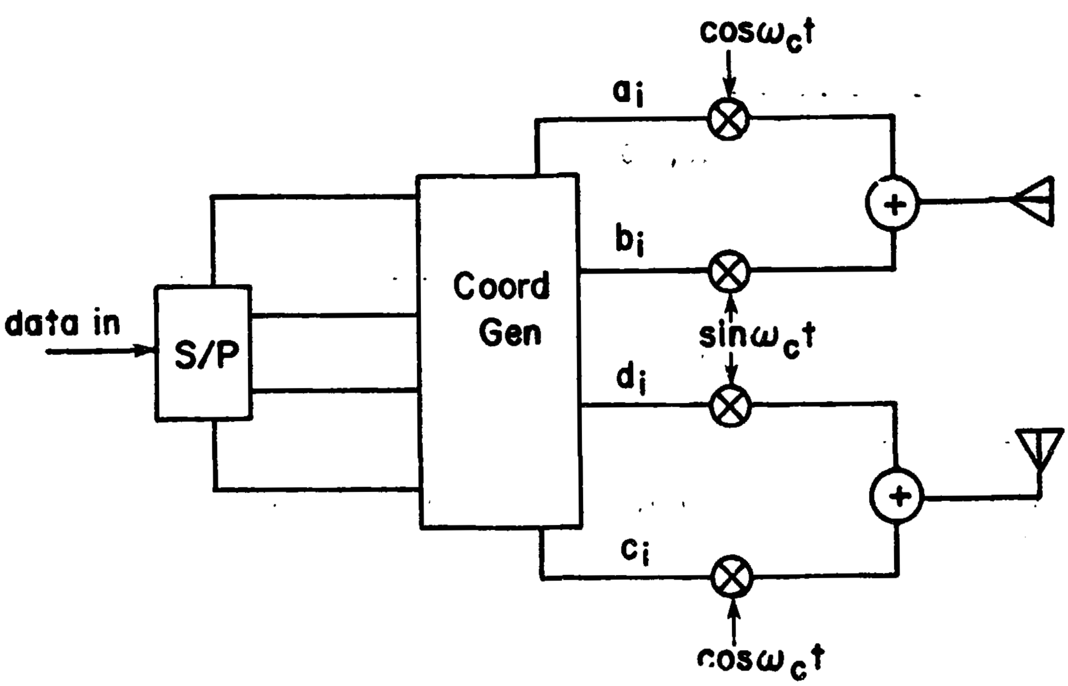
1) offset refers to translation of Z_4 points, where $Z_4 \stackrel{\Delta}{=} \{x_i,\ i=1,4 \mid x_i \text{ an integer}\}$

Table II. Parameters of Best D_4 Designs

M	$\bar{E}/d^2$	E_p/d^2	Offset	Comment
8	0.5	0.5	(1,0,0,0)	($\pm$1,0,0,0), biorthogonal
16	1.0	1.0	(0,0,0,0)	($\pm$1,$\pm$1,$\pm$1,$\pm$1), 4-D cube
	0.88	1.25	(0.5,0.5,0,0)	first two shells plus 6 from 3rd
	0.88	1.37	(0,0.5,0.5,0.5)	first two shells plus 4 from 3rd
32	1.25	2.0	(0,0,0,0)	{$\pm$1,$\pm$1,0,0} U {$\pm$2,0,0,0}
	1.50	1.50	(1,0,0,0)	{+1,+1,+1,0}
	1.25	1.50	(1,0,0,0)	{$\pm$1,0,0,0} U 24 of {$\pm$1,$\pm$1,$\pm$1,0}
64	1.69	2.25	(0.5,0.5,0,0)	first five full shells
128	2.44	3.0	(0,0,0,0)	first 3 shells plus 80 from 4th shell
256	3.375	4.50	(1,0,0,0)	first five full shells
512	4.84	7.50	(1,0,0,0)	first seven shells plus 48 of eighth
1024	6.81	10.5	(1,0,0,0)	first nine shells plus {$\pm$3,$\pm$2,$\pm$2,$\pm$2}

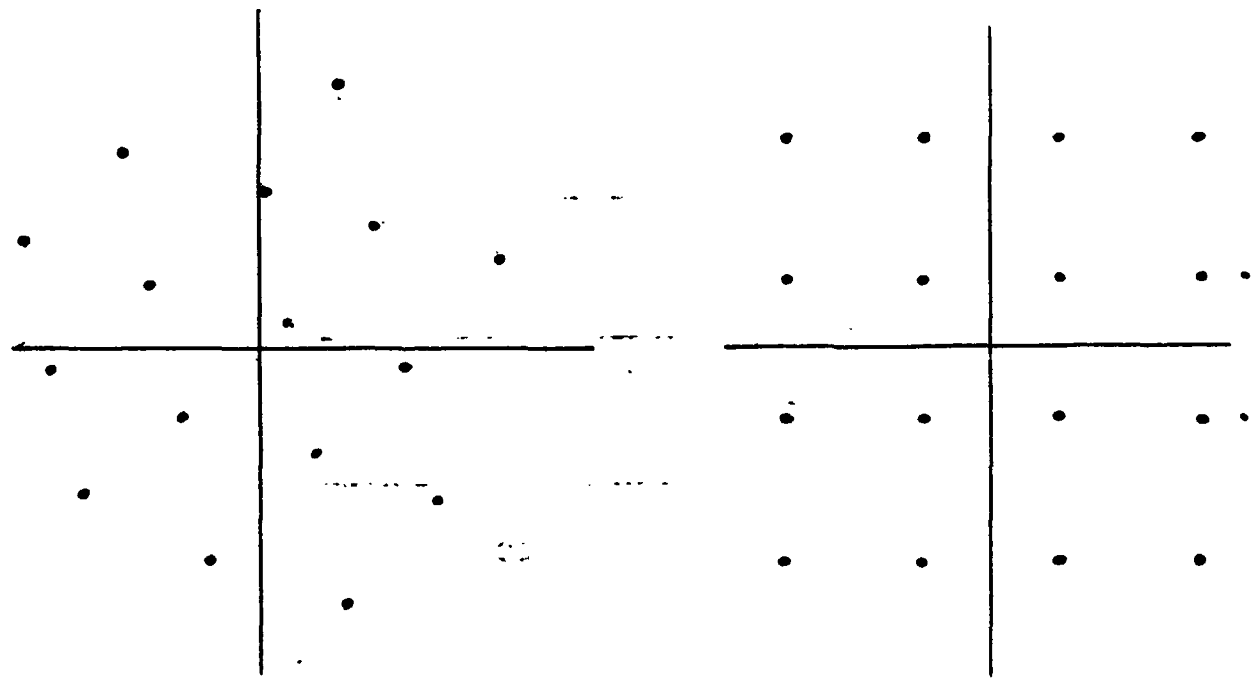


2-D MODULATION WITH REUSE



4-D MODULATION

Figure 1 -- Depiction of Modulators for 2-D/Polarization Reuse and 4-D Modulation



2a. Optimal 16-ary Design on 2-D

2b. Standard 16-QASK Design

Figure 2. 16-ary Constellations in Two-Dimensions

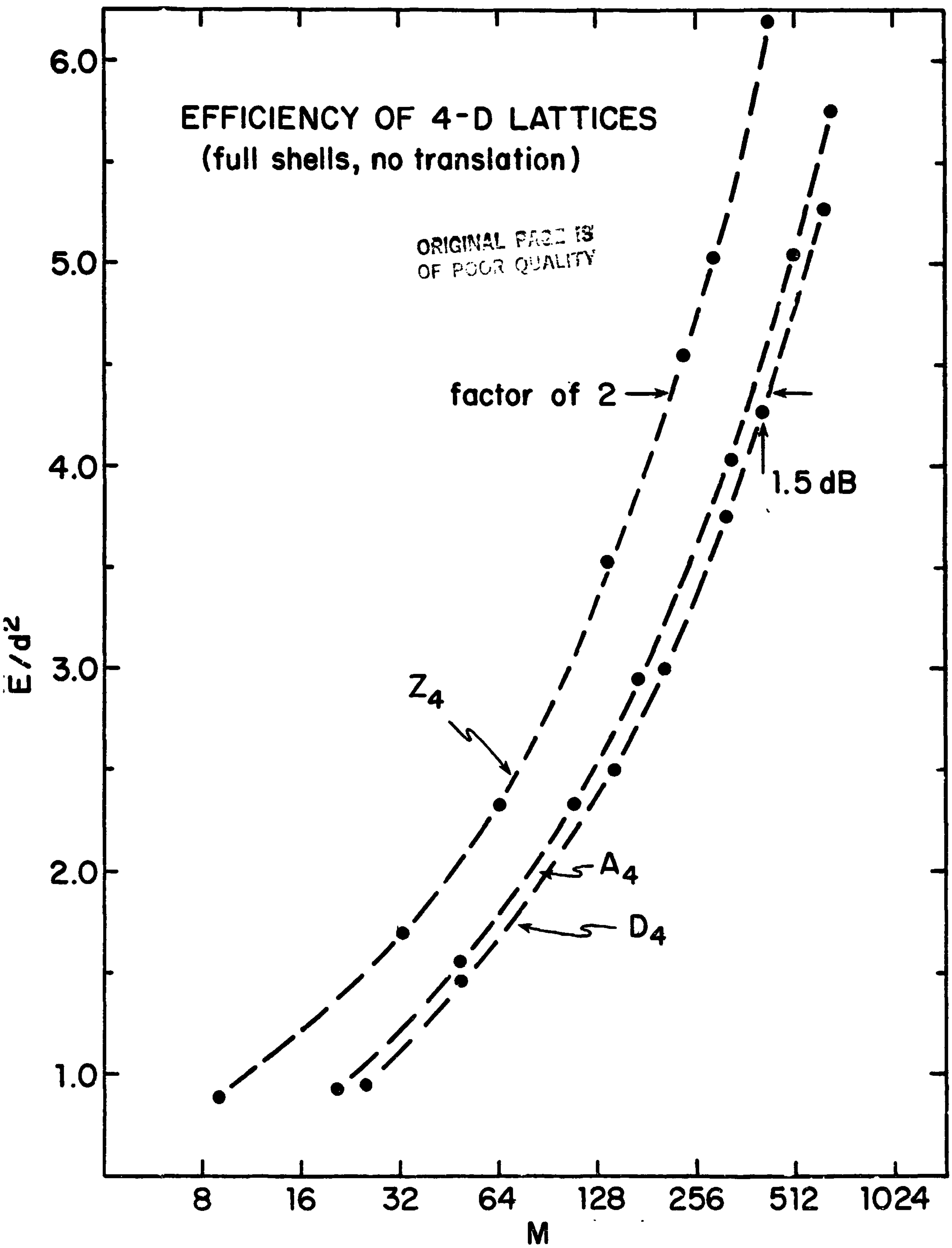


Figure 3. Packing Efficiency of 4-D Lattices Z_4, A_4 and D_4. Points correspond to full-shell constellations.

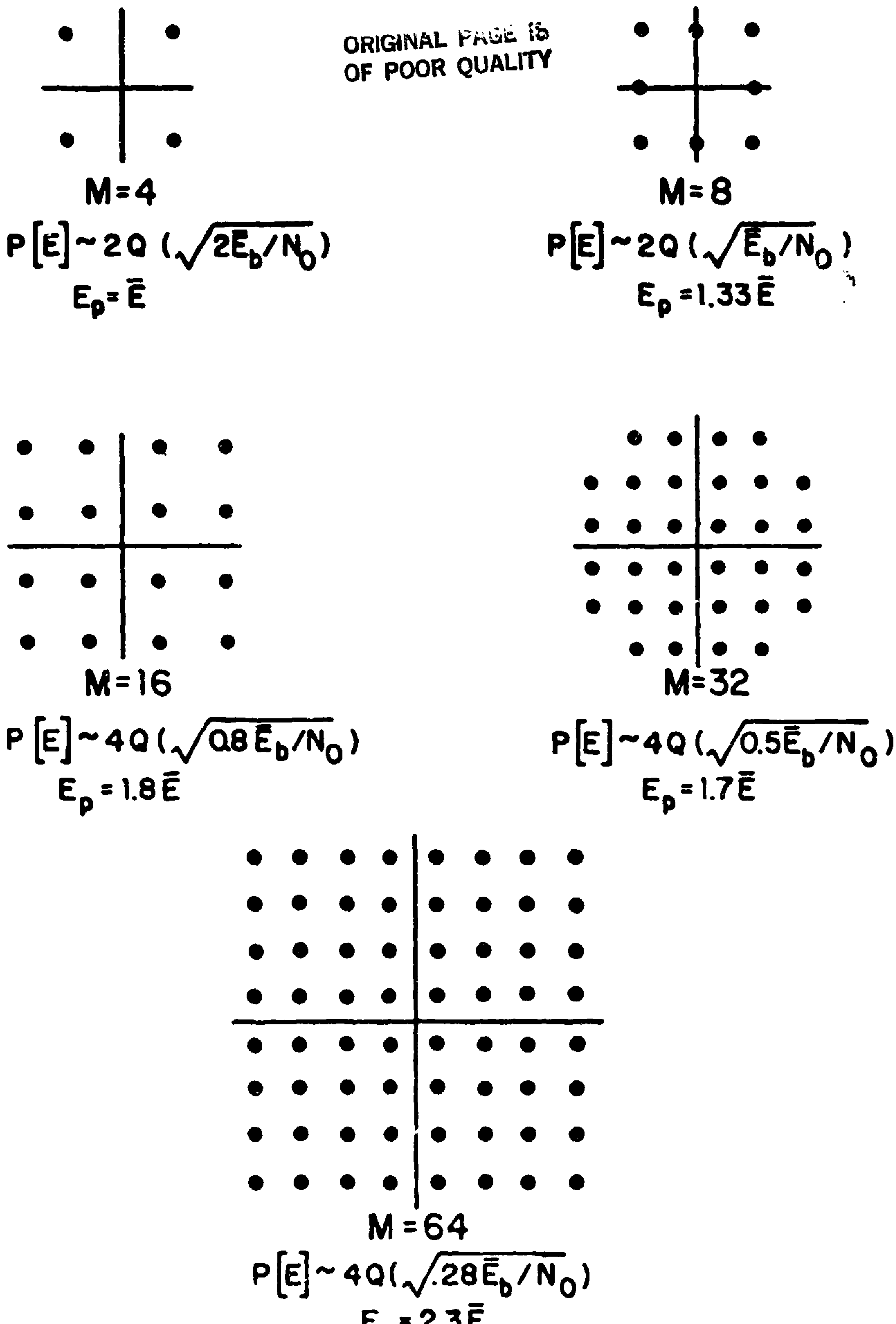


Figure 4. RECTANGULAR 2-D CONSTELLATIONS

ORIGINAL PAGE IS
OF POOR QUALITY

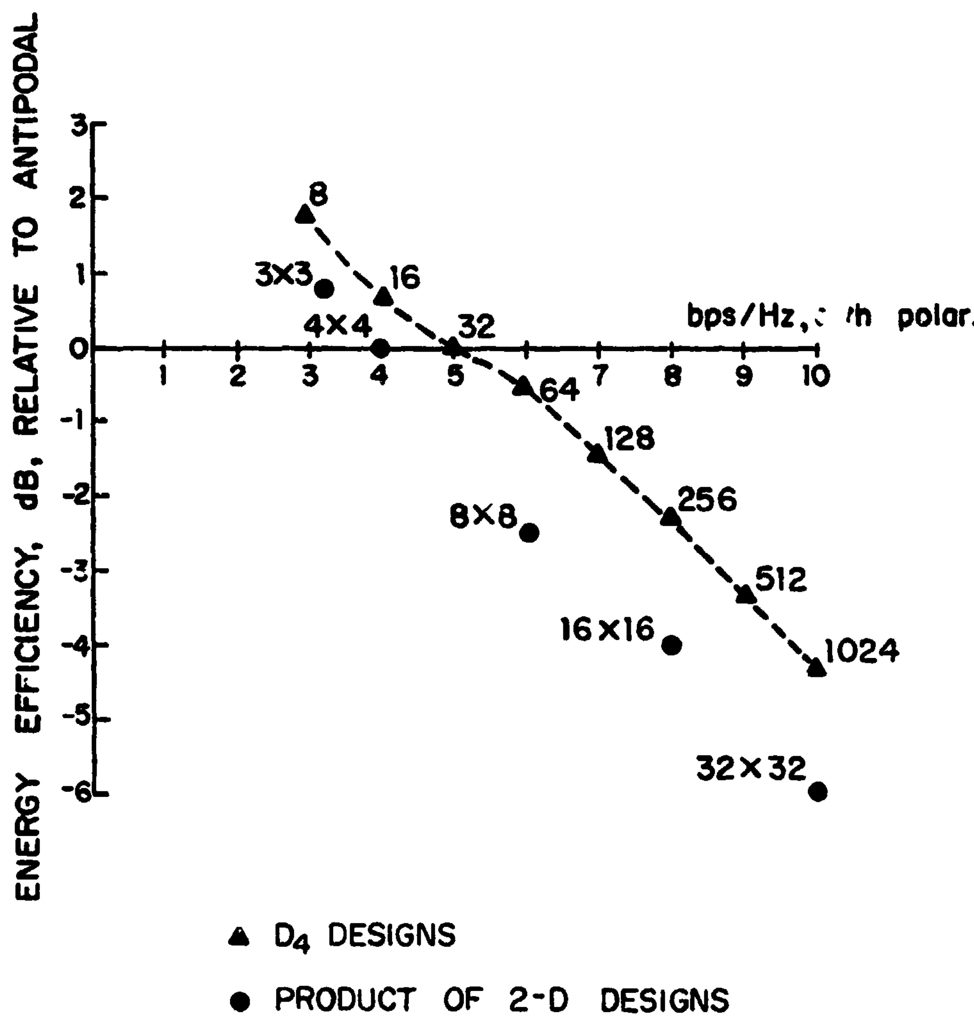


Figure 5. Engery-versus-Bandwidth Efficiency of 4-D and Product-of-2-D Designs. Bandwidth is Nyquist Bandwidth.

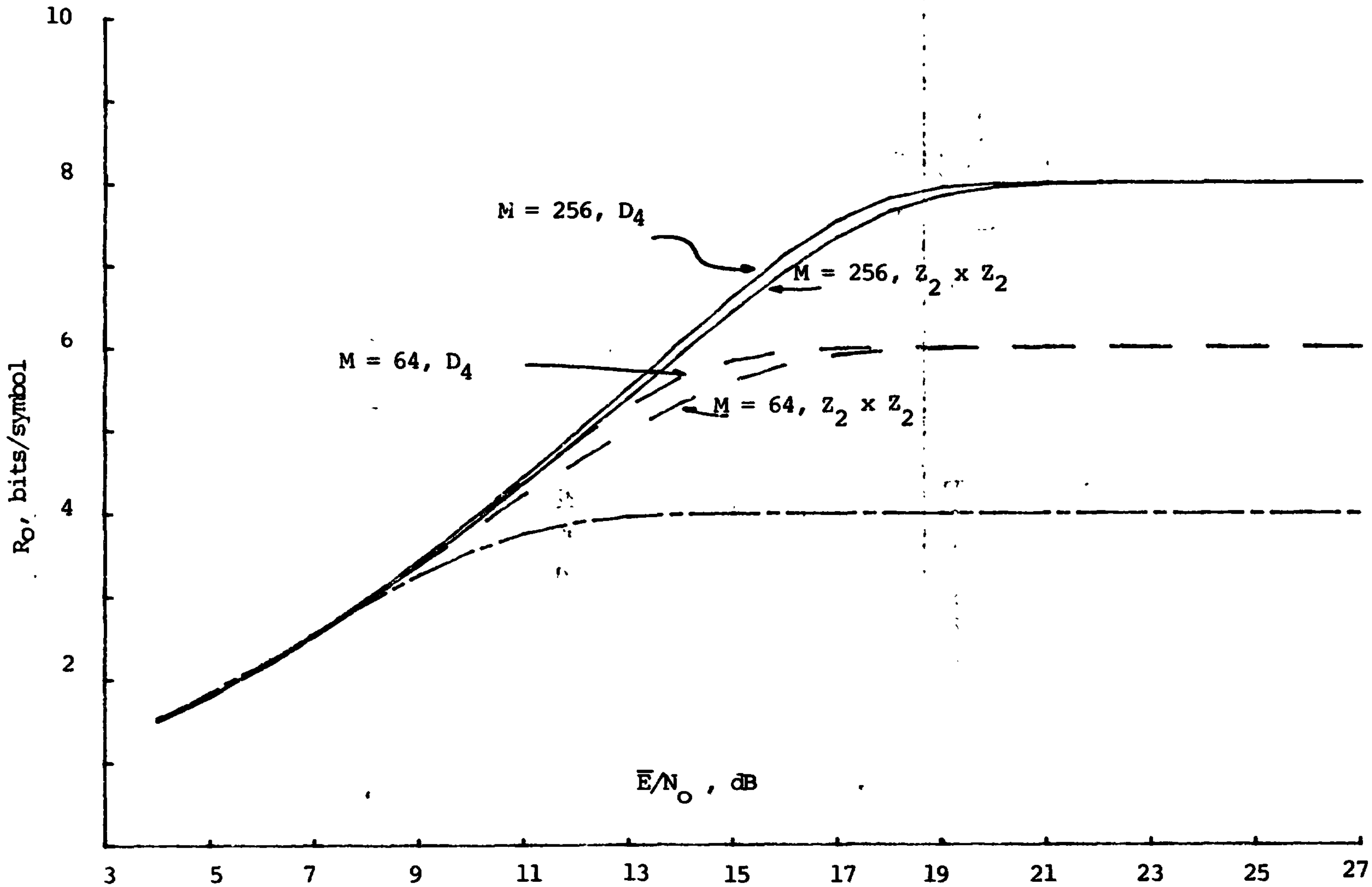
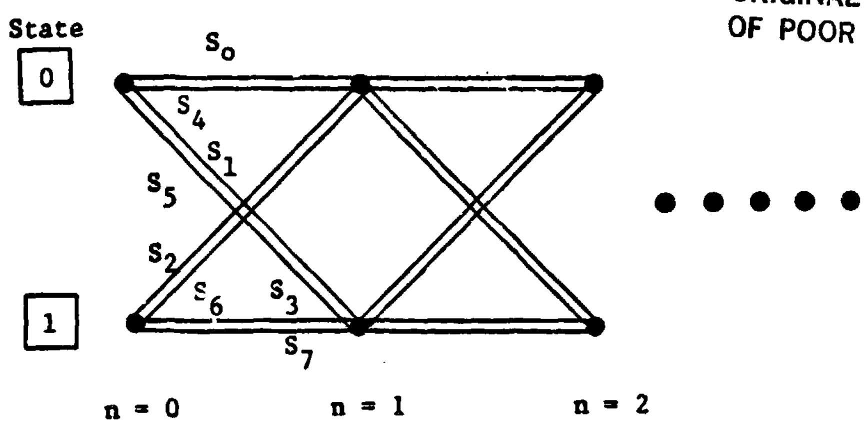
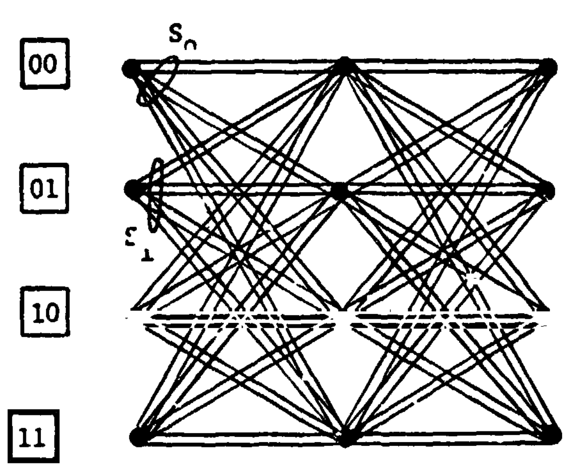


Figure 6. R_o Versus $\overline{E}/N_o$ for Selected 4-D and Product of 2-D Modulations



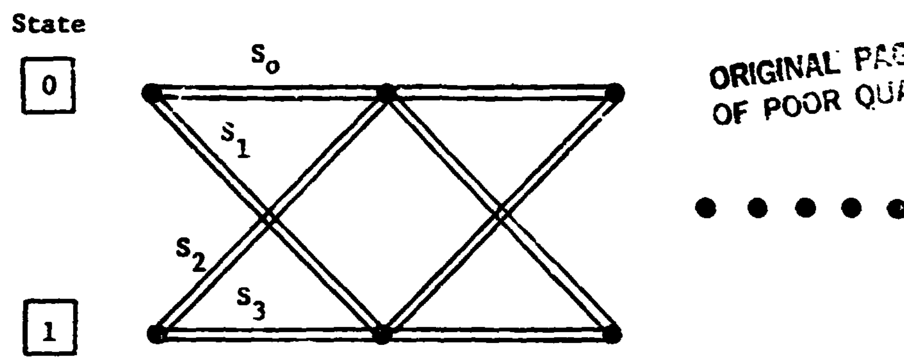
$$S_0 = 1\ 0\ 0\ 0 = -S_4$$
$$S_1 = 0\ 1\ 0\ 0 = -S_5$$
$$S_2 = 0\ 0\ 1\ 0 = -S_6$$
$$S_3 = 0\ 0\ 0\ 1 = -S_7$$

Figure 7a. 2-State Trellis for R = 2, 8-ary Modulation



$$S_0 = \begin{matrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ -1 & -1 & 1 & 1 \\ -1 & -1 & -1 & -1 \end{matrix} \quad S_1 = \begin{matrix} 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \\ -1 & 1 & 1 & -1 \\ -1 & 1 & -1 & 1 \end{matrix} \quad S_2 = \begin{matrix} 1 & 1 & 1 & -1 \\ 1 & 1 & -1 & 1 \\ -1 & -1 & 1 & -1 \\ -1 & -1 & -1 & 1 \end{matrix} \quad S_3 = \begin{matrix} -1 & 1 & 1 & -1 \\ -1 & 1 & 1 & 1 \\ 1 & -1 & -1 & -1 \\ 1 & -1 & 1 & 1 \end{matrix}$$

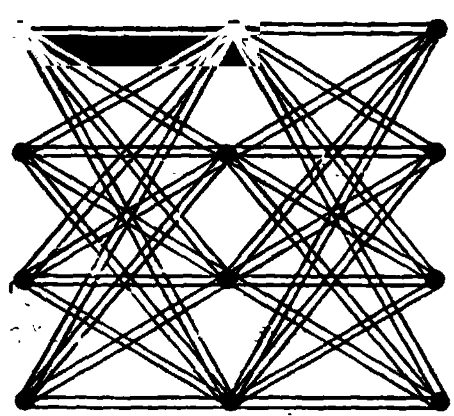
Figure 7b. 4-State Trellis for R = 2, 16-ary Modulation



$$S_o = \begin{matrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ -1 & -1 & 1 & 1 \\ -1 & -1 & -1 & -1 \end{matrix} \qquad S_1 = \begin{matrix} 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \\ -1 & 1 & 1 & -1 \\ -1 & 1 & -1 & 1 \end{matrix} \qquad S_2 = \begin{matrix} 1 & 1 & 1 & -1 \\ 1 & 1 & -1 & 1 \\ -1 & -1 & 1 & -1 \\ -1 & -1 & -1 & 1 \end{matrix}$$

$$S_3 = \begin{matrix} -1 & 1 & \text{[illegible]} & -1 \\ -1 & 1 & 1 & 1 \\ 1 & -1 & -1 & -1 \\ 1 & -1 & 1 & 1 \end{matrix}$$

Figure 8a. 2-State Trellis for R = 3 with 16-ary Modulation

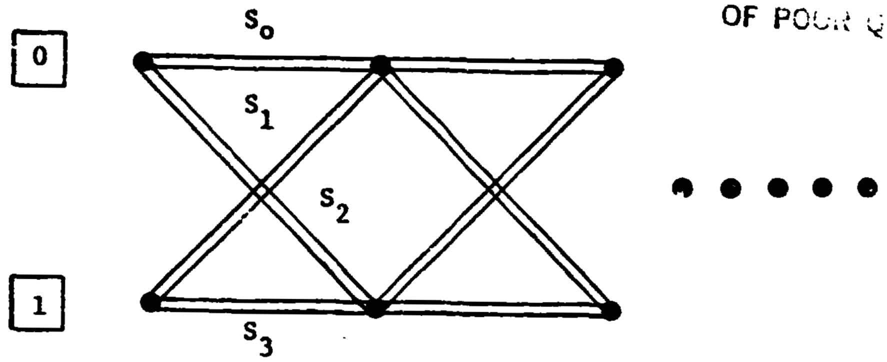


$$C_o = \begin{matrix} 1 & 1 & 1 & 1 \\ -1 & -1 & -1 & -1 \end{matrix} \qquad C_1 = \begin{matrix} -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 \end{matrix}$$

Split of S_o above

$C_3 \ldots C_7$ are splits of $S_1 \ldots S_3$ above

Figure 8b. 4-State Trellis for R = 3 with 16-ary Modulation

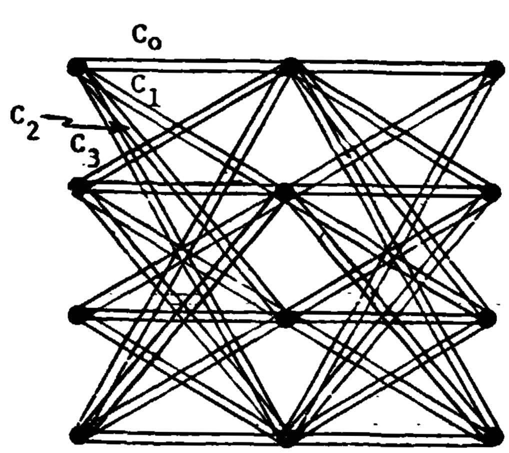


S_o:

1	1	1	0
-1	-1	-1	0
0	-1	1	1
0	1	-1	-1
1	0	-1	1
-1	0	1	-1
1	-1	0	-1
-1	1	0	1

S_1, S_2, S_3: cycle position of 0 relative to S_o

Figure 9a. 2-State Trellis for R = 4 with 32-ary Modulation



C_o:

1	1	1	0
0	-1	1	1
1	0	-1	1
1	-1	0	-1

C_1:

-1	-1	-1	0
0	1	-1	-1
-1	0	1	-1
-1	1	0	1

Splitting of S_o above

$C_2 \ldots C_7$ are splits of $S_1 \ldots S_3$ above

Figure 9b. 4-State Trellis for R = 4 with 32-ary Modulation

APPENDIX A

TABULATION OF Z_4 LATTICE WITH DIFFERENT OFFSETS

.RUN Z4LAT
ANALYSIS OF Z4 LATTICE
ENTER DATA
0,0,0,0
ALL DATA IS IN

$d^2_{lattice} = 1$

	COMBINATIONS	SHELL NORM
1	1	0.00
2	8	1.00 ←
3	24	2.00
4	32	3.00 ←
5	24	4.00
6	48	5.00
7	96	6.00
8	64	7.00 ←
9	24	8.00
10	104	9.00
11	144	10.00
12	96	11.00
13	96	12.00
14	112	13.00
15	192	14.00
16	192	15.00
17	24	16.00
18	144	17.00
19	312	18.00
20	160	19.00
21	144	20.00
22	256	21.00 ←
23	288	22.00
24	192	23.00
25	96	24.00
26	240	25.00
27	288	26.00
28	224	27.00
29	128	28.00
30	192	29.00
31	384	30.00
32	64	31.00
33	24	32.00
34	288	33.00
35	192	34.00
36	192	35.00
37	112	36.00
38	192	37.00
39	192	38.00
40	96	40.00
41	96	41.00
42	192	42.00
43	64	43.00
44	192	45.00
45	32	48.00
46	64	49.00
47	96	50.00
48	64	52.00
49	64	57.00
50	16	64.00

	FSUM	EAVE	EPEAK	
1	1.000	0.000	0.000	
2	9.000	0.889	1.000	
3	33.000	1.697	2.000	← 32 if origin removed
4	65.000	2.338	3.000	← 64 if origin removed
5	89.000	2.787	4.000	
6	137.000	3.562	5.000	← 128 if 1st & 2nd shells removed
7	233.000	4.567	6.000	
8	297.000	5.091	7.000	
9	321.000	5.308	8.000	
10	425.000	6.212	9.000	
11	569.000	7.170	10.000	
12	665.000	7.723	11.000	←
13	761.000	8.263	12.000	
14	873.000	8.871	13.000	
15	1065.000	9.795	14.000	←
16	1257.000	10.590	15.000	
17	1281.000	10.692	16.000	
18	1425.000	11.329	17.000	
19	1737.000	12.527	18.000	
20	1897.000	13.073	19.000	
21	2041.000	13.562	20.000	
22	2297.000	14.391	21.000	
23	2585.000	15.239	22.000	
24	2777.000	15.775	23.000	
25	2873.000	16.050	24.000	
26	3113.000	16.740	25.000	
27	3401.000	17.524	26.000	
28	3625.000	18.110	27.000	
29	3753.000	18.447	28.000	
30	3945.000	18.961	29.000	
31	4329.000	19.940	30.000	
32	4393.000	20.101	31.000	
33	4417.000	20.166	32.000	
34	4705.000	20.951	33.000	
35	4897.000	21.463	34.000	
36	5089.000	21.974	35.000	
37	5201.000	22.276	36.000	
38	5393.000	22.800	37.000	
39	5585.000	23.322	38.000	
40	5681.000	23.604	40.000	
41	5777.000	23.893	41.000	
42	5969.000	24.476	42.000	
43	6033.000	24.672	43.000	
44	6225.000	25.299	45.000	
45	6257.000	25.415	48.000	
46	6321.000	25.654	49.000	
47	6417.000	26.018	50.000	
48	6481.000	26.275	52.000	
49	6545.000	26.575	57.000	
50	6561.000	26.667	64.000	

STOP --

.RUN Z4LAT
ANALYSIS OF Z4 LATTICE
ENTER DATA
0.5,0,0,0
ALL DATA IS IN

$d_{lattice} = 1$

	COMBINATIONS	SHELL NORM
1	0	0.00
2	2	0.25
3	12	1.25
4	26	2.25
5	28	3.25
6	36	4.25
7	64	5.25 ←
8	62	6.25
9	60	7.25
10	96	8.25
11	76	9.25
12	84	10.25
13	156	11.25
14	114	12.25
15	108	13.25
16	160	14.25
17	124	15.25
18	168	16.25
19	192	17.25
20	148	18.25
21	192	19.25
22	241	20.25
23	210	21.25
24	168	22.25
25	248	23.25
26	190	24.25
27	168	25.25
28	312	26.25
29	160	27.25
30	168	28.25
31	238	29.25
32	144	30.25
33	216	31.25
34	176	32.25
35	184	33.25
36	168	34.25
37	144	35.25
38	150	36.25
39	96	37.25
40	204	38.25
41	88	39.25
42	72	40.25
43	192	41.25
44	72	42.25
45	48	43.25
46	48	44.25
47	72	45.25
48	96	46.25
49	56	47.25
50	64	48.25
51	48	49.25
52	16	50.25
53	12	52.25
54	72	53.25
55	40	54.25
56	24	56.25
57	16	60.25
58	24	61.25
59	8	68.25

	FSUM	EAVE	EPEAK
1	0.000	0.000	0.000
2	2.000	0.250	0.250
3	14.000	1.107	1.250
4	40.000	1.850	2.250
5	68.000	2.426	3.250
6	104.000	3.058	4.250
7	168.000	3.893	5.250
8	230.000	4.528	6.250
9	290.000	5.091	7.250
10	386.000	5.877	8.250
11	462.000	6.432	9.250
12	546.000	7.019	10.250
13	702.000	7.959	11.250
14	816.000	8.559	12.250
15	924.000	9.107	13.250
16	1084.000	9.866	14.250
17	1208.000	10.419	15.250
18	1376.000	11.131	16.250
19	1568.000	11.880	17.250
20	1716.000	12.429	18.250
21	1908.000	13.116	19.250
22	2149.000	13.916	20.250
23	2359.000	14.569	21.250
24	2527.000	15.079	22.250
25	2775.000	15.810	23.250
26	2965.000	16.351	24.250
27	3133.000	16.828	25.250
28	3445.000	17.681	26.250
29	3605.000	18.106	27.250
30	3773.000	18.557	28.250
31	4011.000	19.192	29.250
32	4155.000	19.575	30.250
33	4371.000	20.152	31.250
34	4547.000	20.620	32.250
35	4731.000	21.112	33.250
36	4899.000	21.562	34.250
37	5043.000	21.953	35.250
38	5193.000	22.366	36.250
39	5289.000	22.636	37.250
40	5493.000	23.216	38.250
41	5581.000	23.469	39.250
42	5653.000	23.683	40.250
43	5845.000	24.260	41.250
44	5917.000	24.478	42.250
45	5965.000	24.630	43.250
46	6013.000	24.786	44.250
47	6085.000	25.028	45.250
48	6181.000	25.358	46.250
49	6237.000	25.554	47.250
50	6301.000	25.785	48.250
51	6349.000	25.962	49.250
52	6365.000	26.023	50.250
53	6377.000	26.073	52.250
54	6449.000	26.376	53.250
55	6489.000	26.548	54.250
56	6513.000	26.657	56.250
57	6529.000	26.740	60.250
58	6553.000	26.866	61.250
59	6561.000	26.917	68.250

STOP --

.RUN Z4LAT
ANALYSIS OF Z4 LATTICE
ENTER DATA
0.5,0.5,0,0

$d^2_{lattice} = 1$

ALL DATA IS IN

	COMBINATIONS	SHELL NORM
1	0	0.00
2	4	0.50
3	16	1.50
4	24	2.50
5	32	3.50
6	52	4.50
7	48	5.50
8	56	6.50
9	96	7.50
10	72	8.50
11	80	9.50
12	128	10.50
13	96	11.50
14	124	12.50
15	160	13.50
16	120	14.50
17	128	15.50
18	192	16.50
19	192	17.50
20	152	18.50
21	224	19.50
22	164	20.50
23	160	21.50
24	292	22.50
25	176	23.50
26	196	24.50
27	240	25.50
28	164	26.50
29	208	27.50
30	224	28.50
31	176	29.50
32	144	30.50
33	240	31.50
34	196	32.50
35	128	33.50
36	192	34.50
37	112	35.50
38	112	36.50
39	192	37.50
40	128	38.50
41	128	39.50
42	113	40.50
43	52	41.50
44	100	42.50
45	96	43.50
46	100	44.50
47	72	45.50
48	64	46.50
49	32	47.50
50	20	48.50
51	68	49.50
52	56	50.50
53	32	51.50
54	48	52.50
55	8	53.50
56	16	54.50
57	20	56.50
58	40	57.50
59	20	58.50
60	8	60.50
61	16	64.50
62	8	65.50
63	4	72.50

	FSUM	EAVE	EPEAK
1	0.000	0.000	0.000
2	4.000	0.500	0.500
3	20.000	1.300	1.500
4	44.000	1.955	2.500
5	76.000	2.605	3.500
6	128.000	3.375	4.500 ←
7	176.000	3.955	5.500
8	232.000	4.569	6.500
9	328.000	5.427	7.500
10	400.000	5.980	8.500
11	480.000	6.567	9.500
12	608.000	7.395	10.500
13	704.000	7.955	11.500
14	828.000	8.635	12.500
15	988.000	9.423	13.500
16	1108.000	9.973	14.500
17	1236.000	10.545	15.500
18	1428.000	11.346	16.500
19	1620.000	12.075	17.500
20	1772.000	12.626	18.500
21	1996.000	13.398	19.500
22	2160.000	13.937	20.500
23	2320.000	14.459	21.500
24	2612.000	15.358	22.500
25	2788.000	15.872	23.500
26	2984.000	16.438	24.500
27	3224.000	17.113	25.500
28	3388.000	17.567	26.500
29	3596.000	18.142	27.500
30	3820.000	18.749	28.500
31	3996.000	19.223	29.500
32	4140.000	19.615	30.500
33	4380.000	20.266	31.500
34	4576.000	20.790	32.500
35	4704.000	21.136	33.500
36	4896.000	21.660	34.500
37	5008.000	21.970	35.500
38	5120.000	22.288	36.500
39	5312.000	22.837	37.500
40	5440.000	23.206	38.500
41	5568.000	23.580	39.500
42	5681.000	23.917	40.500
43	5733.000	24.076	41.500
44	5833.000	24.392	42.500
45	5929.000	24.702	43.500
46	6029.000	25.030	44.500
47	6101.000	25.272	45.500
48	6165.000	25.492	46.500
49	6197.000	25.606	47.500
50	6217.000	25.679	48.500
51	6285.000	25.937	49.500
52	6341.000	26.154	50.500
53	6373.000	26.281	51.500
54	6421.000	26.477	52.500
55	6429.000	26.511	53.500
56	6445.000	26.580	54.500
57	6465.000	26.673	56.500
58	6505.000	26.862	57.500
59	6525.000	26.959	58.500
60	6533.000	27.001	60.500
61	6549.000	27.092	64.500
62	6557.000	27.139	65.500
63	6561.000	27.167	72.500

STOP --

```
.RUN Z4LAT
ANALYSIS OF Z4 LATTICE
ENTER DATA
0.5,0.5,0.5,0
ALL DATA IS IN
```

	COMBINATIONS	SHELL NORM
1	0	0.00
2	8	0.75
3	16	1.75
4	24	2.75
5	48	3.75
6	40	4.75
7	48	5.75
8	80	6.75
9	64	7.75
10	96	8.75
11	112	9.75
12	88	10.75
13	96	11.75
14	144	12.75
15	144	13.75
16	120	14.75
17	208	15.75
18	136	16.75
19	144	17.75
20	248	18.75
21	160	19.75
22	156	20.75
23	216	21.75
24	200	22.75
25	192	23.75
26	276	24.75
27	168	25.75
28	144	26.75
29	208	27.75
30	240	28.75
31	168	29.75
32	264	30.75
33	96	31.75
34	132	32.75
35	192	33.75
36	184	34.75
37	144	35.75
38	200	36.75
39	64	37.75
40	120	38.75
41	96	39.75
42	142	40.75
43	84	41.75
44	150	42.75
45	60	43.75
46	72	44.75
47	40	45.75
48	66	46.75
49	60	47.75
50	96	48.75
51	12	49.75
52	60	50.75
53	12	51.75
54	22	52.75
55	36	53.75
56	48	54.75
57	12	55.75
58	24	56.75
59	12	58.75

60	25	60.75
61	14	61.75
62	12	62.75
63	2	64.75
64	12	68.75
65	2	69.75
66	2	76.75

	FSUM	EAVE	EPEAK
1	0.000	0.000	0.000
2	8.000	0.750	0.750
3	24.000	1.417	1.750
4	48.000	2.083	2.750
5	96.000	2.917	3.750
6	136.000	3.456	4.750
7	184.000	4.054	5.750
8	264.000	4.871	6.750
9	328.000	5.433	7.750
10	424.000	6.184	8.750
11	536.000	6.929	9.750
12	624.000	7.468	10.750
13	720.000	8.039	11.750
14	864.000	8.924	12.750
15	1008.000	9.528	13.750
16	1128.000	10.083	14.750
17	1336.000	10.966	15.750
18	1472.000	11.500	16.750
19	1616.000	12.057	17.750
20	1864.000	12.947	18.750
21	2024.000	13.485	19.750
22	2180.000	14.005	20.750
23	2396.000	14.703	21.750
24	2596.000	15.323	22.750
25	2788.000	15.904	23.750
26	3064.000	16.700	24.750
27	3232.000	17.171	25.750
28	3376.000	17.579	26.750
29	3584.000	18.170	27.750
30	3824.000	18.834	28.750
31	3992.000	19.293	29.750
32	4256.000	20.004	30.750
33	4352.000	20.263	31.750
34	4484.000	20.630	32.750
35	4676.000	21.169	33.750
36	4860.000	21.683	34.750
37	5004.000	22.088	35.750
38	5204.000	22.652	36.750
39	5268.000	22.835	37.750
40	5388.000	23.189	38.750
41	5484.000	23.479	39.750
42	5626.000	23.915	40.750
43	5710.000	24.178	41.750
44	5860.000	24.653	42.750
45	5920.000	24.847	43.750
46	5992.000	25.086	44.750
47	6032.000	25.223	45.750
48	6098.000	25.456	46.750
49	6158.000	25.673	47.750
50	6254.000	26.027	48.750
51	6266.000	26.073	49.750
52	6326.000	26.307	50.750
53	6338.000	26.355	51.750
54	6360.000	26.446	52.750
55	6396.000	26.600	53.750
56	6444.000	26.810	54.750
57	6456.000	26.863	55.750
58	6480.000	26.974	56.750
59	6492.000	27.033	58.750
60	6517.000	27.162	60.750
61	6531.000	27.236	61.750
62	6543.000	27.301	62.750
63	6545.000	27.313	64.750
64	6557.000	27.389	68.750
65	6559.000	27.402	69.750
66	6561.000	27.417	76.750

STOP --

```
.RUN Z4LAT
ANALYSIS OF Z4 LATTICE
ENTER DATA
0.5,0.5,0.5,0.5
ALL DATA IS IN
```

	COMBINATIONS	SHELL NORM
1	0	0.00
2	16	1.00 ← hypercube
3	64	3.00 ←
4	96	5.00
5	128	7.00 ←
6	208	9.00
7	192	11.00
8	224	13.00
9	384	15.00
10	288	17.00
11	320	19.00
12	480	21.00
13	288	23.00
14	400	25.00
15	512	27.00 ←
16	288	29.00
17	352	31.00
18	384	33.00
19	288	35.00
20	256	37.00
21	288	39.00
22	216	41.00
23	112	43.00
24	216	45.00
25	144	47.00
26	64	49.00
27	96	51.00
28	72	53.00
29	48	55.00
30	32	57.00
31	48	59.00
32	8	61.00
33	8	63.00
34	24	65.00
35	8	67.00
36	8	73.00
37	1	81.00

	FSUM	EAVE	EPEAK
1	0.000	0.000	0.000
2	16.000	1.000	1.000 ←
3	80.000	2.600	3.000
4	176.000	3.909	5.000
5	304.000	5.211	7.000
6	512.000	6.750	9.000 ←
7	704.000	7.909	11.000
8	928.000	9.138	13.000
9	1312.000	10.854	15.000
10	1600.000	11.960	17.000
11	1920.000	13.133	19.000
12	2400.000	14.707	21.000
13	2688.000	15.595	23.000
14	3088.000	16.813	25.000
15	3600.000	18.262	27.000
16	3888.000	19.058	29.000
17	4240.000	20.049	31.000

ORIGINAL PAGE IS
OF POOR QUALITY

18	4624.000	21.125	33.000
19	4912.000	21.938	35.000
20	5168.000	22.684	37.000
21	5456.000	23.545	39.000
22	5672.000	24.210	41.000
23	5784.000	24.574	43.000
24	6000.000	25.309	45.000
25	6144.000	25.818	47.000
26	6208.000	26.057	49.000
27	6304.000	26.437	51.000
28	6376.000	26.737	53.000
29	6424.000	26.948	55.000
30	6456.000	27.097	57.000
31	6504.000	27.332	59.000
32	6512.000	27.373	61.000
33	6520.000	27.417	63.000
34	6544.000	27.555	65.000
35	6552.000	27.603	67.000
36	6560.000	27.659	73.000
37	6561.000	27.667	81.000

STOP --

APPENDIX B

TABULATION OF D_4 LATTICE WITH DIFFERENT OFFSETS

```
.RUN D4EVE
ANALYSIS OF THE D4 LATTICE
ENTER DATA
0,0,0,0
ALL DATA IS IN
```

$d^2_{lattice} = 2$

	COMBINATIONS	SHELL NORM
1	1	0.00
2	24	2.00
3	24	4.00
4	96	6.00
5	24	8.00
6	144	10.00
7	96	12.00
8	192	14.00
9	24	16.00
10	312	18.00
11	144	20.00
12	288	22.00
13	96	24.00
14	288	26.00
15	128	28.00
16	384	30.00
17	24	32.00
18	192	34.00
19	112	36.00
20	192	38.00
21	96	40.00
22	192	42.00
23	32	48.00
24	96	50.00
25	64	52.00
26	16	64.00

	FSUM	EAVE	EPEAK
1	1.000	0.000	0.000
2	25.000	1.920	2.000
3	49.000	2.939	4.000
4	145.000	4.966	6.000
5	169.000	5.396	8.000
6	313.000	7.514	10.000
7	409.000	8.567	12.000
8	601.000	10.303	14.000
9	625.000	10.522	16.000
10	937.000	13.012	18.000
11	1081.000	13.943	20.000
12	1369.000	15.638	22.000
13	1465.000	16.186	24.000
14	1753.000	17.798	26.000
15	1881.000	18.492	28.000
16	2265.000	20.443	30.000
17	2789.000	20.564	32.000
18	2481.000	21.604	34.000
19	2593.000	22.226	36.000
20	2785.000	23.313	38.000
21	2881.000	23.869	40.000
22	3073.000	25.002	42.000
23	3105.000	25.239	48.000
24	3201.000	25.982	50.000
25	3265.000	26.492	52.000
26	3281.000	26.675	64.000

```
STOP --
```

```
.RUN D4EVE
ANALYSIS OF THE D4 LATTICE
ENTER DATA
0,0.5,0,0
ALL DATA IS IN
```

-2.00	-0.50	-1.00	0.00	5.25
-2.00	-0.50	0.00	-1.00	5.25
-2.00	-0.50	0.00	1.00	5.25
-2.00	-0.50	1.00	0.00	5.25
-1.00	-0.50	-2.00	0.00	5.25
-1.00	-0.50	0.00	-2.00	5.25
-1.00	-0.50	0.00	2.00	5.25
-1.00	-0.50	2.00	0.00	5.25
-1.00	1.50	-1.00	-1.00	5.25
-1.00	1.50	-1.00	1.00	5.25
-1.00	1.50	1.00	-1.00	5.25
-1.00	1.50	1.00	1.00	5.25
0.00	-0.50	-2.00	-1.00	5.25
0.00	-0.50	-2.00	1.00	5.25
0.00	-0.50	-1.00	-2.00	5.25
0.00	-0.50	-1.00	2.00	5.25
0.00	-0.50	1.00	-2.00	5.25
0.00	-0.50	1.00	2.00	5.25
0.00	-0.50	2.00	-1.00	5.25
0.00	-0.50	2.00	1.00	5.25
1.00	-0.50	-2.00	0.00	5.25
1.00	-0.50	0.00	-2.00	5.25
1.00	-0.50	0.00	2.00	5.25
1.00	-0.50	2.00	0.00	5.25
1.00	1.50	-1.00	-1.00	5.25
1.00	1.50	-1.00	1.00	5.25
1.00	1.50	1.00	-1.00	5.25
1.00	1.50	1.00	1.00	5.25
2.00	-0.50	-1.00	0.00	5.25
2.00	-0.50	0.00	-1.00	5.25
2.00	-0.50	0.00	1.00	5.25
2.00	-0.50	1.00	0.00	5.25

	COMBINATIONS	SHELL NORM
1	0	0.00
2	1	0.25
3	6	1.25
4	13	2.25
5	14	3.25
6	18	4.25
7	32	5.25 ←
8	31	6.25
9	30	7.25
10	48	8.25
11	38	9.25
12	42	10.25
13	78	11.25
14	57	12.25
15	54	13.25
16	80	14.25
17	62	15.25
18	84	16.25
19	96	17.25
20	74	18.25
21	96	19.25
22	121	20.25
23	102	21.25
24	90	22.25
25	120	23.25
26	98	24.25
27	72	25.25
28	168	26.25
29	80	27.25
30	90	28.25
31	104	29.25
32	84	30.25
33	96	31.25
34	92	32.25
35	80	33.25
36	108	34.25
37	72	35.25
38	76	36.25
39	24	37.25
40	120	38.25
41	32	39.25
42	48	40.25
43	72	41.25
44	48	42.25
45	24	43.25
46	36	44.25
47	24	45.25
48	72	46.25
49	24	47.25
50	32	48.25
51	8	50.25
52	12	52.25
53	24	53.25
54	32	54.25
55	24	56.25
56	8	60.25

--	FSUM	EAVE	EPEAK
1	0.000	0.000	0.000
2	1.000	0.250	0.250
3	7.000	1.107	1.250
4	20.000	1.850	2.250
5	34.000	2.426	3.250
6	52.000	3.058	4.250
7	84.000	3.893	5.250
8	115.000	4.528	6.250
9	145.000	5.091	7.250
10	193.000	5.877	8.250
11	231.000	6.432	9.250
12	273.000	7.019	10.250
13	351.000	7.959	11.250
14	408.000	8.559	12.250
15	462.000	9.107	13.250
16	542.000	9.866	14.250
17	604.000	10.419	15.250
18	688.000	11.131	16.250
19	784.000	11.880	17.250
20	858.000	12.429	18.250
21	954.000	13.116	19.250
22	1075.000	13.919	20.250
23	1177.000	14.554	21.250
24	1267.000	15.101	22.250
25	1387.000	15.806	23.250
26	1485.000	16.363	24.250
27	1557.000	16.774	25.250
28	1725.000	17.697	26.250
29	1805.000	18.120	27.250
30	1895.000	18.601	28.250
31	1999.000	19.155	29.250
32	2083.000	19.603	30.250
33	2179.000	20.116	31.250
34	2271.000	20.608	32.250
35	2351.000	21.038	33.250
36	2459.000	21.618	34.250
37	2531.000	22.006	35.250
38	2609.000	22.432	36.250
39	2633.000	22.567	37.250
40	2753.000	23.250	38.250
41	2785.000	23.434	39.250
42	2833.000	23.719	40.250
43	2905.000	24.154	41.250
44	2953.000	24.448	42.250
45	2977.000	24.599	43.250
46	3013.000	24.834	44.250
47	3037.000	24.995	45.250
48	3109.000	25.488	46.250
49	3133.000	25.654	47.250
50	3165.000	25.883	48.250
51	3173.000	25.944	50.250
52	3185.000	26.043	52.250
53	3209.000	26.247	53.250
54	3241.000	26.523	54.250
55	3265.000	26.742	56.250
56	3273.000	26.824	60.250
57	3281.000	26.925	68.250

STOP --

```
.RUN D4EVE
ANALYSIS OF THE D4 LATTICE
ENTER DATA
0.5,0.5,0,0
ALL DATA IS IN
     -1.50      -0.50      -1.00       0.00       3.50
     -1.50      -0.50       0.00      -1.00       3.50
     -1.50      -0.50       0.00       1.00       3.50
     -1.50      -0.50       1.00       0.00       3.50
     -0.50      -1.50      -1.00       0.00       3.50
     -0.50      -1.50       0.00      -1.00       3.50
     -0.50      -1.50       0.00       1.00       3.50
     -0.50      -1.50       1.00       0.00       3.50
      0.50       1.50      -1.00       0.00       3.50
      0.50       1.50       0.00      -1.00       3.50
      0.50       1.50       0.00       1.00       3.50
      0.50       1.50       1.00       0.00       3.50
      1.50       0.50      -1.00       0.00       3.50
      1.50       0.50       0.00      -1.00       3.50
      1.50       0.50       0.00       1.00       3.50
      1.50       0.50       1.00       0.00       3.50
            COMBINATIONS   SHELL NORM
        1          0           0.00
        2          2           0.50
        3          8           1.50
        4         12           2.50
        5         16           3.50   <--
        6         26           4.50
        7         24           5.50
        8         28           6.50
        9         48           7.50
       10         36           8.50
       11         40           9.50
       12         64          10.50
       13         48          11.50
       14         62          12.50
       15         80          13.50
       16         60          14.50
       17         64          15.50
       18         96          16.50
       19         96          17.50
       20         76          18.50
       21        112          19.50
       22         82          20.50
       23         80          21.50
       24        146          22.50
       25         88          23.50
       26         98          24.50
       27        120          25.50
       28         82          26.50
       29        104          27.50
       30        112          28.50
       31         88          29.50
       32         72          30.50
       33        120          31.50
       34         98          32.50
       35         64          33.50
       36         96          34.50
       37         56          35.50
       38         56          36.50
       39         96          37.50
       40         64          38.50
       41         64          39.50
       42         57          40.50
       43         24          41.50
       44         52          42.50
       45         48          43.50
       46         52          44.50
       47         32          45.50
       48         32          46.50
       49         16          47.50
       50         12          48.50
       51         32          49.50
       52         32          50.50
       53         16          51.50
       54         14          52.50
       55          8          54.50
       56         12          56.50
       57         16          57.50
       58         12          58.50
       59          8          60.50
       60          8          64.50
       61          4          72.50
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
.RUN D4EVE
ANALYSIS OF THE D4 LATTICE
ENTER DATA
1.,0.,0.,0.
ALL DATA IS IN
MOD IS OK
```

	COMBINATIONS	SHELL NORM
1	8	1.00
2	32	3.00
3	48	5.00
4	64	7.00
5	104	9.00
6	96	11.00
7	112	13.00
8	192	15.00
9	138	17.00
10	152	19.00
11	232	21.00
12	192	23.00
13	211	25.00
14	212	27.00
15	174	29.00
16	88	31.00
17	252	33.00
18	192	35.00
19	152	37.00
20	48	39.00
21	78	41.00
22	92	43.00
23	168	45.00
24	24	47.00
25	64	49.00
26	48	51.00
27	52	57.00
28	24	59.00
29	24	61.00
30	8	73.00

	FSUM	EAVE	EPEAK
1	8.000	1.000	1.000
2	40.000	2.600	3.000
3	88.000	3.909	5.000
4	152.000	5.211	7.000
5	256.000	6.750	9.000
6	352.000	7.909	11.000
7	464.000	9.138	13.000
8	656.000	10.854	15.000
9	794.000	11.922	17.000
10	946.000	13.059	19.000
11	1178.000	14.623	21.000
12	1370.000	15.797	23.000
13	1581.000	17.025	25.000
14	1793.000	18.205	27.000
15	1967.000	19.160	29.000
16	2055.000	19.667	31.000
17	2307.000	21.123	33.000
18	2499.000	22.189	35.000
19	2651.000	23.038	37.000
20	2699.000	23.322	39.000
21	2777.000	23.819	41.000
22	2869.000	24.434	43.000
23	3037.000	25.572	45.000
24	3061.000	25.740	47.000
25	3125.000	26.216	49.000
26	3173.000	26.591	51.000
27	3225.000	27.081	57.000
28	3249.000	27.317	59.000
29	3273.000	27.564	61.000
30	3281.000	27.675	73.000

STOP --

	FSUM	EAVE	EPEAK
1	0.000	0.000	0.000
2	2.000	0.500	0.500
3	10.000	1.300	1.500
4	22.000	1.955	2.500
5	38.000	2.605	3.500
→ 6	64.000	3.375	4.500 ←
7	88.000	3.955	5.500
8	116.000	4.569	6.500
9	164.000	5.427	7.500
10	200.000	5.980	8.500
11	240.000	6.567	9.500
12	304.000	7.395	10.500
13	352.000	7.955	11.500
14	414.000	8.635	12.500
15	494.000	9.423	13.500
16	554.000	9.973	14.500
17	618.000	10.545	15.500
18	714.000	11.346	16.500
19	810.000	12.075	17.500
20	886.000	12.626	18.500
21	998.000	13.398	19.500
22	1080.000	13.937	20.500
23	1160.000	14.459	21.500
24	1306.000	15.358	22.500
25	1394.000	15.872	23.500
26	1492.000	16.438	24.500
27	1612.000	17.113	25.500
28	1694.000	17.567	26.500
29	1798.000	18.142	27.500
30	1910.000	18.749	28.500
31	1998.000	19.223	29.500
32	2070.000	19.615	30.500
33	2190.000	20.266	31.500
34	2288.000	20.790	32.500
35	2352.000	21.136	33.500
36	2448.000	21.660	34.500
37	2504.000	21.970	35.500
38	2560.000	22.288	36.500
39	2656.000	22.837	37.500
40	2720.000	23.206	38.500
41	2784.000	23.580	39.500
42	2841.000	23.920	40.500
43	2865.000	24.067	41.500
44	2917.000	24.396	42.500
45	2965.000	24.705	43.500
46	3017.000	25.046	44.500
47	3049.000	25.261	45.500
48	3081.000	25.481	46.500
49	3097.000	25.595	47.500
50	3109.000	25.684	48.500
51	3141.000	25.926	49.500
52	3173.000	26.174	50.500
53	3189.000	26.301	51.500
54	3213.000	26.497	52.500
55	3221.000	26.566	54.500
56	3233.000	26.678	56.500
57	3249.000	26.829	57.500
58	3261.000	26.946	58.500
59	3269.000	27.028	60.500
60	3277.000	27.119	64.500
61	3281.000	27.175	72.500

STOP —