IN-32-CR
51732
81P

Semi-Annual Status Report

submitted to

NASA

Goddard Space Flight Center

Greenbelt, Maryland

Error Control Techniques for Satellite and Space Communications

Grant Number NAG5-557

June 1, 1985 - May 31, 1987

Principal Investigator:

Daniel J. Costello, Jr.

Department of Electrical Engineering

University of Notre Dame

Notre Dame, IN 46556

February, 1987

Summary of Progress

During the period June 1, 1986 - November 30, 1986, progress was made in the following areas:

1) Undetected Error Probability and Throughput Analysis of a Concatenated Coding Scheme.

A paper summarizing our work on the performance analysis of NASA's telecommand system has been accepted for publication by the *IEEE Transactions on Communications* [1]. Copies of this paper were included in a previous report.

2) Capacity and Cutoff Rate Analysis of Concatenated Codes.

A paper summarizing our work on analyzing the capacity and cutoff rate of the "outer channel" formed by the combination of the actual physical channel and the inner encoder and decoder in a NASA concatenated coding system has been accepted for publication by the *IEEE Transactions on Information Theory* [2]. Copies of this paper were included in a previous report.

3) Concatenated Codes Using Bandwidth Efficient Trellis Inner Codes.

A paper summarizing our work on determining the performance of bandwidth efficient trellis inner codes for use in a NASA concatenated coding system is being prepared for submission to the *IEEE Transactions on Communications* [3]. A presentation on this work was also made at the 1986 IEEE International Symposium on Information Theory [4]. A preliminary version of the paper was included in a previous report.

4) Bounds on the Minimum Free Euclidean Distance of Bandwidth Efficient Trellis Codes.

A paper summarizing our work on obtaining lower bounds on the minimum free Euclidean distance of bandwidth efficient trellis codes is being prepared for submission to the *IEEE Transactions on Information Theory* [5]. Two preliminary papers on this subject were recently presented at conferences [6,7]. Copies of these papers are included as Appendices to this report. This work is primarily theoretical and

establishes achievable lower bounds on free distance for trellis codes. These bounds can then be used as a benchmark to compare the performance of various codes, including those which we have constructed.

5) Construction of Multidimensional Bandwidth Efficient Trellis Codes for Use as Inner Codes in a Concatenated Coding System.

Unit-memory (UM) and partial-unit-memory (PUM) convolutional codes are known to offer a performance advantage over standard convolutional codes when used as inner codes in a concatenated coding system [8]. We have investigated the use of multidimensional bandwidth efficient UM and PUM trellis coded phase modulation for use in a NASA concatenated coding system. We feel that these codes will also offer a performance advantage over standard bandwidth efficient trellis codes when used in a concatenated system. These codes are well suited for use with multidimensional signal constellations, which are known to offer an additional performance advantage over two-dimensional constellations [9]. Details of our performance analysis of these codes will be included in our next report.

The construction of the multidimensional UM and PUM trellis codes offered a number of interesting challenges. A paper containing all the details of this construction has been submitted for publication to the *IEEE Transactions of Information Theory* [10] and is also included as an Appendix to this report. A brief summary of this work now follows.

Trellis coded modulation (TCM) can be classified into two basic types, the lattice-type (e.g., M-AM, M-QASK) and the constant-envelope-type (e.g., MPSK). The latter has a slightly lower power efficiency compared with the former but is more suitable for nonlinear band-limited channels. In any TCM design, partitioning of the signal set into subsets with increasing intra-subset distances plays a central role [11]. It defines the signal mapping used by the modulator and provides a tight bound on the Euclidean distance (ED), which permits an efficient search for optimum codes. For lattice-type TCM, Calderbank and Sloane [12] have made the important observation that the partioning of the signal set into subsets corresponds to the partitioning of a

lattice into a sublattice and its cosets. Forney [13] then developed a method, called the "squaring construction", of partitioning higher dimensional lattices from a lower dimensional lattice by using a coset code. However, his approach does not apply to constant-envelope-type TCM.

In this work, we investigate unit-memory (UM) and partial-unit-memory (PUM) trellis coded $2L$-dimensional ($L \geq 2$) 8PSK (or L*8PSK) modulation. The L*8PSK signal set is generated simply by repeating an 8PSK signal L times. Therefore, the $2L$-dimensional 8PSK signal set is the Cartesian product of L 2-dimensional 8PSK signal sets, i.e., $S_{2L} = S_2 \times S_2 \times \cdots \times S_2$ (L times). S_{2L} has some interesting features which are not found in S_2. Codes with higher effective information rates and larger coding gains are possible due to the increased flexibility of coding in S_{2L}.

The construction of trellis coded L*8PSK modulation is accomplished by using Ungerboeck's [11] concept of "mapping by set partitioning". A systematic approach to partitioning the $2L$-dimensional 8PSK signal space into subsets is presented. This approach simplifies both the construction of the coded L*8PSK modulation and the corresponding binary convolutional encoder design. The set partitioning of the $2L$-dimensional 8PSK signal space into subsets is shown to be equivalent to partitioning the L-dimensional binary vector space into a subcode and its cosets. As examples, we give the signal space partitions for 2, 4, and 8-dimensional 8PSK modulation.

TCM has been mostly restricted to the case where the code rate $R = (n-1)/n$. In our code constructions, however, we remove this condition by considering code rates $R=(3L-i)/3L$, $L = 2, 3, 4$, $i = 1, 2, \cdots$, such that the effective information rate $R_{eff} \geq 1$ bits/dimension. TCM designs are illustrated by several examples. The procedure for implementing the TCM designs through the use of binary UM/PUM convolutional encoders is also presented. The UM/PUM trellis coded 2, 4, and 8-dimensional 8PSK modulation is specified by listing the code generator matrices. Codes with coding gains of up to 1.9 *dB* over those achieved with trellis coded 2-dimensional modulation are obtained.

REFERENCES

[1] R. H. Deng and D. J. Costello, Jr., "Reliability and Throughput Analysis of a Concatenated Coding Scheme," *IEEE Trans. Commun.*, accepted for publication.

[2] M. A. Herro, D. J. Costello, Jr., and L. Hu, "Capacity and Cutoff Rate Calculations for a Concatenated Coding System," *IEEE Trans. Inform. Th.*, accepted for publication.

[3] R. H. Deng and D. J. Costello, Jr., "High Rate Concatenated Coding Systems with Bandwidth Efficient Inner Codes," *IEEE Trans. Commun.*, submitted for publication.

[4] D. J. Costello, Jr. and R. H. Deng, "Concatenated Coding Systems Employing Bandwidth Efficient Inner Codes," *IEEE Int. Symp. on Inform. Th.*, Ann Arbor, MI, October 1986.

[5] M. Rouanne and D. J. Costello, Jr., "A Gilbert Lower Bound on the Minimum Free Euclidean Distance of Trellis Coded Modulation," *IEEE Trans. Infom. Th.*, submitted for publication.

[6] M. Rouanne and D. J. Costello, Jr., "Comparison of Trellis Coded Modulation Schemes Using a Lower Bound on the Free Euclidean Distance," Proc. Allerton Conf. on Commun., Cont., and Comp., pp. 1037-1046, Monticello, IL, October 1986.

[7] M. Rouanne and D. J. Costello, Jr., "A Lower Bound on the Minimum Euclidean Distance of Trellis Codes," Colloque 3 Journees sue le Codage, Paris, France, November 1986.

[8] L.-N. Lee, "Concatenated Coding Systems Employing a Unit-Memory Convolutional Code and a Byte-Oriented Decoding Algorithm," *IEEE Trans. Commun.*, COM-25, pp. 1064-1074, October 1977.

[9] L.-F. Wei, "Trellis-Coded Modulation with Multi-Dimensional Constellations," *IEEE Trans. Inform. Th.*, accepted for publication.

[10] A. LaFanchere, R. H. Deng, and D. J. Costello, Jr., "Multidimensional Trellis Coded Phase Modulation Using Unit-Memory and Partial-Unit-Memory Convolutional Codes," *IEEE Trans. Inform. Th.*, submitted for publication.

[11] G. Ungerboeck, "Channel Coding with Multilevel/Phase Signals," *IEEE Trans. Inform. Th.*, Vol. IT-28, pp. 55-67, January 1982.

[12] A. R. Calderbank and N. J. A. Sloane, "New Trellis Codes," *IEEE Trans. Inform. Th.*, to appear, 1987.

[13] G. D. Forney, Jr., "Coset Codes", submitted to *IEEE Trans. Inform. Th.*, 1986.

Appendix A

Comparison of Trellis Coded Modulation Schemes Using a Lower Bound on the Free Euclidean Distance

COMPARISON OF TRELLIS CODED MODULATION SCHEMES USING A LOWER BOUND ON THE FREE EUCLIDEAN DISTANCE

MARC ROUANNE

DANIEL J. COSTELLO, Jr.

Dept. of Elec. & Comp. Engr

Univ. of Notre Dame

Notre Dame, IN 46556

ABSTRACT

A comparison of Trellis Coded Modulation (TCM) schemes is considered. The approach is based upon a lower bound on the minimum free Euclidean distance d_{free} of TCM. The bound is similar to the bound of Costello [1] and Forney [2] on the free distance of convolutional codes. We evaluate the lower bound on d_{free} for various modulation constellations and trellis codes, such as those proposed by Ungerboeck [3], and Lafanechere and Costello [4], and compare it with Calderbank, Mazo, and Wei's upper bound [5,6].

PROBLEM FORMULATION

A TCM scheme is defined by a binary trellis encoder, a signal constellation, and a mapping of signals onto the trellis. Let k be the information block length and ν the memory length of the encoder. Then $\nu_0 = k\,\nu$ is the constraint length of the code (we assume that the encoder contains k memory registers of equal length ν). The encoder is a finite state machine with 2^{ν_0} states and 2^k branches to and from each state. The error probability of a code used with maximum-likelihood (Viterbi) decoding on an AWGN channel can be bounded in terms of its minimum free distance d_{free}. Therefore an efficient mapping will assign channel signals to branches to achieve maximum d_{free} when using maximum likelihood decoding. The lower bound on d_{free} uses a random coding argument based on the moment generating function $\overline{e^{\alpha\rho d^2} \sum_{\mathbf{x}} \left[\sum_{\mathbf{x}'} e^{-\alpha d_e^2(\mathbf{x},\mathbf{x}')} \right]^{\rho}}$, where the overbar indicates an average over the ensemble of all code

This work was supported by NASA grant NAG5-557 and NSF grant ECS84-14608.

choices (for a given trellis), the first sum is over all choices of paths $\underline{y}$, the second sum is over all paths $\underline{y}'$ diverging from $\underline{y}$, α and ρ are arbitrary constants, d is the lower bound on d_{free}, and d_e is the Euclidean distance associated with a particular pair of paths. The lower bound is a function of a trellis and a signal constellation, but not of a specific mapping (i.e., it is a random coding bound). A trellis is characterized by ν_0 and k. A signal constellation is characterized by its dimensionality, energy, point separation and set partitioning. We now see how these parameters affect the lower bound on d_{free}.

DERIVATION OF THE BOUND

Forney [2] set forth clearly the following bounding technique, which owes much to Chernoff [7], Gallager [8], and Viterbi [9]. The technique includes 6 steps. We use only the first 4.

1) Step 1: Define a random ensemble

The random ensemble is the set of all "non-linear, time-varying codes" associated with a specific signal constellation and a specific trellis.

2) Step 2: Obtain a bound on the moment-generating function: $\overline{e^{\alpha\rho d^2} \sum_{\underline{y}} \left[\sum_{\underline{y}'} e^{-\alpha d_e^2(\underline{y},\underline{y}')} \right]^{\rho}}$

If the probability over all codes in the random ensemble that d_{free} is smaller than some d is less than 1, there exists a code for which d_{free} is actually larger than or equal to d. This can be written as follows:

$$P_{\text{all codes}}(d_{free}(code) < d) < 1 \rightarrow \textit{there exists a code such that } d_{free} \geq d.$$

Furthermore,

$$d_{free}(code) \geq d \quad \text{if} \quad T_{\alpha,\rho}(code,d) < 1$$

where $$T_{\alpha,\rho}(code,d) \equiv e^{\alpha\rho d^2} \sum_{\underline{y}} \left[\sum_{\underline{y}'} e^{-\alpha d_e^2(\underline{y},\underline{y}')} \right]^{\rho}.$$

Hence,

$$\textit{there exists a code such that } d_{free} \geq d \text{ if } \overline{e^{\alpha\rho d^2} \sum_{\underline{y}} \left[\sum_{\underline{y}'} e^{-\alpha d_e^2(\underline{y},\underline{y}')} \right]^{\rho}} < 1. \qquad (1)$$

3) Step 3: Configuration counting

Equation (1) contains an average over all codes of a function $T_{\alpha,\rho}$. $T_{\alpha,\rho}(code)$ depends on the codewords of a specific *code*. Shannon's random coding technique switches the sum over all codes and the sum over all codewords and obtains an average over all codewords of another function $\hat{T}_{\alpha,\rho}$. $\hat{T}_{\alpha,\rho}(codeword)$ represents the probability over all codes that a specific *codeword* is chosen. Here, the codewords are the paths through the trellis. "Configuration counting" determines the number of occurrences of a specific path in all the codes. This enables us to calculate $\hat{T}_{\alpha,\rho}$.

4) Step 4: Reduce the above bound to a usable form (Gallager's Lemma)

Step 3 puts the bound in random coding form. Step 4 (see Gallager [8]) puts the bound in its final form.

STATEMENT OF THE BOUND

Given a signal constellation S, there exists a (k, ν_0) trellis code with minimum Euclidean distance d_{free} such that:

$$d_{free}^2 \geq \max_{\substack{E(\alpha,\rho) > \ln 2 \\ 0 \leq \alpha \\ 0 \leq \rho \leq 1 \\ p(s)}} \left[\nu_0 \frac{E(\alpha,\rho)}{\alpha} - \frac{F[E(\alpha,\rho)]}{\alpha} \right] \tag{2}$$

where:

k is the information block size,

ν_0 is the constraint length,

S is the signal constellation,

α and ρ are parameters to optimize the bound,

$$E(\alpha,\rho) = -\ln\left[\sum_{s \in S} p(s) \left[\sum_{s' \in S} p(s')\, e^{-\alpha\, d_e^2(s,s')} \right]^{\rho} \right]^{\frac{1}{\rho k}},$$

s and s' are signal points from the signal constellation S,

$d_e(s,s')$ is the Euclidean distance between the signal points s and s',

and $p(s)$ is a probability distribution on S.

APPLICATION OF THE BOUND

A signal s can itself be a subset of the constellation when a trellis contains parallel transitions. The subset is composed of signals assigned to the parallel transitions. This enables us to study the effects of uncoded bits in the encoder, or, equivalently, the effects of mapping signals onto a trellis with parallel transitions using Ungerboeck's "set partitioning" method [3].

A trellis is entirely characterized by k and ν_0. Thus each evaluation of the bound corresponds to one trellis, and represents the average free distance of all the corresponding trellis codes. However, since the bound is a random coding bound, it does not represent the best free distance achievable with a particular trellis.

Lower bounds on d_{free} have been obtained for binary convolutional codes using the Hamming distance metric [1,2]. Like the lower bounds on the Hamming d_{free}, this lower bound on the Euclidean d_{free} is exponentially tight and varies linearly with ν_0 for large ν_0. Like any other "Gilbert type" bound, this lower bound is not expected to be tight for small ν_0.

The bound depends explicitly on the constraint length ν_0 and implicitly on the information rate per dimension $R \equiv \frac{k}{dim}$. Although only certain discrete values of ν_0 and R are possible, we sketch $d^2_{free}(\nu_0)$ and $\frac{d_{free}}{\nu_0}(R)$ as continuous functions.

USING THE BOUND TO EVALUATE TCM SCHEMES

1) d_{free} viewed as a function of ν_0:

The lower bound on d_{free} increases with the constraint length ν_0. Bounds on the free Hamming distance of binary convolutional codes have usually been expressed as functions of ν_0. This leads to asymptotic, exponentially tight bounds on d_{free} when ν_0 is large but the bounds are not tight for small constraint lengths. In the following paragraphs we compare constellations at a given rate R of information bits per dimension and at a given average energy E_{avg} per dimension.

* The effect of increasing the number of signals on d_{free} (same rate R, same dimensionality dim, same average energy E_{avg}):

Increasing the number M of signals increases d_{free} because it provides more flexibility in the choice of signals assigned to trellis branches (flexibility in the mapping). Signals with greater distance can be chosen on branches of merging paths, which increases d_{free}. Note that increasing M, while keeping k constant, lowers the code rate $R_c \equiv \frac{k}{\log_2 M}$. Fig.1 shows PAM schemes with different numbers of signals (M = 4, 8, 16, 128). Increasing the number of signals provides more gain for small constraint lengths ν_0 than for large constraint lengths, because when ν_0 is large flexibility in mapping is provided by the complexity of the encoder. In Fig.1, going from 4-PAM to 8-PAM or even 16-PAM provides most of the gain and higher order PAM schemes do not improve d_{free} significantly. Note that the gain in d_{free} may not be worth the added modulation complexity.

* The effect of increasing the signal set dimensionality on d_{free} (same rate R, same number of signals M, same average energy E_{avg}):

Increasing the signal set dimensionality increases d_{free}. Higher dimensional schemes are obtained by using a basic one- or two-dimensional constellation for several transmission intervals [4]. Fig.2 shows L-dimensional schemes for which the constellation is constructed from L uses of a given 1-dimensional constellation (here 4-PAM). For a given constraint length ν_0, higher dimensionality yields a larger d_{free}, and as ν_0 increases the curves diverge and the gain goes to infinity. However, the error coefficient increases with dimensionality [10], which may cancel the gain from the increased d_{free} at moderate decoded bit error rates.

* The effect of changing the modulation scheme on d_{free} (same dimensionality dim, same number of signals M, same average energy E_{avg}, same rate R):

Changing the modulation scheme affects d_{free}. For example, rectangular constellations yield a larger d_{free} than constant envelope constellations. Fig.3 shows M-PSK vs. M-QASK. This suggests that a d_{free} penalty is associated with constant envelope modulation. Note that this penalty increases with the constraint length. Furthermore, the bound gives a means of optimizing some parameters associated with a specific type of constellation without searching exhaustively for the best constellation.

* The effect of uncoded bits or parallel transitions on d_{free} when the signal constellation is "mapped by set partitioning":

Uncoded bits generate parallel transitions in the trellis and limit d_{free}, since d_{free} cannot be larger than the minimum distance between parallel transitions (the minimum distance within the subsets created by "set partitioning"). Fig.4 shows that for 8-PSK, 1 uncoded bit increases d_{free} for small constraint lengths. For small ν_0, this agrees with the best known codes constructed [3]. For large ν_0, however, we see that 1 uncoded bit limits the achievable d_{free}. The bound shows whether uncoded bits combined with mapping by set partitioning will increase d_{free} for small constraint lengths ν_0.

2) Comparison of the lower bound on d_{free} with upper bound [5,6] and known codes [3]:

The upper bound is expected to be tighter than the lower bound for small constraint lengths, and the lower bound is expected to be tighter for large constraint lengths (this is analogous to the way bounds on binary convolutional codes behave). The lower bound is a "Chernoff type" bound and therefore is exponentially tight when ν_0 is large. Fig.5 shows that the upper bound is tighter than the lower bound for small constraint lengths. But the lower bound becomes tighter as ν_0 increases. The slope of the lower bound gives a precise indication of the asymptotic rate of increase in d_{free}. Finally, the lower bound guaranties the existence of codes that can achieve a certain d_{free}.

3) Asymptotic behaviour: $\frac{d_{free}}{\nu_0}$ as a function of $R = \frac{k}{dim}$ for large ν_0.

For a given constellation, the achievable $\frac{d_{free}}{\nu_0}$ is larger for small R than for large R. Graphs of $\frac{d_{free}}{\nu_0}(R)$ give a means of comparing the asymptotic behaviour (large ν_0) of d_{free} for different constellations. Several examples are shown in Fig.6. The largest achievable rate R depends on the constellation and its dimensionality : it is $\frac{\log_2 M}{dim} R_c$ for a code rate R_c. These largest achievable rates appear clearly in Fig.6. Rectangular lattices constructed from the same 1-dimensional constellation (for example 4-PAM in Fig.6) yield the same graph, which means that these rectangular lattices have the same asymptotic behaviour. In other words the curves $d_{free}(\nu_0)$ are parallel for large ν_0. The lower bounds are compared with an asymptotic upper bound [6] in Fig.6.

CONCLUSION

We have obtained a random coding lower bound on d_{free}. It takes into account the actual Euclidean distances between points of the channel signal constellation. As with any "Gilbert" bound, it is most useful as an asymptotic bound. It provides a means of comparing the asymptotic performance of different modulation schemes.

REFERENCES

[1] D. J. Costello, Jr. "Free Distance Bounds for Convolutional Codes", IEEE Trans. Inform. Theory, vol. IT-20, pp. 356-365, May 1974.

[2] G. D. Forney, Jr. "Convolutional Codes II: Maximum Likelihood Decoding", Information and Control , vol. 25, pp. 222-266, July 1974.

[3] G. Ungerboeck, "Channel Coding with Multilevel Phase Signals", IEEE Trans. Inform. Theory, vol. IT-28, pp. 55-67, Jan. 1982.

[4] A. Lafanechere and D. J. Costello, Jr., "Multidimensional Coded PSK Systems Using Unit-Memory Trellis Codes", Proceedings Allerton Conf. on Communication, Control, and Computing, pp. 428-429, Monticello, IL, Sept. 1985.

[5] A. R. Calderbank, J. E. Mazo, and V. K. Wei, "Asymptotic Upper Bounds on the Minimum Distance of Trellis Codes", IEEE Trans. Commun. , vol. COM 33, no. 4, pp. 305-310, April 1985.

[6] A. R. Calderbank, J. E. Mazo, and H. M. Shapiro, "Upper Bounds on the Minimum Distance of Trellis Codes", Bell Syst. Tech. J. vol. 62, pp. 2617-2646, Oct. 1983, part I.

[7] H. Chernoff, "A Measure of Asymptotic Efficiency for Tests of a Hypothesis Based on a Sum of Observations," Ann. Math. Stat. , vol. 23, pp. 493-507, 1952.

[8] R. G. Gallager, "A simple Derivation of the Coding Theorem and Some Applications," IEEE Trans. Inform. Theory , vol. IT-11, pp. 3-18, Jan. 1965.

[9] A. J. Viterbi, "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm," IEEE Trans. Inform. Theory , vol. IT-13, pp. 260-269, Apr. 1967.

[10] G. D. Forney, Jr., R. G. Gallager, G. R. Lang, F. M. Longstaff, and S. U. Qureshi, "Efficient Modulation for Band-limited Channels", IEEE J. on Selected Areas in Communication , vol. SAC-2, no. 5, pp. 632-647, Sept. 1984.

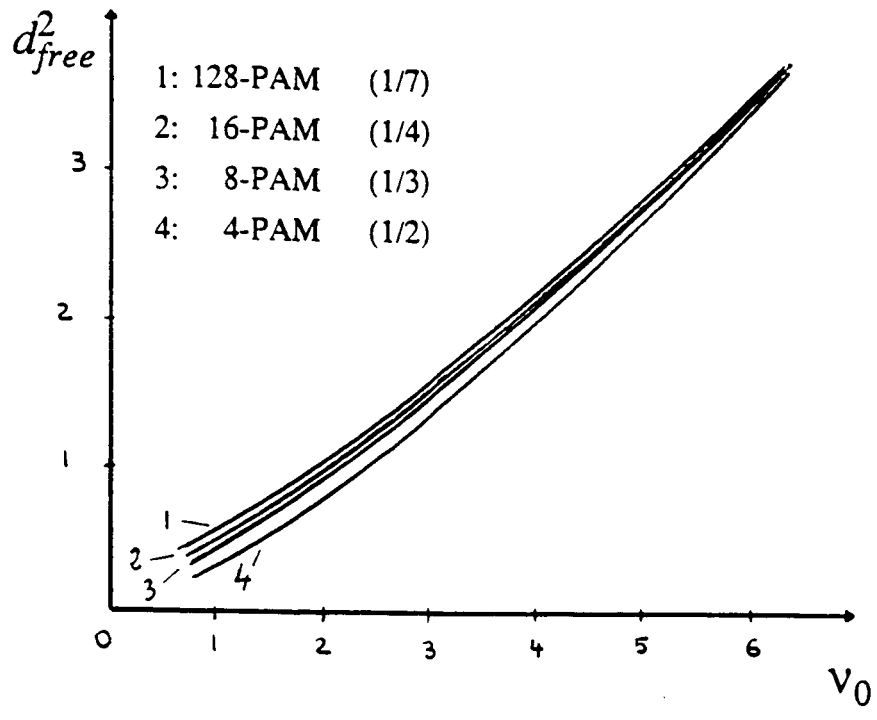


Fig. 1: PAM comparison, $R = 1 bit/dim$, $E_{avg} = 1$.

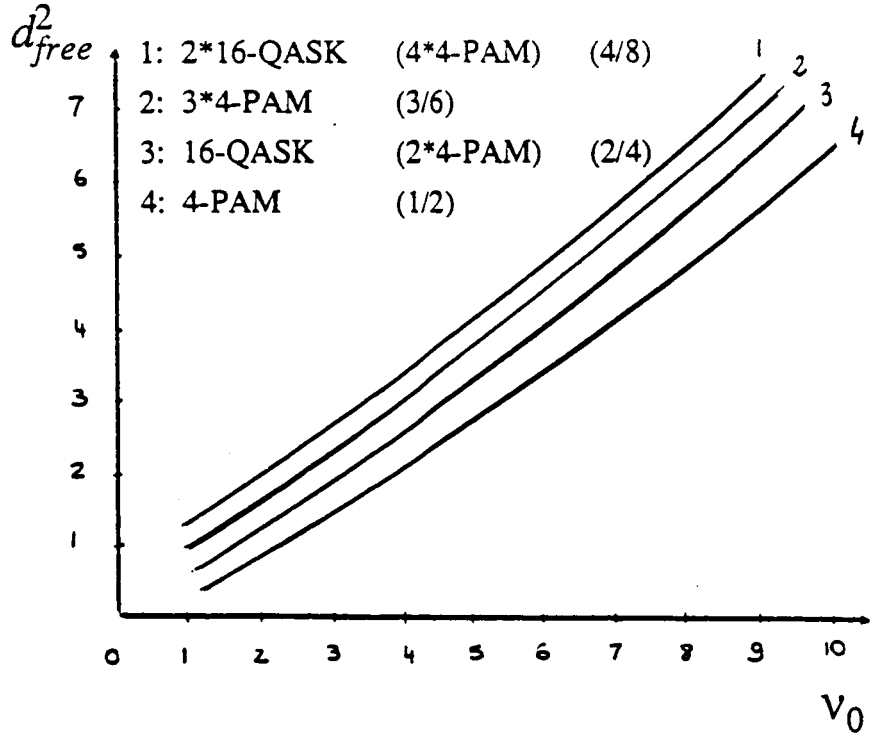


Fig. 2: Comparison of schemes with different dimensionalities, $R = 1 bit/dim$, $E_{avg} = 1$.

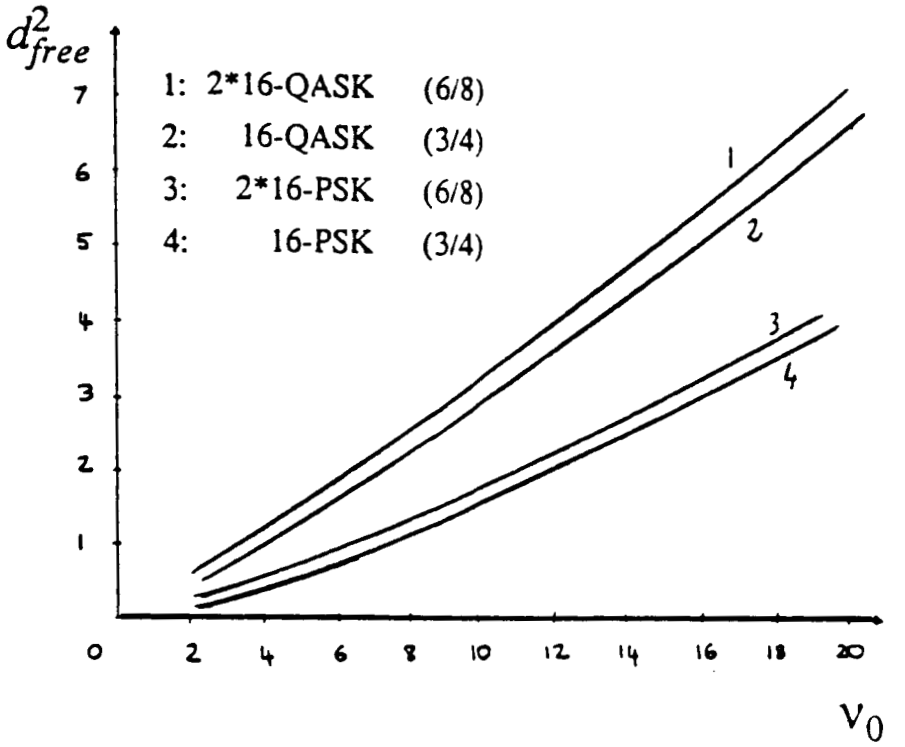


Fig. 3: Comparison of M-PSK, M-QASK, L*M-PSK and L*M-QASK schemes, R = 1.5 *bit/dim*, E_{avg} = 1.

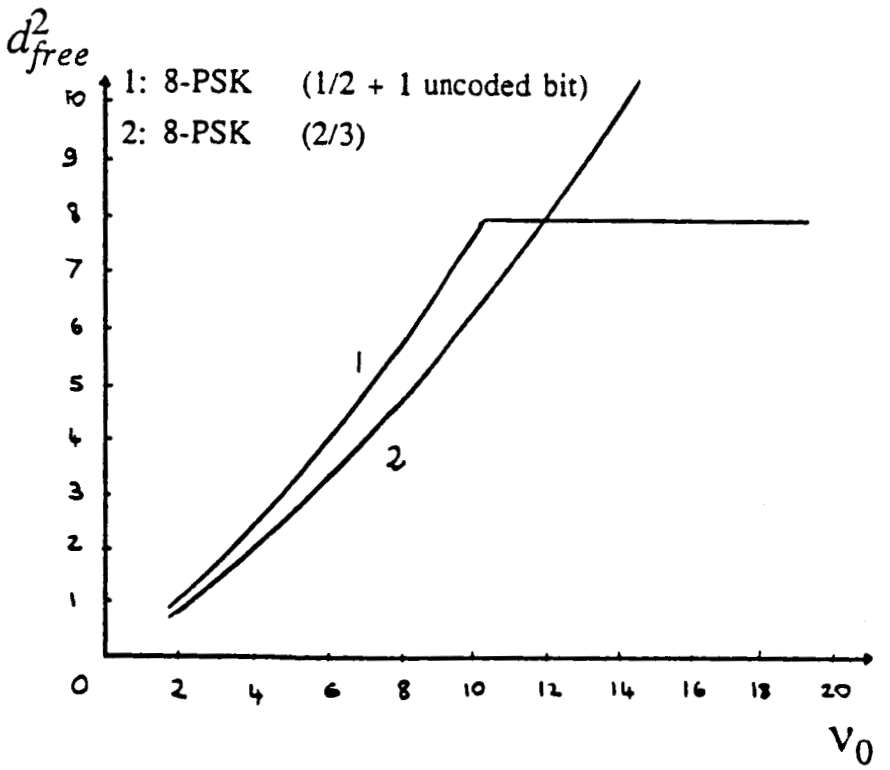


Fig. 4: Trellis coding and uncoded bits (Ungerboeck) for 8-PSK. R = 1 *bit/dim*, E_{avg} = 1.

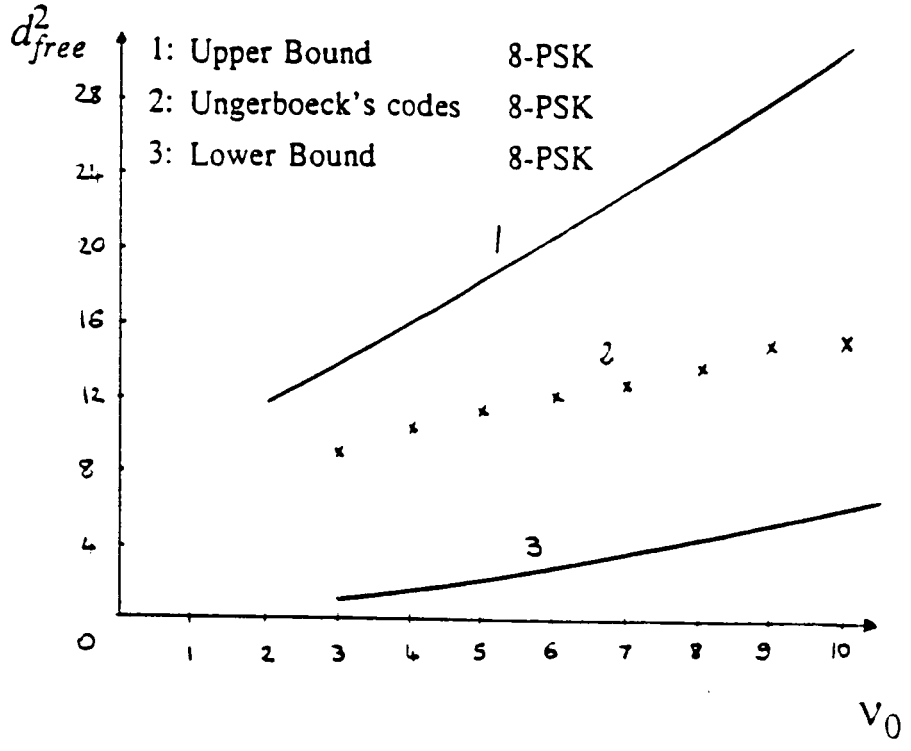


Fig. 5: Comparison of the lower bound on d^2_{free} with upper bounds (Calderbank, Mazo, and Wei).

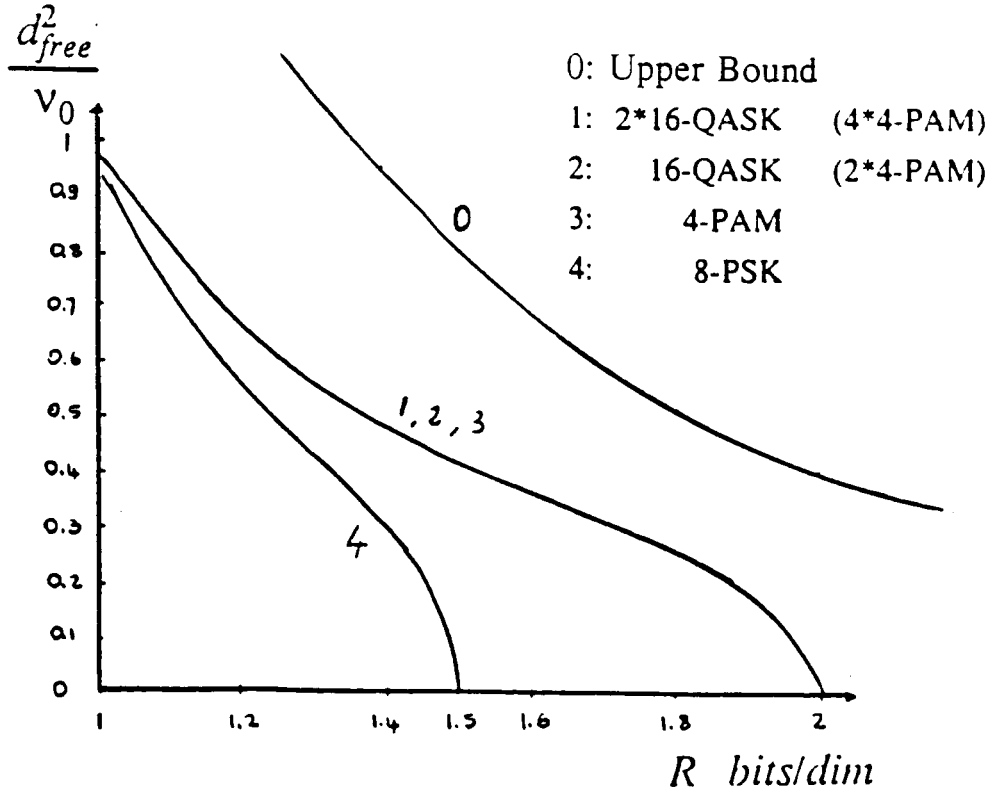


Fig. 6: Asymptotic bounds vs rate, $E_{avg} = 1$.

COMPARISON OF TRELLIS CODED MODULATION SCHEMES USING A LOWER BOUND ON THE FREE EUCLIDEAN DISTANCE

MARC ROUANNE

DANIEL J. COSTELLO, Jr.

Dept. of Elec. & Comp. Engr

Univ. of Notre Dame

Notre Dame, IN 46556

ABSTRACT

A comparison of Trellis Coded Modulation (TCM) schemes is considered. The approach is based upon a lower bound on the minimum free Euclidean distance d_{free} of TCM. The bound is similar to the bound of Costello [1] and Forney [2] on the free distance of convolutional codes. We evaluate the lower bound on d_{free} for various modulation constellations and trellis codes, such as those proposed by Ungerboeck [3], and Lafanechere and Costello [4], and compare it with Calderbank, Mazo, and Wei's upper bound [5,6].

PROBLEM FORMULATION

A TCM scheme is defined by a binary trellis encoder, a signal constellation, and a mapping of signals onto the trellis. Let k be the information block length and ν the memory length of the encoder. Then $\nu_0 = k\,\nu$ is the constraint length of the code (we assume that the encoder contains k memory registers of equal length ν). The encoder is a finite state machine with 2^{ν_0} states and 2^k branches to and from each state. The error probability of a code used with maximum-likelihood (Viterbi) decoding on an AWGN channel can be bounded in terms of its minimum free distance d_{free}. Therefore an efficient mapping will assign channel signals to branches to achieve maximum d_{free} when using maximum likelihood decoding. The lower bound on d_{free} uses a random coding argument based on the moment generating function $\overline{e^{\alpha\rho d^2} \sum_{x} \left[\sum_{x'} e^{-\alpha d_e^2(x,x')} \right]^{\rho}}$, where the overbar indicates an average over the ensemble of all code

This work was supported by NASA grant NAG5-557 and NSF grant ECS84-14608.

Appendix B

A Lower Bound on the Minimum Euclidean Distance of Trellis Codes

A LOWER BOUND ON THE MINIMUM EUCLIDEAN DISTANCE OF TRELLIS CODES

Marc Rouanne

Daniel J. Costello, Jr.

Dept. of Elec. & Comp. Engr.

Univ. of Notre Dame

Notre Dame, IN 46556

Submitted to

"Trois journées sur le codage"

December 1986

Abstract: A lower bound on the minimum Euclidean distance of trellis codes is considered. The bound is based upon Costello's free distance bound for convolutional codes [1]. The bound is a random coding bound over the ensemble of nonlinear time-varying Euclidean trellis codes. We compare schemes using different signal constellations and mappings and apply the bound to particular trellis coded modulation (TCM) schemes such as Ungerboeck's [3] and Lafanechere and Costello's [4].

Keywords: Trellis-coded modulation, minimum Euclidean distance, random coding bound.

This work was supported by NASA grant NAG5-557 and NSF grant ECS84-14608.

Mots-clés: Codes de treillis et modulation, distance Euclidiennne minimale, borne minorante.

1 INTRODUCTION

A TCM scheme is defined by a binary trellis encoder, a signal constellation, and a mapping of signals onto the trellis. Let k be the information block length and ν the memory length of the encoder. Then $\nu_0 = k\,\nu$ is the constraint length of the code (we assume that the encoder contains k memory registers of equal length ν). The encoder is a finite state machine with 2^{ν_0} states and 2^k branches to and from each state. The error probability of a code used with maximum-likelihood (Viterbi) decoding on an AWGN channel can be bounded in terms of its free distance d_{free}. Therefore an efficient mapping will assign channel signals to branches to achieve maximum free distance $\max(d_{free})$. The lower bound on $\max(d_{free})$ uses a random coding argument based on the moment generating function $\overline{e^{\alpha\rho d^2} \sum_{\underline{y}} \left[\sum_{\underline{y}'} e^{-\alpha d_e^2(\underline{y},\underline{y}')}\right]^\rho}$, where the overbar indicates an average over the ensemble of all code choices (for a given trellis), the first sum is over all choices of paths $\underline{y}$, the second sum is over all paths $\underline{y}'$ diverging from $\underline{y}$, α and ρ are arbitrary constants, d is the lower bound on $\max(d_{free})$, and $d_e(\underline{y},\underline{y}')$ is the Euclidean distance between paths $\underline{y}$ and $\underline{y}'$.

2 DERIVATION OF THE BOUND

Forney [2] set forth clearly in six steps a bounding technique which owes much to Chernoff [7], Gallager [8], and Viterbi [9]. We use only the first four steps.

2.1 Step 1: Define a random ensemble

The random ensemble is the set of all "non-linear, time-varying codes" associated with a specific signal constellation and a specific trellis.

2.2 Step 2: Obtain a bound of the moment-generating function form:

If the probability over all codes in the random ensemble that d_{free} is smaller than some d is less than 1, there must exist at least one code for which d_{free} is actually larger than or equal to d. This can be written as follows:

$$\underset{\text{all codes}}{P}(d_{free}(code) < d) < 1 \rightarrow \textit{there exists a code such that } d_{free} \geq d.$$

Furthermore,

$$d_{free}(code) \geq d \quad \text{if} \quad T_{\alpha,\rho}(code,d) < 1,$$

where $$T_{\alpha,\rho}(code,d) \equiv e^{\alpha\rho d^2} \sum_{\gamma} \left[\sum_{\gamma'} e^{-\alpha d_e^2(\gamma,\gamma')} \right]^{\rho}.$$

Hence,

$$\textit{there exists a code such that } d_{free} \geq d \text{ if } \overline{e^{\alpha\rho d^2} \sum_{\gamma} \left[\sum_{\gamma'} e^{-\alpha d_e^2(\gamma,\gamma')} \right]^{\rho}} < 1. \qquad (1)$$

2.3 Step 3: Configuration counting

Equation (1) contains an average over all codes of a function $T_{\alpha,\rho}$, where $T_{\alpha,\rho}$ depends on the codewords of a *code*. Shannon's random coding technique switches these two operations and obtains an average over all codewords of another function $\hat{T}_{\alpha,\rho}$, where $\hat{T}_{\alpha,\rho}$ is an average over all the codes (it can be viewed as the probability that a codeword is chosen). Here, the codewords are the paths through the trellis. "Configuration counting" determines the number of occurrences of a path in all the codes. This enables us to calculate $\hat{T}_{\alpha,\rho}$.

2.4 Step 4: Reduce the above bound to a usable form (Gallager's Lemma)

Step 4 (see Gallager [8]) puts the bound in its final form.

3 STATEMENT OF THE BOUND

Given a signal constellation S, there exists a (k, ν_0) trellis code with minimum Euclidean distance d_{free} such that:

$$d_{free}^2 \geq \max_{\substack{E(\alpha,\rho) > \ln 2 \\ 0 \leq \alpha \\ 0 \leq \rho \leq 1 \\ p(s)}} \left[\nu_0 \frac{E(\alpha, \rho)}{\alpha} - \frac{F[\, E(\alpha, \rho)\,]}{\alpha} \right], \tag{2}$$

where

α and ρ are parameters that enable us to optimize the bound,

s and s' are signal points from S,

between s and s',

$p(s)$ is a probability distribution on S, and

$$E(\alpha,\rho) = -\ln\left[\sum_{s \in S} p(s) \left[\sum_{s' \in S} p(s')\, e^{-\alpha\, d_e^2(s,\, s')} \right]^{\rho} \right]^{\frac{1}{\rho k}} .$$

4 USING THE BOUND TO EVALUATE TCM SCHEMES

4.1 d_{free} viewed as a function of ν_0:

The lower bound on d_{free} increases with the constraint length ν_0. Bounds on the free Hamming distance of binary convolutional codes have usually been expressed as functions of ν_0. This leads to asymptotic, exponentially tight bounds on $\max(d_{free})$ for large ν_0, but the bounds are not tight for small constraint lengths. In the following paragraphs we compare constellations at a given rate R of information bits per dimension and at a given average energy E_{avg} per dimension.

4.1.1 The effect of increasing the number M of signals on $\max(d_{free})$ (same rate R, same dimensionality, same average energy E_{avg}):

Increasing the number M of signals increases $\max(d_{free})$ because it provides more flexibility in the mapping of signals to trellis branches. In particular, this flexibility generates some codes with larger distances between correct and incorrect paths than codes generated with fewer signals. Note that increasing M, while keeping k constant, lowers the code rate $R_c \equiv \frac{k}{\log_2 M}$.

4.1.2 The effect of increasing the signal set dimensionality on $\max(d_{free})$ (same rate R, same number of signals M, same average energy E_{avg}):

Increasing the signal set dimensionality increases $\max(d_{free})$. We obtain higher dimensional schemes by using a basic one- or two-dimensional constellation for several transmission intervals [4]. Fig.1 shows an example of an L-dimensional scheme (L uses of 4-PAM, L=1,2,3, and 4). For a given constraint length ν_0, higher dimensionality yields a larger $\max(d_{free})$, and as ν_0 increases the curves diverge and the gain goes to infinity. However, the error coefficient increases with dimensionality [10], which may cancel the gain from the increased d_{free} at moderate decoded bit error rates.

4.1.3 The effect of changing the modulation scheme on d_{free} (same dimensionality, same number of signals M, same average energy E_{avg}, same rate R):

Changing the modulation scheme affects $\max(d_{free})$. For example, rectangular constellations yield a larger $\max(d_{free})$ than constant envelope constellations. Fig.2 shows M-PSK vs. M-QASK, and suggests that a d_{free} penalty is associated with constant envelope modulation.

4.1.4 The effect of uncoded bits or parallel transitions on $\max(d_{free})$ when the signal constellation is "mapped by set partitioning":

Uncoded bits generate parallel transitions in the trellis and limit $\max(d_{free})$, since d_{free} cannot be larger than the minimum distance between parallel transitions (the minimum distance within the subsets created by "set partitioning"). Fig.3 shows that for 8-PSK, 1 uncoded bit increases d_{free} for small constraint lengths. Although the bound is not tight, its slope agrees with the best known codes constructed [3]. For large ν_0, we see that 1 uncoded bit limits the achievable d_{free}.

4.2 Comparison of the lower bound on $\max(d_{free})$ with an upper bound [5,6] and known codes [3]:

We expect upper bounds to be tighter than our lower bound for small constraint lengths, and our lower bound to be tighter for large constraint lengths (this is analogous to the way bounds on binary convolutional codes behave). The lower bound is a "Chernoff type" bound and therefore is exponentially tight when ν_0 is large. Fig.4 shows that the upper bound is tighter than the lower bound for small constraint lengths. But the lower bound becomes tighter as ν_0 increases. The slope of the lower bound gives a precise indication of the asymptotic rate of increase in $\max(d_{free})$. Finally, the lower bound guarantees the existence of codes that can achieve a certain d_{free}.

5 CONCLUSION

We have obtained a random coding lower bound on $\max(d_{free})$. It takes into account the actual Euclidean distances between points in the channel signal constellation. As with any "Gilbert" bound, it is most useful as an asymptotic bound. It provides a means of comparing the asymptotic performance of different modulation schemes, and proves the existence of codes that achieve d_{free} larger than the bound.

REFERENCES

[1] D. J. Costello, Jr., "Free Distance Bounds for Convolutional Codes", IEEE Trans. Inform. Theory, vol. IT-20, pp. 356-365, May 1974.

[2] G. D. Forney, Jr., "Convolutional Codes II: Maximum Likelihood Decoding", Informa- tion and Control , vol. 25, pp. 222-266, July 1974.

[3] G. Ungerboeck, "Channel Coding with Multilevel Phase Signals", IEEE Trans. Inform. Theory, vol. IT-28, pp. 55-67, Jan. 1982.

[4] A. Lafanechere and D. J. Costello, Jr., "Multidimensional Coded PSK Systems Using Unit-Memory Trellis Codes", Proceedings Allerton Conf. on Communication, Control, and Computing, pp. 428-429, Monticello, IL, Sept. 1985.

[5] A. R. Calderbank, J. E. Mazo, and V. K. Wei, "Asymptotic Upper Bounds on the Minimum Distance of Trellis Codes", IEEE Trans. Commun. , vol. COM-33, no. 4, pp. 305-310, April 1985.

[6] A. R. Calderbank, J. E. Mazo, and H. M. Shapiro, "Upper Bounds on the Minimum Distance of Trellis Codes", Bell Syst. Tech. J., vol. 62, pp. 2617-2646, Oct. 1983, part I.

[7] H. Chernoff, "A Measure of Asymptotic Efficiency for Tests of a Hypothesis Based on a Sum of Observations," Ann. Math. Stat. , vol. 23, pp. 493-507, 1952.

[8] R. G. Gallager, "A simple Derivation of the Coding Theorem and Some Applications," IEEE Trans. Inform. Theory , vol. IT-11, pp. 3-18, Jan. 1965.

[9] A. J. Viterbi, "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm," IEEE Trans. Inform. Theory , vol. IT-13, pp. 260-269, Apr. 1967.

[10] G. D. Forney, Jr., R. G. Gallager, G. R. Lang, F. M. Longstaff, and S. U. Qureshi, "Efficient Modulation for Band-limited Channels", IEEE J. on Selected

Areas in Com- munication , vol. SAC-2, no. 5, pp. 632-647, Sept. 1984.

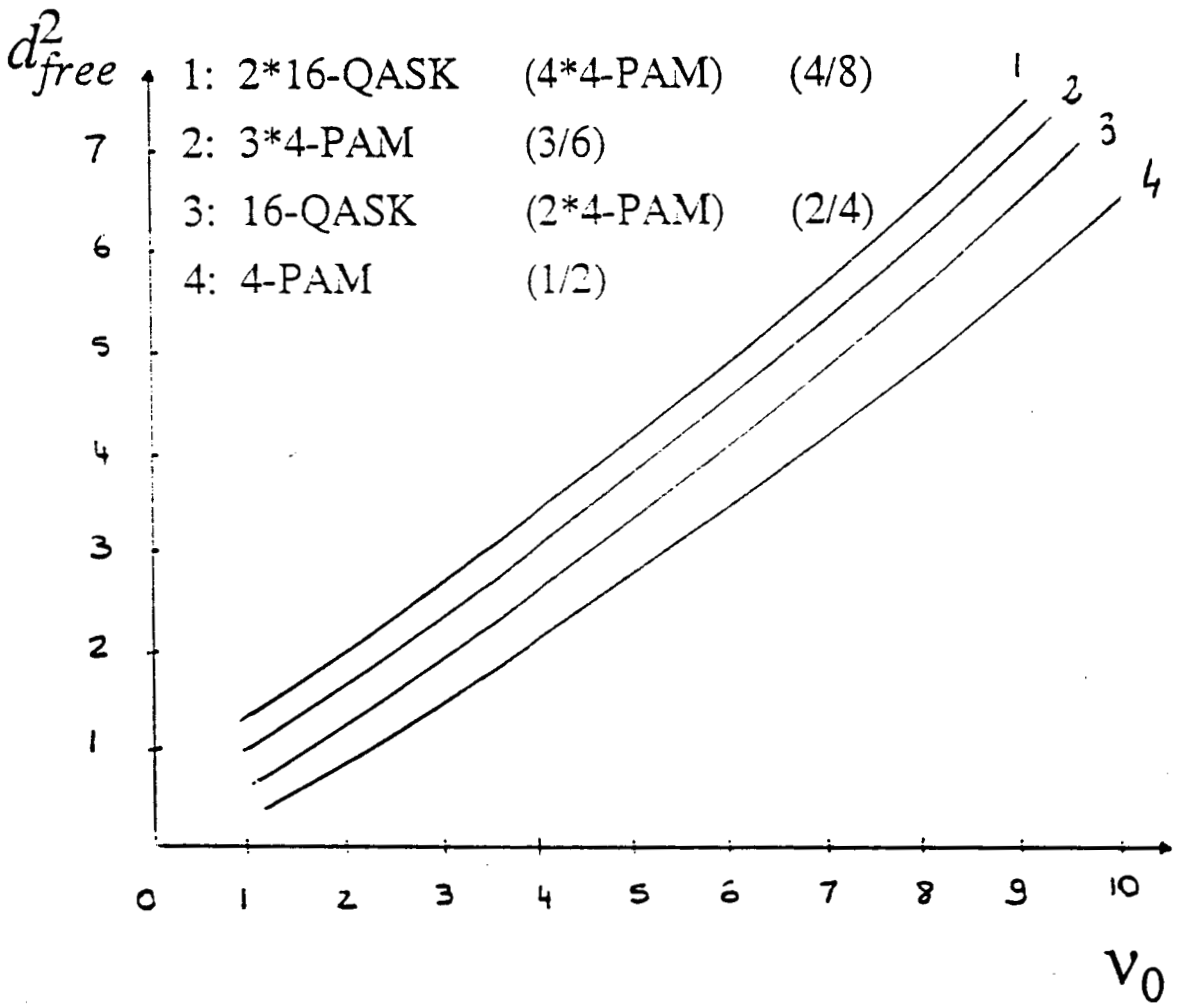


Fig. 1: Comparison of schemes with different dimensionalities, $R = 1 bit/dim$, $E_{avg} = 1$.

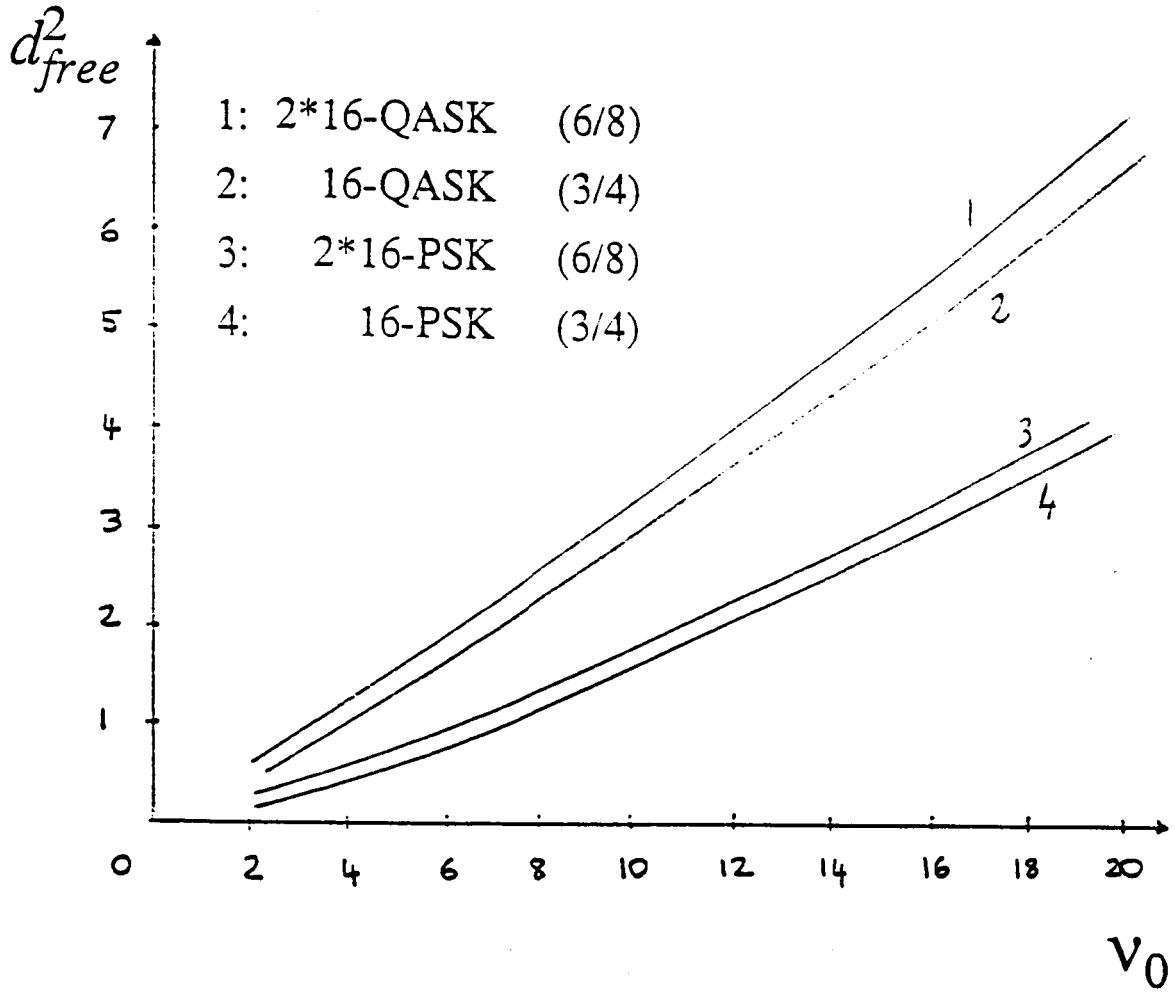


Fig. 2: Comparison of M-PSK, M-QASK, L*M-PSK and L*M-QASK schemes, $R = 1.5$ *bit/dim*, $E_{avg} = 1$.

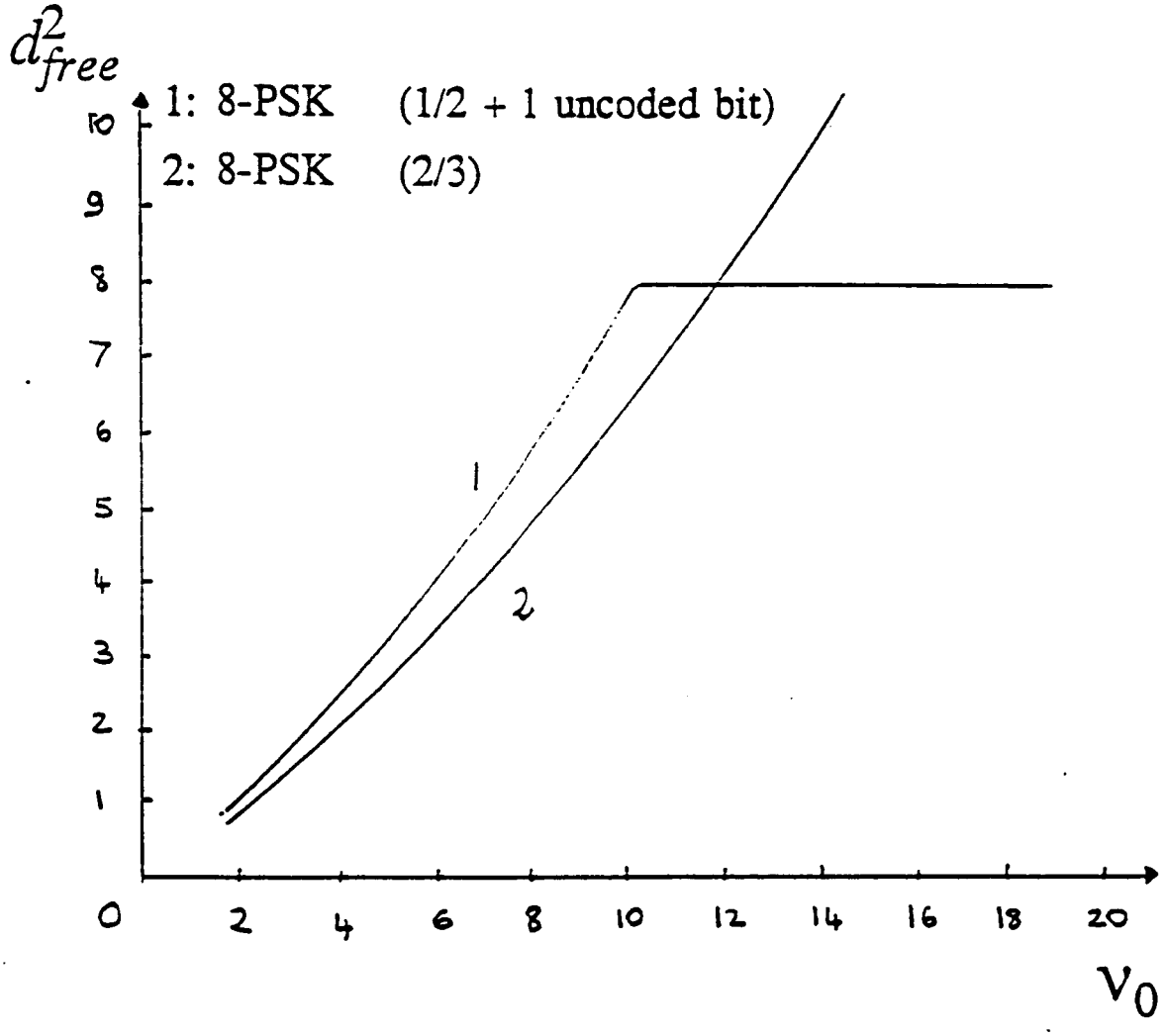


Fig. 3: Trellis coding and uncoded bits (Ungerboeck) for 8-PSK. $R = 1$ *bit/dim*, $E_{avg} = 1$.

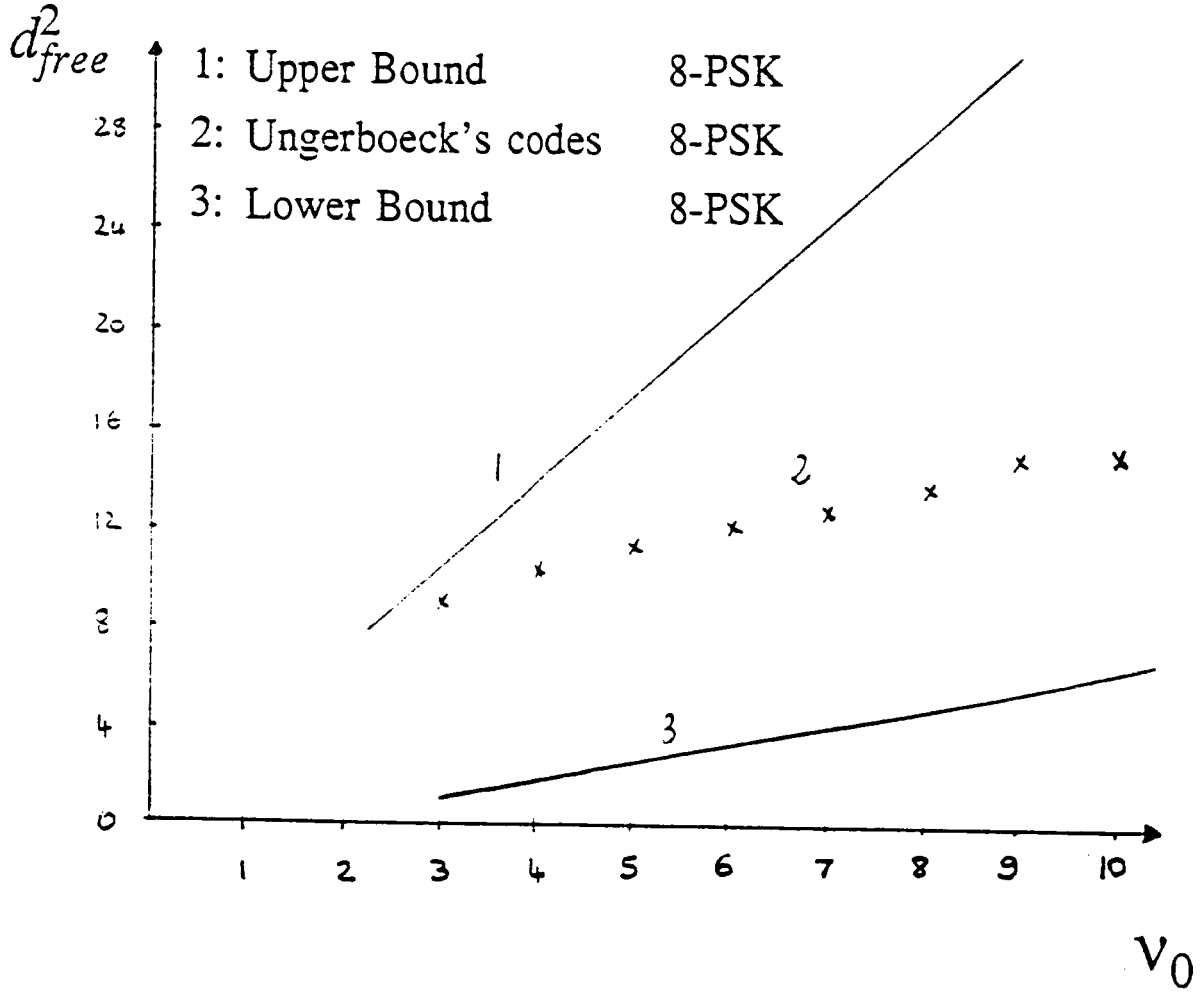


Fig. **4** : Comparison of the lower bound on d^2_{free} with upper bounds (Calderbank, Mazo, and Wei).

Appendix C

Multidimensional Trellis Coded Phase Modulation Using Unit-Memory and Partial-Unit-Memory Convolutional Codes

Multidimensional Trellis Coded Phase Modulation Using Unit-Memory and Partial-Unit-Memory Convolutional Codes

*Alain LaFanchere**

and
Robert H. Deng
Daniel J. Costello, Jr.

Dept. of Electrical and Computer Engineering
University of Notre Dame
Notre Dame, IN 46556

Submitted to
IEEE Transactions on Information Theory

ABSTRACT

In this paper, we investigate the use of unit-memory and partial-unit-memory trellis codes along with multi-dimensional MPSK modulation. A $2L$-dimensional ($L \geq 2$) MPSK signal set is obtained by forming the Cartesian product of L 2-dimensional MPSK signal sets.

A systematic approach to partitioning the signal set is introduced, followed by the trellis code construction and a discussion of encoder implementation. Codes with coding gains of up to 1.9 dB over those achievable with trellis coded 2-dimensional modulation are obtained.

January 16, 1987

This work was supported by NASA Grant NAG5-557 and by NSF Grant ECS84-14608.

* Formerly with the Department of Electrical and Computer Engineering, Illinois Institute of Technology, Chicago, IL.

Multidimensional Trellis Coded Phase Modulation Using Unit-Memory and Partial-Unit-Memory Convolutional Codes

*Alain LaFanchere**

and
Robert H. Deng
Daniel J. Costello, Jr.

Dept. of Electrical and Computer Engineering
University of Notre Dame
Notre Dame, IN 46556

Submitted to
IEEE Transactions on Information Theory

I. Introduction

Since the publication of the paper by Ungerboeck [1], trellis coded modulation (TCM) has become a very active research area [2-8]. The basic idea of TCM is that by trellis coding onto an expanded signal set (relative to that needed for uncoded transmission), both power and bandwidth efficient communication can be achieved.

TCM can be classified into two basic types, the lattice-type (e.g., M-AM, M-QASK) and the constant-envelope-type (e.g., MPSK). The latter has a slightly lower power efficiency compared with the former but is more suitable for nonlinear band-limited channels. In any TCM design, partitioning of the signal set into subsets with increasing intra-subset distances plays a central role. It defines the signal mapping used by the modulator, and provides a tight bound on the Euclidean distance (ED), which permits an efficient search for optimum codes. For lattice-type TCM, Calderbank and Sloane [3] have made the important observation that partitioning the signal set into subsets corresponds to partitioning a lattice into a sublattice and its cosets.

This work was supported by NASA Grant NAG5-557 and by NSF Grant ECS84-14608.

* Formerly with the Department of Electrical and Computer Engineering, Illinois Institute of Technology, Chicago, IL.

Forney [4] has developed a method, called the "squaring construction", of partitioning higher dimensional lattices from a lower dimensional lattice by using a coset code. However, his approach does not apply to constant- envelope-type TCM.

In this paper, we investigate unit - memory (UM) and partial - unit - memory (PUM) trellis coded $2L$-dimensional ($L \geq 2$) QPSK (or L*QPSK) and 8PSK (or L*8PSK) modulation. The L*MPSK signal set is generated simply by repeating an MPSK signal set L times. Therefore, the $2L$-dimensional MPSK signal set S_{2L} is the Cartesian product of L 2-dimensional MPSK signal sets S_2, i.e., $S_{2L} = S_2 \times S_2 \times \cdots \times S_2$ (L times). S_{2L} has some interesting features which are not found in S_2. Codes with higher effective information rates and larger coding gains are possible due to the increased flexibility of coding in S_{2L}.

In section II, we give a brief description of the structure of a trellis encoder and of the means of measuring trellis code performance. Section III presents results for UM/PUM trellis coded $2L$-dimensional QPSK modulation. In this case, the Hamming distance (HD) is proportional to the squared Euclidean distance (ED^2), and the TCM design is equivalent to the design of a conventional binary convolutional code.

For 8PSK modulation, HD and ED^2 are no longer proportional and Ungerboeck's concept of "mapping by set partitioning" must be invoked. In section IV, set partitioning of the $2L$-dimensional 8PSK signal set into subsets is shown to be equivalent to partitioning the L-dimensional binary vector space $\mathbf{C} = \{0, 1\}^L$ into a subcode and its cosets. As examples, we give the signal set partitions for 2, 4, and 8-dimensional 8PSK modulation. Some trellis coded multi-dimensional 8PSK modulation design examples are given in section V. In section VI, we illustrate the procedure for implementing the TCM designs through the use of binary convolutional encoders. The UM/PUM trellis coded 2, 4, and 8-dimentional 8PSK modulation designs are specified by listing the code generator matrices.

II General Description of Trellis Coded Modulation

The first general description of TCM was given by Forney et. al. [2] and is shown in Figure 1. The $k = k_1 + k_2$ information bits are split into a k_1-tuple and a k_2-tuple. The k_1-tuple is encoded by a binary encoder of constraint length ν, resulting in n_1 coded bits. The n_1 bits are then used to choose which subsets are to be used for

each symbol. The remaining k_2-tuple is not coded but merely selects points from the chosen subsets, with the signal set being large enough to accommodate all incoming bits. Thus, the implementation of the TCM scheme is reduced to synthesizing the (n_1, k_1, ν) binary encoder.

For the trellis coded multi-dimensional MPSK modulation discussed in this paper, the subset is selected jointly by the n_1 output bits and some of the k_2 uncoded bits, as indicated by the dotted connection in Figure 1, while the signal points are selected only by the k_2 uncoded bits. Hence, in the implementation of trellis coded multi-dimensional MPSK modulation scheme, the (n, k, ν) convolutional encoder outlined by the dotted box, with $n = n_1 + k_2$ and $k = k_1 + k_2$, must be considered.

Once the constraint length ν and the set partitioning is determined, the free ED^2, denoted d_f^2, of the TCM scheme is uniquely determined by the trellis structure and the number of parallel transitions between trellis states. The number of parallel transitions equals 2^{k_2}, which also equals the number of signal points within each subset. The intra-set ED^2 of the subset places an upper bound on d_f^2. Generally speaking, the fewer the number of points within the subset, the larger the intra-set ED^2. Therefore, reducing the number of uncoded bits k_2 can result in a larger d_f^2. However, this may also lead to increasing the constraint length ν, and hence the decoder complexity, in order to reduce the connectivity of the trellis.

The performance of TCM is measured in terms of its effective information rate R_{eff} (bits/dimension) and its error probability. For rate $R = k/n$ coded $2L$-dimensional MPSK modulation,

$$R_{eff} = k/2L \text{ bits/dimension.} \tag{1}$$

Let d_f^2 be the free ED^2 of the code normalized by the symbol energy E_s at the modulator output. At high signal-to-noise ratios, an asymptotically tight expression for the event error probability P(E) is

$$P(E) \simeq M(d_f^2)\, Q\left[\sqrt{\frac{d_f^2 E_s}{2N_o}}\right],$$

where $M(d_f^2)$ is the multiplicity of the error events with squared distance d_f^2 from the correct path and N_o is the single-sided noise power spectral density. Since $LE_s = kE_b$,

where E_b is the energy per information bit, we have

$$P(E) \simeq M(d_f^2)\, Q\left[\sqrt{\frac{d_f^2\, E_b\, k}{2N_o\, L}}\right]$$

$$= M(d_f^2)\, Q\left[\sqrt{\frac{d_f^2\, R_{eff}\, E_b}{N_o}}\right]. \tag{2}$$

Let d_{ref}^2 and R_{ref} be the normalized free ED^2 and the effective information rate of the reference system, respectively. The coding gain γ over the reference system, from (2), is defined as

$$\gamma \equiv 10 \log_{10} \frac{d_f^2\, R_{eff}}{d_{ref}^2\, R_{ref}}\ dB. \tag{3}$$

For example, for QPSK modulation, $d_{ref}^2 = 2$ and $R_{ref} = 1$ bit/dimension; for 8PSK modulation, $d_{ref}^2 = 2-\sqrt{2} = 0.586$ and $R_{ref} = 1.5$ bits/dimension, and the coding gains are

$$\gamma_{QPSK} = 10\log_{10} \frac{d_f^2\, R_{eff}}{2}\ dB, \tag{4}$$

and

$$\gamma_{8PSK} = 10 \log_{10} \frac{d_f^2\, Reff}{0.879}\ dB, \tag{5}$$

respectively.

III UM and PUM Trellis Coded L*QPSK Modulation

Combining L QPSK signals to obtain a modulation signal in $2L$ dimensions is achieved by using a QPSK modulator L times, i.e., by time-sharing the channel. This section is devoted to a discussion of the coding gains achievable with $2L$-dimensional QPSK signals combined with UM and PUM convolutional codes.

Figure 2 illustrates the 2-dimensional QPSK signal set along with a commonly used two bit mapping. By combining L of these signal sets, we create a set of 4^L signals in $2L$ dimensions, and each signal consists of L QPSK signals. Let a and b be two signals in the $2L$-dimensional space. Let $\underline{V}$ and $\underline{W}$ be the two $2L$-dimensional

binary representations associated with a and b, respectively. Let $HD(\underline{V},\underline{W})$ denote the HD between $\underline{V}$ and $\underline{W}$, and $ED^2(a,b)$ denote the ED^2 between a and b. It can be seen from Figure 2 that

$$ED^2(a,b) = 2 \cdot HD(\underline{V}, \underline{W}).$$

Consequently, maximizing the free ED^2 is equivalent to maximizing the free HD, and coded $2L$-dimensional QPSK modulation reduces to conventional binary convolutional code design.

In most applications of (n, k, ν) convolutional codes, k and n are taken to be relatively prime, i.e., $GCD(n, k) = 1$. However, codes with $GCD(n, k) > 1$ can sometimes achieve a larger free HD for a given state complexity due to the increased flexibility available when coding more bits at the same time [9,10]. We summarize in Table 1 the gains achievable by taking $GCD(n, k) > 1$, where the reference system is the best code with $GCD(n, k) = 1$ and with the same state complexity. The codes in Table 1 are taken from [9, 10].

It is seen that two of these gains (0.67dB and 0.62dB) are obtained for odd n. These codes do not fit the L*QPSK signals we propose, unless we allow some time delay and links between two successive sets of output bits. All other gains correspond to even n. Note in particular the 1.25 dB gain obtained for encoder constraint lengths $\nu = 1$ and 3 by using rate 2/4 and 4/8 codes instead of the usual rate 1/2 code. These results encourage us to apply these codes to L*8PSK modulation.

IV Partitioning the 2L-Dimensional 8PSK Signal Space

The remainder of the paper deals with multi-dimensional 8PSK signals. In this section, we present a method for partitioning the $2L$-dimensional 8PSK signal space into subsets. This approach simplifies both the construction of trellis coded multi-dimensional 8PSK modulation and the corresponding binary convolutional encoder design. The set partitioning of the $2L$-dimensional 8PSK signal space into subsets can be described as partitioning an L-dimensional binary vector space into a subspace (or a linear block code) and its cosets. Therefore, we first examine some of the properties of partitioning an L-dimensional binary vector space.

A. Partitioning a binary vector space

The L-dimensional binary vector space is given by $\mathbf{C} = \{0,1\}^L$. Let $\mathbf{C}_i$ be an $(L, L-i)$, $1 \le i \le L$, binary linear block code specified by the $(L-i) \times L$ generator matrix $\mathbf{G}_i$. The 2^i-way binary vector space partition, $\mathbf{C}/\mathbf{C}_i$, divides $\mathbf{C}$ into $\mathbf{C}_i$ and its $2^i - 1$ cosets, $\mathbf{T}_1(i), \mathbf{T}_2(i), \ldots, \mathbf{T}_{2^i-1}(i)$. Define $|\mathbf{C}/\mathbf{C}_i|$ as the set of coset leaders for $\mathbf{C}_i$, $\mathbf{T}_1(i), \mathbf{T}_2(i), \ldots, \mathbf{T}_{2^i-1}(i)$, i.e.,

$$|\mathbf{C}/\mathbf{C}_i| = \{\mathbf{t}_0, \mathbf{t}_1(i), \mathbf{t}_2(i), \ldots, \mathbf{t}_{2^i-1}(i)\}, \tag{6}$$

where $\mathbf{t}_0$, the all-zero vector, is the coset leader of $\mathbf{C}_i$, and $\mathbf{t}_j(i)$, $j = 1,2, \ldots, 2^i-1$, is the coset leader of $\mathbf{T}_j(i)$ [11]. The code $\mathbf{C}_i$ and its coset $\mathbf{T}_j(i)$, $j = 1,2, \ldots, 2^i-1$ are related by

$$\mathbf{T}_j(i) = \mathbf{C}_i + \mathbf{t}_j(i). \tag{7}$$

From (6) and (7), we see that the 2^i-way partition $\mathbf{C}/\mathbf{C}_i$ can be expressed as

$$\mathbf{C} = \mathbf{C}_i + |\mathbf{C}/\mathbf{C}_i| = \{\mathbf{C}_i, \mathbf{T}_1(i), \mathbf{T}_2(i), \ldots, \mathbf{T}_{2^i-1}(i)\}. \tag{8}$$

Therefore, the partition $\mathbf{C}/\mathbf{C}_i$ can be determined from $\mathbf{C}_i$ and $|\mathbf{C}/\mathbf{C}_i|$.

Now we form an $(L, L-i-1)$ binary linear code $\mathbf{C}_{i+1}$ such that $\mathbf{C}_{i+1} \subset \mathbf{C}_i$, i.e., $\mathbf{C}_{i+1}$ is a subcode of $\mathbf{C}_i$. Then $\mathbf{C}_i/\mathbf{C}_{i+1}$ is a 2-way partition, and it divides $\mathbf{C}_i$ into $\mathbf{C}_{i+1}$ and its coset $\mathbf{T}_1(i+1)$. $\mathbf{C}/\mathbf{C}_{i+1}$ is a 2^{i+1}-way partition which partitions $\mathbf{C}$ into $\mathbf{C}_{i+1}$ and its $2^{i+1}-1$ cosets, $\mathbf{T}_1(i+1), \mathbf{T}_2(i+1), \ldots, \mathbf{T}_{2^{i+1}-1}(i+1)$. Therefore, $\mathbf{C}/\mathbf{C}_i/\mathbf{C}_{i+1}$ forms a 2^i/2-way partition chain. The relationships between the partitions $\mathbf{C}/\mathbf{C}_i$, $\mathbf{C}_i/\mathbf{C}_{i+1}$, $\mathbf{C}/\mathbf{C}_{i+1}$ and the partition chain $\mathbf{C}/\mathbf{C}_i/\mathbf{C}_{i+1}$ are depicted in Figure 3.

Let $|\mathbf{C}_i/\mathbf{C}_{i+1}|$ and $|\mathbf{C}/\mathbf{C}_{i+1}|$ be the two sets of coset leaders associated with the partitions $\mathbf{C}_i/\mathbf{C}_{i+1}$ and $\mathbf{C}/\mathbf{C}_{i+1}$, respectively. That is

$$|\mathbf{C}_i/\mathbf{C}_{i+1}| = \{\mathbf{t}_0, \mathbf{t}_1(i+1)\},$$

and

$$|\mathbf{C}/\mathbf{C}_{i+1}| = \{\mathbf{t}_0, \mathbf{t}_1(i+1), \mathbf{t}_2(i+1), \ldots, \mathbf{t}_{2^{i+1}-1}(i+1)\}.$$

$|\mathbf{C}/\mathbf{C}_{i+1}|$ can be found from $|\mathbf{C}/\mathbf{C}_i|$ and $|\mathbf{C}_i/\mathbf{C}_{i+1}|$ as follows:

$$\mathbf{t}_{2j}(i+1) = \mathbf{t}_j(i), \quad j = 1, \ldots, 2^i-1, \tag{9.1}$$

$$\mathbf{t}_{2j+1}(i+1) = \mathbf{t}_j(i) + \mathbf{t}_1(i+1), \quad j = 1, \ldots, 2^i-1. \tag{9.2}$$

Since

$$\mathbf{C} = \mathbf{C}_i + |\mathbf{C}/\mathbf{C}_i| = \{\mathbf{C}_i, \mathbf{T}_1(i), \mathbf{T}_2(i) , \ldots , \mathbf{T}_{2^i-1}(i)\},$$

$$\mathbf{C}_i = \mathbf{C}_{i+1} + |\mathbf{C}_i/\mathbf{C}_{i+1}| = \{\mathbf{C}_{i+1}, \mathbf{T}_1(i+1)\},$$

and

$$\mathbf{C} = \mathbf{C}_{i+1} + |\mathbf{C}/\mathbf{C}_{i+1}| = \{\mathbf{C}_{i+1}, \mathbf{T}_1(i+1), \mathbf{T}_2(i+1) , \ldots , \mathbf{T}_{2^{i+1}-1}(i+1)\},$$

knowing $|\mathbf{C}/\mathbf{C}_i|$, $\mathbf{C}_i$, and $\mathbf{C}_{i+1}$, we can completely determine the partition $\mathbf{C}/\mathbf{C}_{i+1}$ and the partition chain $\mathbf{C}/\mathbf{C}_i/\mathbf{C}_{i+1}$. In general, the 2^{i+1}-way partition $\mathbf{C}/\mathbf{C}_{i+1}$ and the partition chain $\mathbf{C}/\mathbf{C}_1/\mathbf{C}_2/ \cdots /\mathbf{C}_i/\mathbf{C}_{i+1}$ can be determined from $\mathbf{C}_1, \mathbf{C}_2 , \ldots , \mathbf{C}_i$, and $\mathbf{C}_{i+1}$, where $\mathbf{C}_{i+1} \subset \mathbf{C}_i \subset \cdots \subset \mathbf{C}_2 \subset \mathbf{C}_1$.

Let $HD_{\min}(\mathbf{C}_i)$ be the minimum HD of $\mathbf{C}_i$. Then we have $HD_{\min}(\mathbf{C}_i) = HD_{\min}(\mathbf{T}_1(i)) = \cdots = HD_{\min}(\mathbf{T}_{2^i-1}(i))$. In what follows, we will relate the binary vector space partition with the $2L$-dimensional 8PSK signal space partition.

B. Partitioning the two-dimensional 8PSK signal space

The 2-dimensional 8PSK signal space is shown in Figure 4, along with a 3 bit binary mapping (i_1, i_2, i_3) for each signal point. The optimal partition of the 2-dimensional 8PSK signal space $S_2 = (0, 1, 2, 3, 4, 5, 6, 7)$, given by Ungerboeck [1], is illustrated by the partition tree shown in Figure 5, where

$$P_0 = \{S_2 \mid i_3 = 0\} = \{Q_0 = (P_0 \mid i_2 = 0), Q_1 = (P_0 \mid i_2 = 1)\},$$

i.e.,

$$P_0 = (0, 2, 4, 6), \ Q_0 = (0, 4), \ Q_1 = (2, 6);$$

and

$$P_1 = \{S_2 \mid i_3 = 1\} = \{Q'_0 = (P_1 \mid i_2 = 0), Q'_1 = (P_1 \mid i_2 = 1)\},$$

i.e.,

$$P_1 = (1, 3, 5, 7), \ Q'_0 = (1, 5), \ Q'_1 = (3, 7).$$

Let $ED^2_{\min}(\cdot)$ denote the minimum intra-set ED^2. Then we see that

$$ED^2_{\min}(S_2) = \Delta_0^2 = 2-\sqrt{2},$$

$$ED^2_{\min}(P_0) = ED^2_{\min}(P_1) = \Delta_1^2 = 2,$$

$$ED^2_{\min}(Q_i) = ED^2_{\min}(Q'_i) = 2\Delta_1^2 = 4, \quad i = 0, 1.$$

We also note that the minimum ED^2 between two subsets is related to the minimum HD of their respective subscripts by

$$ED^2_{\min}(P_i , P_j) = HD_{\min}(i, j)\cdot\Delta_0^2, \tag{10.1}$$

$$ED^2_{\min}(Q_i , Q_j) = HD_{\min}(i, j)\cdot\Delta_1^2, \tag{10.2}$$

$$ED^2_{\min}(Q'_i , Q'_j) = HD_{\min}(i, j)\cdot\Delta_1^2, \tag{10.3}$$

for $i, j = 0, 1$.

The above observation is very important and forms the basis for partitioning the $2L$-dimensional 8PSK signal space. It enables us to relate the signal space partitioning into subsets with the partitioning of a binary vector space into a subspace (subcode) and its cosets.

C. Partitioning the four-dimensional 8PSK signal space

The 4-dimensional 8PSK signal space is formed from the Cartesian product of two 2-dimensional 8PSK signal spaces, i.e., $S_4 = S_2 \times S_2 = (P_0 , P_1)^2$.

The first level partition. First, we partition S_4 into two subsets A_1 and A_2 such that their minimum intra-subset ED^2 is maximized. S_4 can be rewritten as

$$S_4 = (P_0, P_1)^2 = (P_0 \times P_0, P_0 \times P_1, P_1 \times P_0, P_1 \times P_1). \tag{11}$$

It is easy to see that we should choose

$$A_1 = (P_0 \times P_0, P_1 \times P_1), \tag{12}$$

$$A_2 = (P_0 \times P_1, P_1 \times P_0), \tag{13}$$

and from (10.1)

$$\begin{aligned} ED^2_{\min}(A_1) &= \min \{ED^2_{\min}(P_0), ED^2_{\min}(P_1), HD_{\min}(00,11)\cdot\Delta_0^2\} \\ &= HD_{\min}(00, 11)\cdot\Delta_0^2 = 2(2-\sqrt{2}). \end{aligned} \tag{14}$$

Similarly,

$$ED^2_{\min}(A_2) = HD_{\min}(01, 10)\cdot\Delta_0^2 = 2(2-\sqrt{2}). \tag{15}$$

Now consider partitioning the 2-dimensional binary vector space $\mathbf{C} = \{0, 1\}^2$ into a (2, 1) linear block code $\mathbf{C}_1$ and its coset $\mathbf{T}_1(1)$. Let $\mathbf{C}_1$ be generated by the generator matrix $\mathbf{G}_1 = [1\ 1]$. The 2-way partition $\mathbf{C}/\mathbf{C}_1$ divides $\mathbf{C}$ into $\mathbf{C}_1$ and its coset $\mathbf{T}_1(1)$. That is

$$\mathbf{C} = \{0, 1\}^2 = \{00, 01, 10, 11\} = \{\mathbf{C}_1, \mathbf{T}_1(1)\}, \tag{16}$$

$$\mathbf{C}_1 = \{00, 11\}, \tag{17}$$

$$\mathbf{T}_1(1) = \mathbf{t}_1(1) + \mathbf{C}_1 = (01) + \mathbf{C}_1 = \{01, 10\}. \tag{18}$$

Obviously,

$$HD_{\min}(\mathbf{C}_1) = HD_{\min}(00, 11) = 2, \tag{19}$$

$$HD_{\min}(\mathbf{T}_1(1)) = HD_{\min}(01, 10) = 2. \tag{20}$$

Comparing (11) - (15) with (16) - (20), we see that the correspondence between the signal space partition and the binary vector space partition is one-to-one. Knowing $\mathbf{C}_1$ and $\mathbf{T}_1(1)$, the two signal subsets A_1 and A_2 can be obtained by mapping the elements of $\mathbf{C}_1$ and $\mathbf{T}_1(1)$ into the subscripts of $P_i \times P_j$, as indicated by (12) and (13). To simplify the notation, we denote the signal space partitioning at this level by $\mathbf{C}/\mathbf{C}_1$: $\mathbf{G}_1 = [1\ 1]$.

The second level partition. Now we partition A_1 and A_2 into subsets B_1, B_2, B_3, and B_4, respectively. Let $\mathbf{C}_2$ be a (2, 0) linear code, i.e., $\mathbf{C}_2 = \{00\}$. The 2-way partition $\mathbf{C}_1/\mathbf{C}_2$ divides $\mathbf{C}_1$ into $\mathbf{C}_2$ and its coset $\mathbf{T}_1(2)$. Therefore, $\mathbf{C}/\mathbf{C}_2$ is a 4-way partition which divides $\mathbf{C}$ into $\mathbf{C}_2$ and its three cosets,$\mathbf{T}_1(2)$, $\mathbf{T}_2(2)$, and $\mathbf{T}_3(2)$ and $\mathbf{C}/\mathbf{C}_1/\mathbf{C}_2$ forms a 2/2-way partition chain. The partition chain $\mathbf{C}/\mathbf{C}_1/\mathbf{C}_2$, along with the 2-way partitions $\mathbf{C}/\mathbf{C}_1$, $\mathbf{C}_1/\mathbf{C}_2$ and the 4-way partition $\mathbf{C}/\mathbf{C}_2$, is shown in Figure 6(a). Note that $\mathbf{C}_1 = \{\mathbf{C}_2, \mathbf{T}_1(2)\}$. $\mathbf{T}_1(1) = \{\mathbf{T}_2(2), \mathbf{T}_3(2)\}$, and $HD_{\min}(\mathbf{C}_2) = HD_{\min}(\mathbf{T}_j(2)) = \infty$, $j = 1, 2, 3$. Consequently, we have

$$A_1 = (B_1, B_2), \text{ where } B_1 = P_0 \times P_0,\ B_2 = P_1 \times P_1;$$

$$A_2 = (B_3, B_4), \text{ where } B_3 = P_0 \times P_1,\ B_4 = P_1 \times P_0;$$

and

$$ED^2_{\min}(B_j) = ED^2_{\min}(P_0) = ED^2_{\min}(P_1) = \Delta_1^2 = 2,\ j = 1, 2, 3, 4.$$

We denote the partitioning at this level by $\mathbf{C}/\mathbf{C}_2 : \mathbf{C}_2 = \{0\ 0\}$.

The third level partition. To partition the subsets at level two, we use the fact that $P_0 = (Q_0, Q_1)$ and $P_1 = (Q'_0, Q'_1)$. Then the subsets B_1, B_2, B_3, and B_4 can be rewritten as

$$B_1 = P_0 \times P_0 = (Q_0, Q_1) \times (Q_0, Q_1),$$

$$B_2 = P_1 \times P_1 = (Q'_0, Q'_1) \times (Q'_0, Q'_1),$$

$$B_3 = P_0 \times P_1 = (Q_0, Q_1) \times (Q'_0, Q'_1),$$

$$B_4 = P_1 \times P_0 = (Q'_0, Q'_1) \times (Q_0, Q_1).$$

Observing the similarity with the partition at level one, and using (10.2), B_1 can be partitioned into $B_1 = \{C_1, C_2\}$ according to the binary partition rule $\mathbf{C}/\mathbf{C}_1 : \mathbf{G}_1 = [1\ 1]$, where

$$C_1 = \{Q_0 \times Q_0, Q_1 \times Q_1\}, \quad C_2 = \{Q_0 \times Q_1, Q_1 \times Q_0\}.$$

Applying the same rule to B_2, B_3, and B_4, respectively, we obtain

$$B_2 = \{C_3, C_4\}, \quad C_3 = \{Q'_0 \times Q'_0, Q'_1 \times Q'_1\}, \quad C_4 = \{Q'_0 \times Q'_1, Q'_1 \times Q'_0\},$$

$$B_3 = \{C_5, C_6\}, \quad C_5 = \{Q_0 \times Q'_0, Q_1 \times Q'_1\}, \quad C_6 = \{Q_0 \times Q'_1, Q_1 \times Q'_0\},$$

$$B_4 = \{C_7, C_8\}, \quad C_7 = \{Q'_0 \times Q_0, Q'_1 \times Q_1\}, \quad C_8 = \{Q'_0 \times Q_1, Q'_1 \times Q_0\},$$

and

$$ED^2_{\min}(C_j) = HD_{\min}(\mathbf{C}_1)\cdot\Delta_1^2 = 2\Delta_1^2 = 4, \quad j = 1, 2, \ldots, 8.$$

Since the same binary partition rule $\mathbf{C}/\mathbf{C}_1 : \mathbf{G}_1 = [1\ 1]$ has been used four times, we denote the set partitioning at this level by $4 \times [\mathbf{C}/\mathbf{C}_3] : \mathbf{G}_3 = [1\ 1]$. (Even though $\mathbf{C}_3 = \mathbf{C}_1$, the subscript 3 is used here to indicate the partition level.)

The fourth level partition. Similar to the second level partition, using the rule $\mathbf{C}/\mathbf{C}_2 : \mathbf{C}_2 = \{0\ 0\}$, we can partition $B_1 = \{C_1, C_2\}$ into $D_1 = Q_0 \times Q_0$, $D_2 = Q_1 \times Q_1$, $D_3 = Q_0 \times Q_1$, and $D_4 = Q_1 \times Q_0$, with $C_1 = \{D_1, D_2\}$ and $C_2 = \{D_3, D_4\}$. Applying the same rule three more times on $B_2 = \{C_3, C_4\}$, $B_3 = \{C_5, C_6\}$, and $B_4 = \{C_7, C_8\}$, respectively, we obtain the subsets $D_5, D_6, \cdots, D_{16}$, and

$$ED^2_{\min}(D_j) = ED^2_{\min}(Q_0) = 2\Delta_1^2 = 4, \quad j = 1, 2, \ldots, 16.$$

The set partitioning at this level is denoted by $4 \times [\mathbf{C}/\mathbf{C}_4] : \mathbf{C}_4 = \{0\ 0\}$.

The set partitioning can be carried out further by splitting Q_0, Q_1, Q'_0 and Q'_1, respectively, into two parts (i.e., $Q_0 = (0, 4)$, etc.). But we will stop here since only the subsets obtained above will be used in the code constructions of section V. The 4-dimensional 8PSK signal space partition is summarized in Table 2, where we use $N(\cdot)$ to designate the number of signal points within a given subset. In Figure 6(b) we show the corresponding partition tree.

D. Partitioning the six-dimensional 8PSK signal space

The above technique can be generalized to partition higher-dimensional signal spaces. The 6-dimensional 8PSK signal space is given by $S_6 = S_2^3 = (P_0, P_1)^3$, and it can be partitioned according to the 3-dimensional binary vector space partition. Tables 3(a) and 3(b) show two 6-dimensional 8PSK signal space partitions. To read these tables, we only need to understand the corresponding binary vector space partition. We show this by going through the partition given in Table 3(a). From Table 3(a), we first find the binary linear block codes used in the binary vector space partitions:

$$\mathbf{C}_1 : (3, 2) \text{ code}, \quad \mathbf{G}_1 = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}, \quad HD_{\min}(\mathbf{C}_1) = 2;$$

$$\mathbf{C}_2 : (3, 1) \text{ code}, \quad \mathbf{G}_2 = [0\ 1\ 1], \quad HD_{\min}(\mathbf{C}_2) = 2;$$

$$\mathbf{C}_3 : (3, 0) \text{ code}, \quad \mathbf{C}_3 = \{0\ 0\ 0\}, \quad HD_{\min}(\mathbf{C}_3) = \infty;$$

and $\mathbf{C}_4 = \mathbf{C}_1$, $\mathbf{C}_5 = \mathbf{C}_2$. Let $\mathbf{C} = \{0, 1\}^3$, the 3-dimensional binary vector space. We can see that $\mathbf{C}_3 \subset \mathbf{C}_2 \subset \mathbf{C}_1 \subset \mathbf{C}$. Therefore, $\mathbf{C}/\mathbf{C}_1$, $\mathbf{C}_1/\mathbf{C}_2$, and $\mathbf{C}_2/\mathbf{C}_3$ are 2-way partitions, $\mathbf{C}/\mathbf{C}_2$ and $\mathbf{C}/\mathbf{C}_3$ are 4 and 8-way partitions, respectively, and $\mathbf{C}/\mathbf{C}_1/\mathbf{C}_2/\mathbf{C}_3$ is a 2/2/2-way partition chain. This partition chain is shown in the upper part of Figure 7. From (8) we see that, to determine $\mathbf{T}_j(i)$, $j = 1, 2, \ldots, 2^i-1$, we need to know $|\mathbf{C}/\mathbf{C}_i| = \{\mathbf{t}_0, \mathbf{t}_1(i), \ldots, \mathbf{t}_{2^i-1}(i)\}$, the set of coset leaders. Thus, from (9.1) and (9.2), knowing $\mathbf{C}_1, \mathbf{C}_2, \ldots, \mathbf{C}_i$, the set of coset leaders $|\mathbf{C}/\mathbf{C}_i|$, or equivalently $\mathbf{T}_1(i), \mathbf{T}_2(i), \ldots, \mathbf{T}_{2^i-1}(i)$, can be determined iteratively from $\mathbf{t}_0$ and $\mathbf{t}_1(1)$.

After finding $\mathbf{C}_i$, $\mathbf{T}_1(i), \ldots, \mathbf{T}_{2^i-1}(i)$, the signal subsets at partition level i can be found by mapping the codewords of $\mathbf{C}_i$, $\mathbf{T}_1(i), \ldots, \mathbf{T}_{2^i-1}(i)$ into the subscripts of $P_{j_1} \times P_{j_2} \times P_{j_3}$. For example, at partition level 3, we have

$$\mathbf{C}/\mathbf{C}_3: \ \mathbf{C}_3 = \{0\ 0\ 0\}, \ \mathbf{T}_1(3) = \{0\ 1\ 1\}, \ \mathbf{T}_2(3) = \{1\ 0\ 1\}, \ \mathbf{T}_3(3) = \{1\ 1\ 0\},$$

$$\mathbf{T}_4(3) = \{0\ 0\ 1\},\ \mathbf{T}_5(3) = \{0\ 1\ 0\},\ \mathbf{T}_6(3) = \{1\ 0\ 0\},\ \mathbf{T}_7(3) = \{1\ 1\ 1\}.$$

Then the eight signal subsets $C_1, C_2, \ldots, C_8$ are given by $C_1 = \{P_0 \times P_0 \times P_0\}$, $C_2 = \{P_0 \times P_1 \times P_1\}, \cdots$.

The minimum intra-subset ED^2 at level i equals

$$ED^2_{\min}(\cdot) = \min\{ED^2_{\min}(P_0), HD_{\min}(\mathbf{C}_i)\cdot\Delta_0^2\},\ i = 1, 2, 3.$$

Since $ED^2_{\min}(P_0) = 2 > HD_{\min}(\mathbf{C}_i)\ (2-\sqrt{2})$ for $i = 1, 2$, and $ED^2_{\min}(P_0) = 2 < HD_{\min}(\mathbf{C}_3)\ (2-\sqrt{2}) = \infty$, we have

$$ED^2_{\min}(A_j) = HD_{\min}(\mathbf{C}_1)\cdot\Delta_0^2 = 2(2-\sqrt{2}),\ j = 1, 2,$$

$$ED^2_{\min}(B_j) = HD_{\min}(\mathbf{C}_2)\cdot\Delta_0^2 = 2(2-\sqrt{2}),\ j = 1, 2, 3, 4,$$

and

$$ED^2_{\min}(C_j) = ED^2_{\min}(P_0) = 2,\ j = 1, 2, \ldots, 8.$$

At partition level 4, we express $C_1, C_2, \ldots, C_8$ in terms of Q_0, Q_1, or Q'_0, Q'_1, e.g., $C_1 = (Q_0, Q_1)^3$, $C_2 = (Q_0, Q_1) \times (Q'_0, Q'_1)^2, \cdots$. Now C_j, $j = 1, 2, \ldots, 8$, corresponds to $\mathbf{C} = \{0, 1\}^3$. The signal subsets at partition levels 4 and 5 can be found by applying the binary partitions $\mathbf{C}/\mathbf{C}_4$, and $\mathbf{C}/\mathbf{C}_5$ to every C_j, $j = 1, 2, \ldots, 8$, as indicated by the partition chain $\mathbf{C}/\mathbf{C}_4/\mathbf{C}_5$ shown in Figure 7. We can see from (10.2) and (10.3), that

$$ED^2_{\min}(D_j) = \min\{ED^2_{\min}(Q_0), HD_{\min}(\mathbf{C}_4)\cdot\Delta_1^2\}$$

$$= HD_{\min}(\mathbf{C}_4)\cdot\Delta_1^2 = 4,\ j = 1, 2, \ldots, 16,$$

$$ED^2_{\min}(E_j) = \min\{ED^2_{\min}(Q_0), HD_{\min}(\mathbf{C}_5)\cdot\Delta_1^2\}$$

$$= HD_{\min}(\mathbf{C}_5)\cdot\Delta_1^2 = 4,\ j = 1, 2, \ldots, 32.$$

E. Partitioning the eight-dimensional 8PSK signal space

The 8-dimensional 8PSK signal space is given by $S_8 = (P_0, P_1)^4$. The signal space partition is shown in Table 4, where the following binary codes are used:

$$\mathbf{C}_1 : (4, 3)\ \text{code},\ \mathbf{G}_1 = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix},\ HD_{\min}(\mathbf{C}_1) = 2;$$

$$\mathbf{C}_2 : (4, 2) \text{ code}, \quad \mathbf{G}_2 = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}, \quad HD_{\min}(\mathbf{C}_2) = 2;$$

$$\mathbf{C}_3 : (4, 1) \text{ code}, \quad \mathbf{G}_3 = [\ 1\ 1\ 1\ 1\ ], \quad HD_{\min}(\mathbf{C}_3) = 4;$$

$\mathbf{C}_4 = \mathbf{C}_5 = \mathbf{C}_1$, $\mathbf{C}_6 = \mathbf{C}_2$, and $\mathbf{C}_7 = \mathbf{C}_3$. The binary vector space partition rule is shown in Figure 8.

The signal subsets at partition level $i = 1, 2, 3$ are obtained by mapping the codewords of $\mathbf{C}_i$, or $\mathbf{T}_j(i)$, into the subscripts of $P_{j_1} \times P_{j_2} \times P_{j_3} \times P_{j_4}$. For instance, at partition level 3, the binary code $\mathbf{C}_3$ and its cosets are $\mathbf{C}_3 = \{0\ 0\ 0\ 0,\ 1\ 1\ 1\ 1\}$, $\mathbf{T}_1(3) = \{\ 0\ 1\ 0\ 1,\ 1\ 0\ 1\ 0\ \}, \cdots$. Then the corresponding signal subsets are $C_1 = \{P_0 \times P_0 \times P_0 \times P_0, P_1 \times P_1 \times P_1 \times P_1\}$, $C_2 = \{P_0 \times P_1 \times P_0 \times P_1, P_1 \times P_0 \times P_1 \times P_0\}, \cdots$.

At partition level 4, we have two options. In the first approach, the signal subsets can be obtained simply in terms of P_0 and P_1, e.g., $D_1 = P_0 \times P_0 \times P_0 \times P_0$, $D_2 = P_1 \times P_1 \times P_1 \times P_1, \cdots$. However, in this case $ED^2_{\min}(D_j) = \Delta_1^2 = 2$, $j = 1, 2, \ldots, 16$. In the second approach, we first express C_j, $j = 1, 2, \ldots, 8$, in terms of Q_0, Q_1, or Q'_0, Q'_1, e.g., $C_1 = \{(Q_0, Q_1)^4, (Q'_0, Q'_1)^4\}, \cdots$. Therefore, they correspond to the binary vector space $\{\mathbf{C}, \mathbf{C}\} \equiv 2\mathbf{C}$, as shown in Figure 8. In Figure 8, we perform the partition $2[\mathbf{C}/\mathbf{C}_4]$ (note that $\mathbf{C}_4 = \mathbf{C}_1$), which divides $2\mathbf{C}$ into $\{\mathbf{C}_1, \mathbf{C}_1\} \equiv 2\mathbf{C}_1$ and $\{\mathbf{T}_1(1), \mathbf{T}_1(1)\} \equiv 2\mathbf{T}_1(1)$. Then the signal subset D_j, $j = 1, 2, \ldots, 16$, is obtained by mapping the elements of $2\mathbf{C}_1$ or $2\mathbf{T}_1(1)$ into the subscripts of $Q_{j_1} \times Q_{j_2} \times Q_{j_3} \times Q_{j_4}$, $Q'_{j_1} \times Q'_{j_2} \times Q'_{j_3} \times Q'_{j_4}, \cdots$. In this case, $ED^2(D_j) = 4\Delta_0^2 = 4(2-\sqrt{2}) > \Delta_1^2 = 2$. The rest of the set partitioning can be seen from Table 4 and Figure 8.

The signal space partitions of Table 2, Tables 3(a) and 3(b), and Table 4 are not complete, i.e., they can be partitioned further in a similar way. However, only those subsets which are used in the code constructions to follow are included in these tables.

V. Trellis Coded Modulation Design

Having introduced the signal set partitioning in the last section, we now show how to construct TCM schemes with multi-dimensional 8PSK signal sets.

Trellis coded modulation has been mostly restricted to the case where the code rate $R = (n-1)/n$. In our code constructions, however, we remove this condition by

considering code rates $R = (3L-i)/3L$, $L = 2, 3, 4$, $i = 1, 2, \ldots,$ such that the effective information rate $R_{eff} \geq 1$ bit/dimension. The construction procedures are illustrated by three examples with 4, 6, and 8-dimensional 8PSK modulation, respectively. Several other codes are tabulated in the next section.

A. Two-state R = 5/6 trellis coded 4-dimensional 8 PSK modulation

The 64 4-dimensional 8PSK signals are partitioned according to Table 2. Since $R = 5/6$, there are $2^5 = 32$ transitions to and from each state. Because $\nu = 1$ (two states), the only possible trellis structure is shown in Figure 9. Each state has two sets of 16 parallel transitions leaving and entering it. The signal subsets B_1, B_2, B_3, and B_4 each contains 16 signal points. Therefore, they are assigned to the parallel transitions in the trellis, as shown in Figure 9. Since $ED^2_{\min}(B_1, B_2) = ED^2_{\min}(A_1) = 2\Delta_0^2$, the minimum ED^2 on the diverging branches is $2\Delta_0^2$. The minimum ED^2 on the remerging branches is $ED^2_{\min}(B_1, B_3) = ED^2_{\min}(S_2) = \Delta_0^2$. Thus a 2-level error event has a total distance $ED^2 = 3\Delta_0^2 = 1.757$. Since this is smaller than that of the one-level error events, $ED^2_{\min}(B_1) = \Delta_1^2 = 2$, and of all longer error events, we see that $d_f^2 = 3\Delta_0^2 = 1.757$.

Let $M(B_2)$ and $M(B_3)$ be the number of points in B_2 and B_3 with distances $2\Delta_0^2$ and Δ_0^2 from the zero point in B_1, respectively. Then the multiplicity of error events with distance d_f^2 from the all zero path is $M(d_f^2) = M(B_2) \cdot M(B_3)$. A close examination of B_2 and B_3 reveals that $M(B_2) = 4$ and $M(B_3) = 2$. Therefore, $M(d_f^2) = 4 \times 2 = 8$. The coding gains of this code over 8PSK and QPSK, from (4) and (5), are $\gamma_{8PSK} = 3.977dB$ and $\gamma_{QPSK} = 0.407\ dB$, respectively. The effective information rate is $R_{eff} = 5/4 = 1.25$ bits/dimension.

B. Four-state R = 8/9 trellis coded 6-dimensional 8PSK modulation

This time we have two different signal space partitions. First consider partition II given in Table 3(b). Since $R = 8/9$, there are $2^8 = 256$ transitions to and from each state. Because the trellis has four states, i.e., $\nu = 2$, two trellis structures must be examined, one with two sets of 128 parallel transitions and one with four sets of 64 parallel transitions. It turns out that the former is the best arrangement, with the subset assignments shown in Figure 10. There are two paths with $d_f^2 = 3\Delta_0^2 = 1.757$, one of one level, the other of three levels. These are shown by the highlighted transitions in

Figure 10. The coding gains are $\gamma_{8PSK} = 4.257\ dB$ and $\gamma_{QPSK} = 0.687\ dB$, respectively. We can show that $M(d_f^2) = M(B_1) + M(B_2) \times M(B_3) \times M(B_2) = 8 + 2 \times 2 \times 2 = 16$, and $R_{eff} = 8/6 = 1.33$ bits/dimension .

If partition I of Table 3(a) is used instead, the best trellis structure will be the one with four sets of 64 parallel transitions. It also achieves $d_f^2 = 3\Delta_0^2 = 1.757$, but with a larger path multiplicity $M(d_f^2) = 24$.

C. Eight-state Rate = 9/12 trellis coded 8-dimensional 8PSK modulation.

The 8-dimensional 8PSK signal space partition is given in Table 4. Since $R = 9/12$, there are $2^9 = 512$ transitions to and from each state and $\nu = 3$ implies that three trellis structures must be examined. After some trial-and-error, we arrived at the optimal structure and the subset labeling given in Figure 11. Each state has four sets of 128 parallel transitions branching from it. The minimum ED^2 is $d_f^2 = 2\Delta_1^2 = 4$, which is achieved with a one-level path, and all longer paths have ED^2 larger than 4. Thus $M(d_f^2) = M(E_1) = 28$, $\gamma_{8PSK} = 7.092\ dB$, $\gamma_{QPSK} = 3.522\ dB$, and $R_{eff} = 9/8 = 1.125$ bits/dimension. We note that this eight-state code has the same coding gain as eight-state $R = 2/3$ coded 8PSK [1], yet has 12.5 percent greater bandwidth efficiency.

The same procedure can be applied to construct codes with various rates and numbers of dimensions. However, we now turn to the encoder implementation in the next section.

VI Encoder Implementation

Once the trellis and the branch labeling is determined, the actual encoder can be synthesized according to the set partitioning and the general encoding scheme shown in Figure 1. We adopt the mapping shown in Figure 4. The procedure for synthesizing a binary convolutional encoder from the trellis will be illustrated by two examples.

Example 1:

We first determine the encoder implementation for two-state $R = 5/6$ trellis coded 2*8PSK modulation. The encoder accepts 5 bits and forms two 3-bit output symbols, (b_1, b_2, b_3) and (b_4, b_5, b_6). These two symbols are then used to serially drive the 8PSK modulator according to the mapping: (b_1, b_2, b_3) $\rightarrow$ (i_1, i_2, i_3),

(b_4, b_5, b_6) $\rightarrow$ (i_1, i_2, i_3). From the signal space partition of Table 2, we obtain the mapping from subsets B_1, B_2, B_3, and B_4 to the encoder output bits b_1, b_2, b_3, b_4, b_5, and b_6 shown in Table 5. For each subset, b_1, b_2, b_4, and b_5 can be either 0 or 1. This means these bits must be uncoded (the trellis of Figure 9 has 16 parallel transitions, which corresponds to 4 uncoded bits). Hence $k_2 = 4$, $k_1 = 1$, $n_1 = 2$ (see Figure 1). The values of b_3 and b_6 for each subset B_i are simply the binary (2, 0) code $\mathbf{C}_2$ and its cosets $\mathbf{T}_1(B)$, $\mathbf{T}_2(B)$, $\mathbf{T}_3(B)$, obtained from the binary vector space partition $\mathbf{C}/\mathbf{C}_2 : \mathbf{C}_2 = \{ 0\ 0 \}$ (see Table 2). These two bits are the $n_1 = 2$ output bits of the (2, 1, 1) encoder. Now the problem of synthesizing a (6, 5, 1) encoder is reduced to synthesizing a (2, 1, 1) encoder according to the two-state trellis shown in Figure 12(a). The (2, 1, 1) encoder is given in Figure 12(b) and the (6, 5, 1) encoder is shown in Figure 12(c). This code can be represented by the generator matrices

$$\mathbf{G}_0 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \quad \mathbf{G}_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

From the binary encoder shown in Figure 12(c), we see that the two coded bits b_3 and b_6 are used to select a subset, and the four uncoded bits b_1, b_2, b_4, and b_5 are used to select a signal point within the selected subset.

Example 2:

In this example, we show how to synthesize a (9, 8, 2) binary convulutional encoder for the four-state $R = 8/9$ trellis coded 3*8PSK modulation scheme in section V. The encoder has eight inputs and three 3-bit symbol outputs, (b_1, b_2, b_3), (b_4, b_5, b_6), and (b_7, b_8, b_9), to serially modulate the eight phases according to the mapping: (b_1, b_2, b_3) $\rightarrow$ (i_1, i_2, i_3), (b_4, b_5, b_6) $\rightarrow$ (i_1, i_2, i_3), and (b_7, b_8, b_9) $\rightarrow$ (i_1, i_2, i_3). Since the trellis of Figure 10 has 128 parallel transitions between states, there are seven uncoded bits, and $k_1 = 1$, $k_2 = 7$, and $n_1 = 2$.

The mapping from signal subsets B_1, B_2, B_3, and B_4 to the nine output bits b_1, b_2 , . . . , b_9 is shown in Table 6, which is obtained from the signal space partition of Table 3(b). From Table 6 we see that b_1, b_2, b_4, b_5, b_7, and b_8 can be either 0 or 1 for all four subsets, so that these six bits must be uncoded bits. We take b_3 to be the

7th uncoded bit. Note that b_3 is also used to select a subset. Now we only need to consider the two coded bits b_6 and b_9 and the uncoded bit b_3. The state transition $00 \rightarrow 00$ (Figure 10) corresponds to the subset B_1 and requires b_3, b_6, and b_9 to be either 000 or 111 (Table 6). To meet this requirement, we connect b_3 to b_6 and b_9, as shown in Figure 13(a). Note that the output (b_3, b_6, b_9) generates the (3, 1) binary linear code $\mathbf{C}_2$. The state transition $00 \rightarrow 10$ (Figure 10) corresponds to the subset B_2 and requires (b_3, b_6, b_9) to belong to $\mathbf{T}_1(B) = \mathbf{C}_2 + (\ 010\ ) = \{\ 010, 101\ \}$, where (010) is the coset leader of $\mathbf{T}_1(B)$. This is achieved by connecting point (1) in Figure 13(b) to b_6. Now consider the state transition $10 \rightarrow 01$ (Figure 10). This corresponds to the subset B_3 and to $\mathbf{T}_2(B) = \mathbf{C}_2 + (\ 001\ ) = \{001, 110\}$, where (001) is the coset leader of $\mathbf{T}_2(B)$. Therefore, we connect point (2) in Figure 13(c) to b_9. The state transition $10 \rightarrow 11$ corresponds to the subset B_4 and to $\mathbf{T}_3(B) = \mathbf{C}_2 + (011) = \mathbf{C}_2 + (010) + (001) = \{011, 100\}$. Since points (1) and (2) have been connected to b_6 and b_9, respectively, no more connections are made. The encoder design at this stage is given in Figure 13(d). The connections corresponding to the other state transitions can be assigned in the same way. The complete encoder (including the uncoded bits b_1, b_2, b_4, b_5, b_7, and b_8) is shown in Figure 13(e). The code generator matrices are

$$\mathbf{G}_0 = \begin{bmatrix} 1&0&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0&0 \\ 0&0&1&0&0&1&0&0&1 \\ 0&0&0&1&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0&0 \\ 0&0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&1&0 \end{bmatrix}, \quad \mathbf{G}_1 = \begin{bmatrix} 0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&1 \\ 0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0 \end{bmatrix},$$

$$\mathbf{G}_2 = \begin{bmatrix} 0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0 \end{bmatrix}.$$

It should be noted that, for a given TCM scheme, the binary convolutional encoder design is not unique. In Figure 14, we show another encoder for the same TCM scheme considered in Example 2.

Several rate R = 5/6 and 4/6 trellis coded 2*8PSK modulation designs are listed in Tables 7(a) and 7(b), respectively. Rate R = 8/9, 7/9, and 6/9 trellis coded 3*8PSK modulation designs are shown in Tables 8(a), 8(b), and 8(c), respectively. Rate R = 11/12, 10/12, 9/12, and 8/12 trellis coded 4*8PSK modulation designs are given in Tables 9(a), 9(b), 9(c), and 9(d), respectively. For each code rate and number of dimensions, only two, four, and eight-state codes have been constructed.

VII Discussion and Conclusions

From Tables 7(a) - 9(d), we see that trellis coded multi-dimensional 8PSK modulation provides a variety of effective information rates ranging from 1 bit/dimension to slightly less than 1.5 bits/dimension. These rates cannot be achieved by trellis coded 2-dimensional 8PSK modulation. The coding gains are also encouraging. The two-state rate R = 4/6, 6/9, and 8/12 codes, have coding gains over uncoded 8PSK of 5.57, 6.31, and 6.58 dB, respectively, while the coding gains over Ungerboeck's two-state R = 2/3 trellis coded 8PSK are 0.88, 1.62, and 1.89 dB. Note that Ungerboeck's four-state R = 2/3 trellis coded 8PSK, a remarkably good code, has a coding gain of 6.58 dB over uncoded 8PSK, the same as R = 8/12 trellis coded 4*8PSK with only two states.

The four-state and eight-state R = 4/6, 6/9, and 8/12 codes have the same or slightly smaller coding gains compared with the R = 2/3 codes. However, since the unit-memory and partial-unit-memory codes are byte oriented, they have potential advantages for applications such as concatenated coding [12]. Investigations into this application are currently in progress.

The rate R = 5/6, 8/9, 7/9, 11/12, 10/12, and 9/12 coded multi-dimensional 8PSK modulation designs have greater bandwidth efficiency than rate R = 2/3 coded 8PSK. As can be expected, they are less power efficient. Exceptions are the eight-state rate R = 7/9 and 9/12 codes. Their coding gains over uncoded 8PSK (6.98 and 7.09 dB, respectively) are almost equivalent to eight-state rate R = 2/3 trellis coded 8PSK (7.17 dB), yet they are 16.7% and 12.5% more bandwidth efficient, respectively.

Finally, we note that the two-state rate R = 5/6, 8/9, and 11/12 codes have coding gains over uncoded 8PSK of 3.98, 2.50, and 2.63 dB, respectively. Their effective information rates of 1.25, 1.33, and 1.375 bits/dimension are close to 1.5

bits/dimension, the uncoded 8PSK effective information rate. We now look at the coding gain achievable as the number of dimensions goes to infinity. For $2L$-dimensional 8PSK modulation, the signal space is given by $S_{2L} = (P_0, P_1)^L$. We partition S_{2L} into $S_{2L} = (A_1, A_2) = (B_1, B_2, B_3, B_4)$, with $A_1 = (B_1, B_2)$ and $A_2 = (B_3, B_4)$. For any $L \geq 2$, there exists an $(L, L-1)$ single-parity-check code $\mathbf{C}_1$ with $HD_{\min}(\mathbf{C}_1) = 2$. According to our set partitioning procedure, we see that $ED^2_{\min}(A_1) = ED^2_{\min}(A_2) = 2\Delta_0^2$. Also, since $B_1, B_2, B_3,$ and B_4 are subsets of A_1 and A_2, we have $ED^2_{\min}(B_1) \geq 2\Delta_0$, $i = 1, 2, 3, 4$.

Now consider two-state $R = (3L-1)/3L$ trellis coded $2L$-dimensional 8PSK modulation. The trellis structure and the branch labeling are shown in Figure 9. For such a trellis, the one-level ED^2 is $ED^2_{\min}(B_1) \geq 2\Delta_0^2$, the two-level ED^2 is $ED^2_{\min}(B_1, B_2) + ED^2_{\min}(B_1, B_3) = ED^2_{\min}(A_1) + ED^2_{\min}(S_{2L}) = 3\Delta_0^2$. Any longer path has ED^2 larger than $3\Delta_0^2$. Hence, $d_f^2 \geq 2\Delta_0^2$ and $R_{eff} = (3L-1)/2L$ bits/dimension. The coding gain over uncoded 8PSK is given by

$$\gamma_{8PSK} = 10 \log_{10} \frac{R_{eff}\, d_f^2}{1.5\Delta_0^2} \geq 10 \log_{10} \frac{2(3L-1)}{3L}\ dB.$$

Thus, as the number of dimensions goes to infinity, we have

$$\lim_{L \geq \infty} \gamma_{8PSK} \geq 3dB$$

and

$$\lim_{L \to \infty} R_{eff} = 1.5 \text{ bits/dimension.}$$

These results are somewhat surprising, for we achieve at least a $3dB$ coding gain with no bandwidth expansion with only two-state trellis coding! It should be noted, however, that as L goes to infinity, the path multiplicity also goes to infinity (although linearly, not exponentially).

REFERENCES

[1] G. Ungerboeck, "Channel Coding with Multilevel/Phase Signals", *IEEE Trans. Inform. Theory,* Vol. IT-28, pp. 55-67, January 1982.

[2] G. D. Forney, Jr., et. al., "Efficient Modulation for Band-Limited Channels", *IEEE J. Selected Areas Commun.,* Vol. SAC-2, pp. 659- 686, Sept. 1984.

[3] A. R. Calderbank and N. J. A. Sloane, "New Trellis Codes", *IEEE Trans. Inform. Theory,* to appear, 1987.

[4] G. D. Forney, Jr., "Coset Codes", submitted to *IEEE Trans. Inform. Theory,* 1986.

[5] A. Lafanechere and D. J. Costello, Jr., "Multidimensional Coded PSK Systems Using Unit-Memory Trellis Codes", Proc. Allerton Conf. on Commun., Cont., and Comput., pp. 428-429, Monticello, IL, Sept. 1985.

[6] S. G. Wilson, "Rate 5/6 Trellis-Coded 8-PSK", *IEEE Trans. Commun.,* Vol. COM-34, pp. 1045-1049, Oct. 1986.

[7] E. Biglieri, "High-Level Modulation and Coding for Nonlinear Satellite Channels", *IEEE Trans. Commun.,* Vol. COM-32, pp. 616-627, May 1984.

[8] L. - F. Wei, "Trellis-Coded Modulation with Multi-Dimensional Constellations", *IEEE Trans. Inform. Theory,* to appear, 1987.

[9] L. - N. Lee, "Short, Unit-Memory, Byte-Oriented, Binary Convolutional Codes Having Maximal Free Distance", *IEEE Trans. Inform. Theory,* Vol. IT-22, pp. 349-352, May 1976.

[10] G. S. Lauer, "Some Optimum Partial-Unit-Memory Convolutional Codes", *IEEE Trans. Inform. Theory,* Vol. IT-25, pp. 240-243, March 1979.

[11] S. Lin and D. J. Costello, Jr., *Error-Control Coding: Fundamentals and Applications,* Prentice-Hall, N. J., 1983.

[12] L.-N. Lee, "Concatenated Coding Systems Employing a Unit-Memory Convolutional Code and a Byte-Oriented Decoding Algorithm", *IEEE Trans. Commun.,* Vol. COM-25, pp. 1064-1074, Oct. 1977.

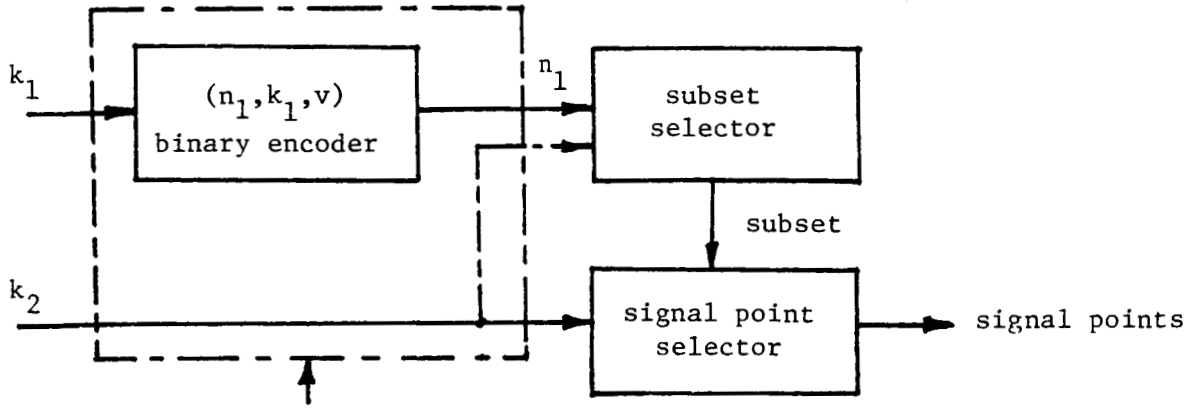


Fig. 1 General encoding scheme for trellis coded modulation

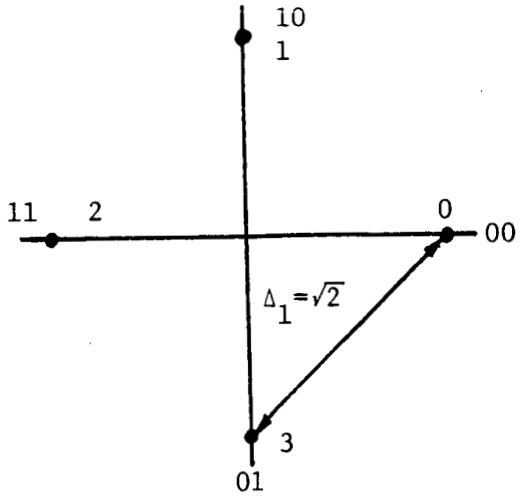


Fig. 2 QPSK signals and a mapping

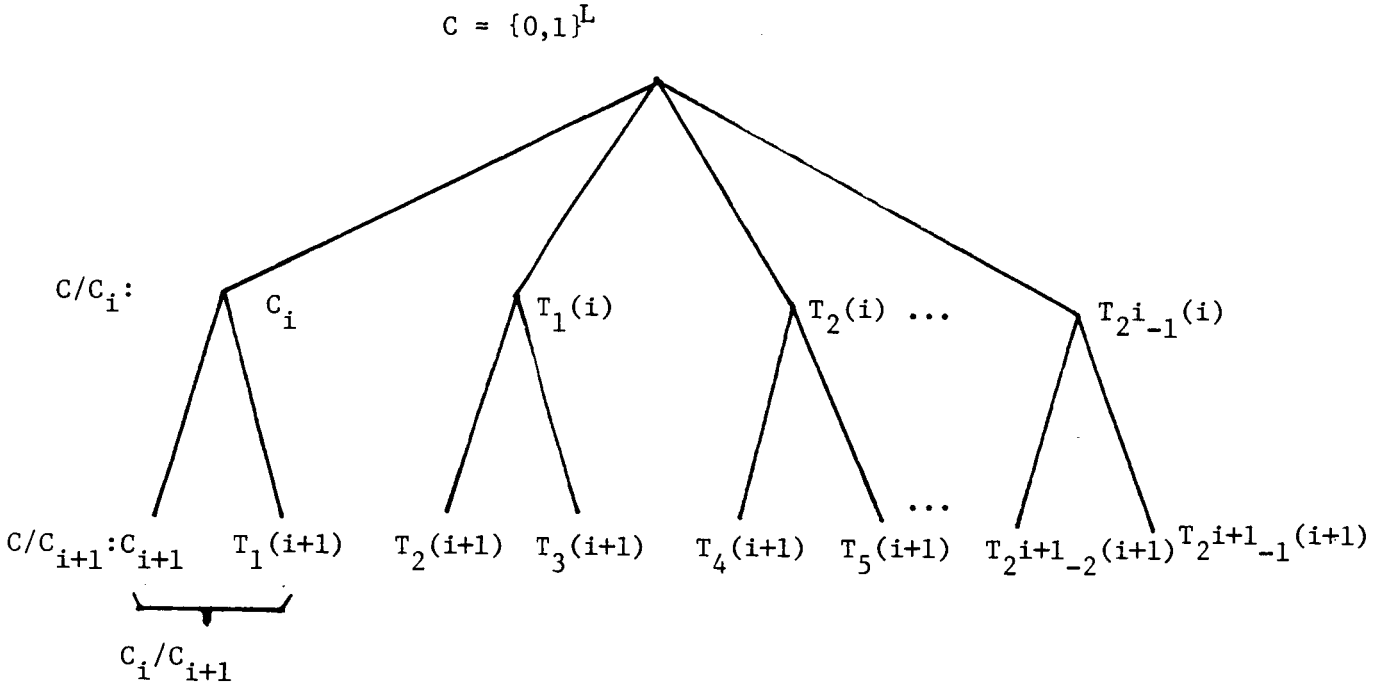


Fig. 3 The 2^i/2-way partition chain $C/C_i/C_{i+1}$ in the L-dimensional binary vector space.

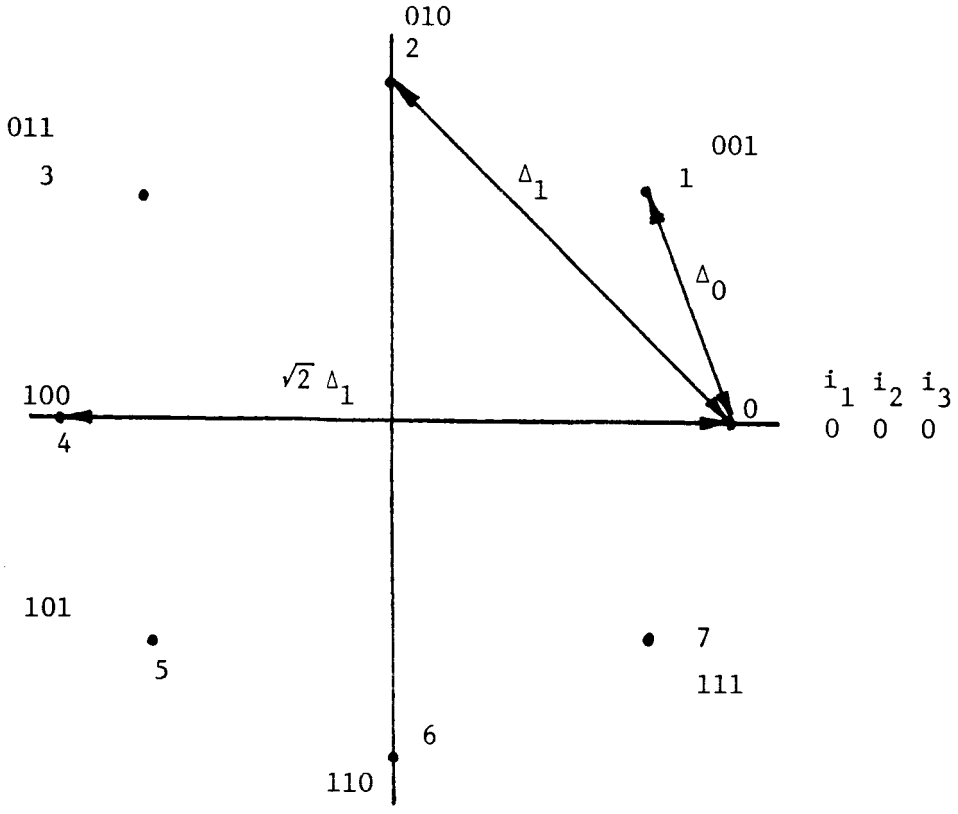


Fig. 4 8PSK signals and a mapping

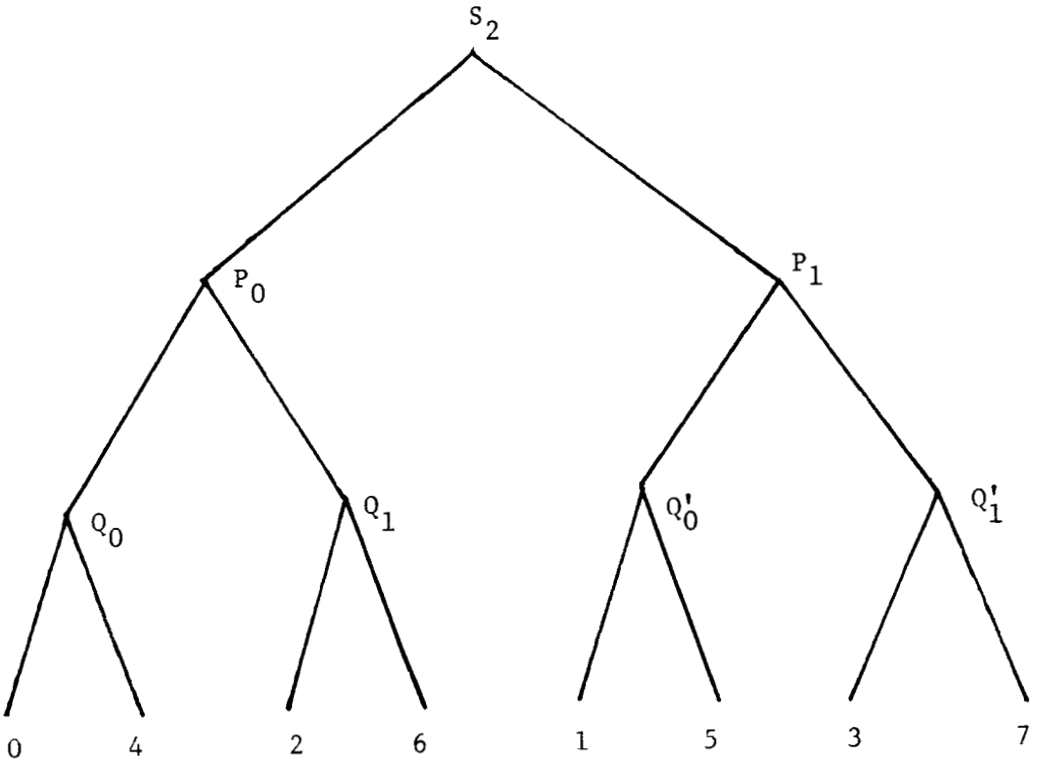


Fig. 5 The 2-dimensional 8PSK signal space partition tree.

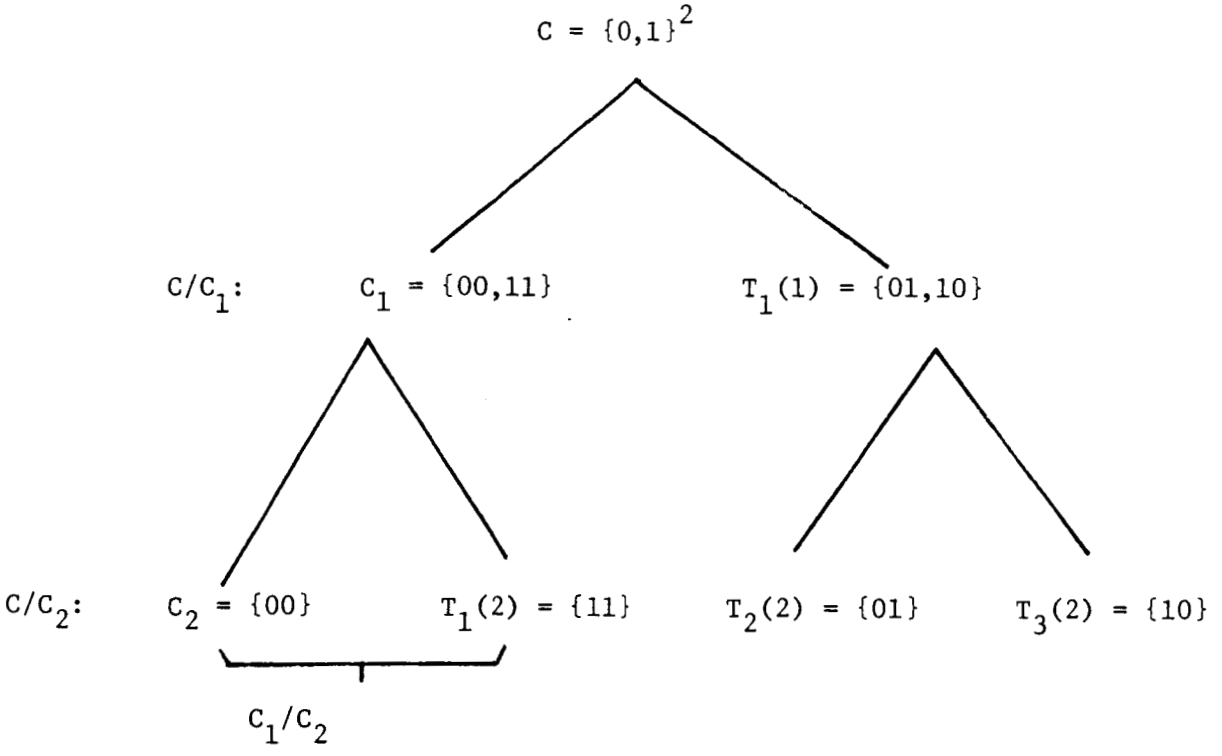


Fig. 6(a) The 2/2-way partition chain $C/C_1/C_2$ in the 2-dimensional binary vector space.

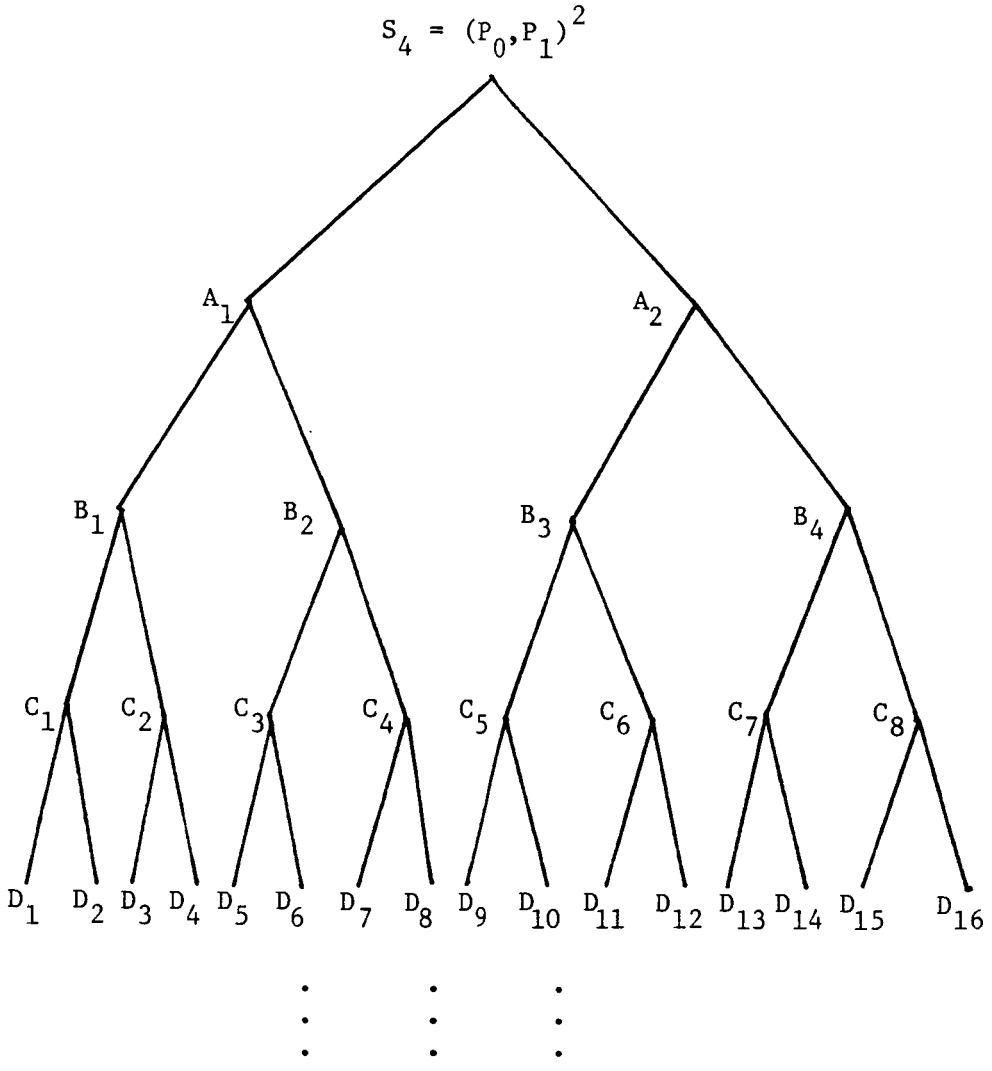


Fig. 6(b) The 4-dimensional 8PSK signal space partition tree.

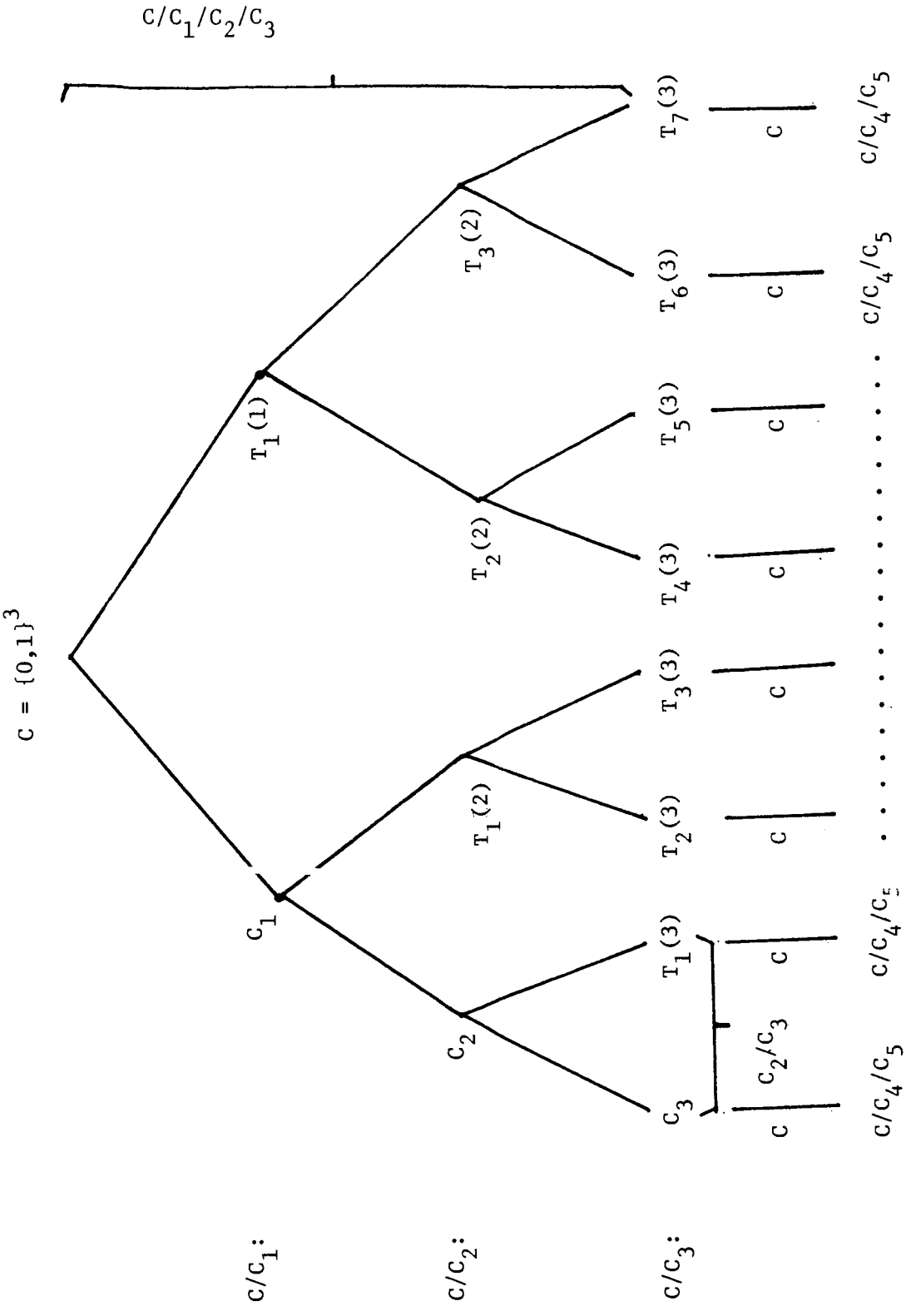


Fig. 7 The binary vector space partition rule for the 6-dimensional 8PSK signal space partition.

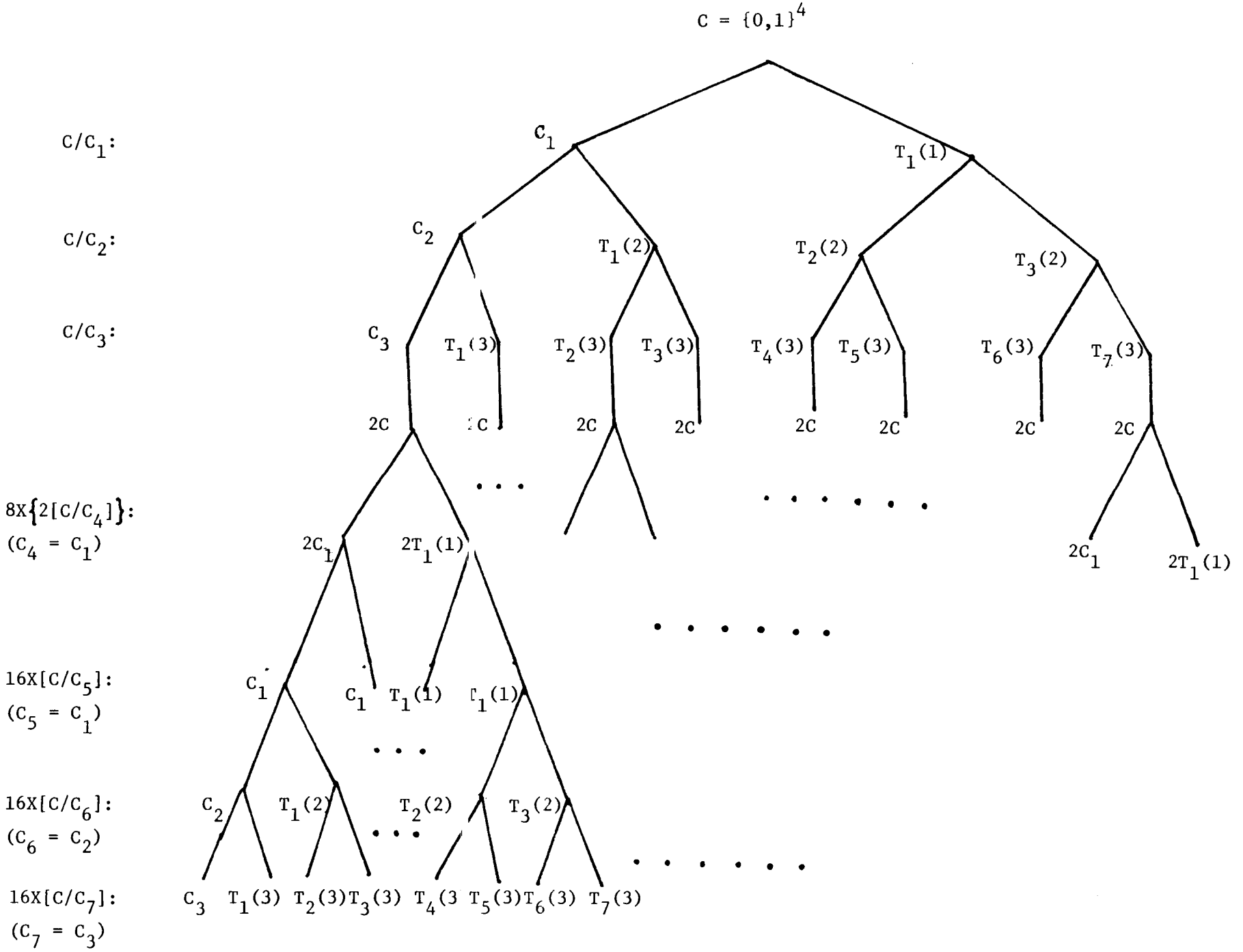
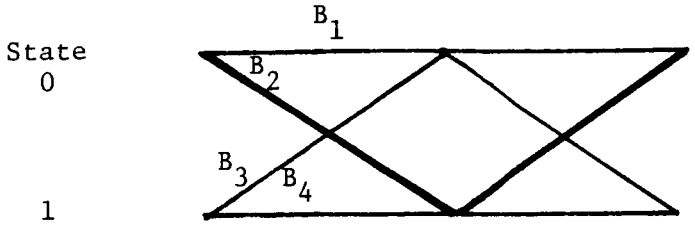
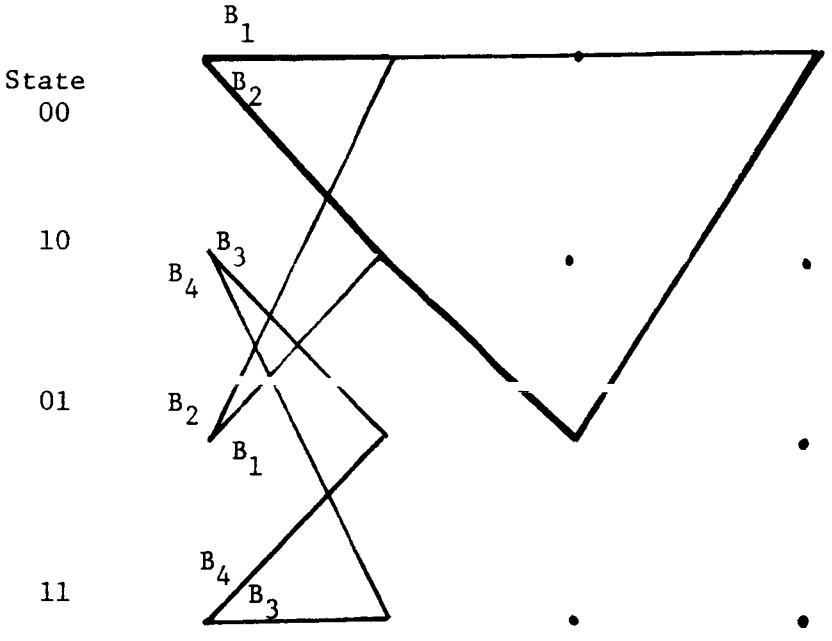


Fig. 8 The binary vector space partition rule for the 8-dimensional 8PSK signal space partition.



$$d_f^2 = 2\Delta_0^2 + \Delta_0^2 = 1.757$$

Fig. 9 Two-state R = 5/6 coded 2*8PSK



$$d_f^2 = 3\Delta_0^2 = 1.757$$

Fig. 10 Four-state R = 8/9 coded 3*8PSK

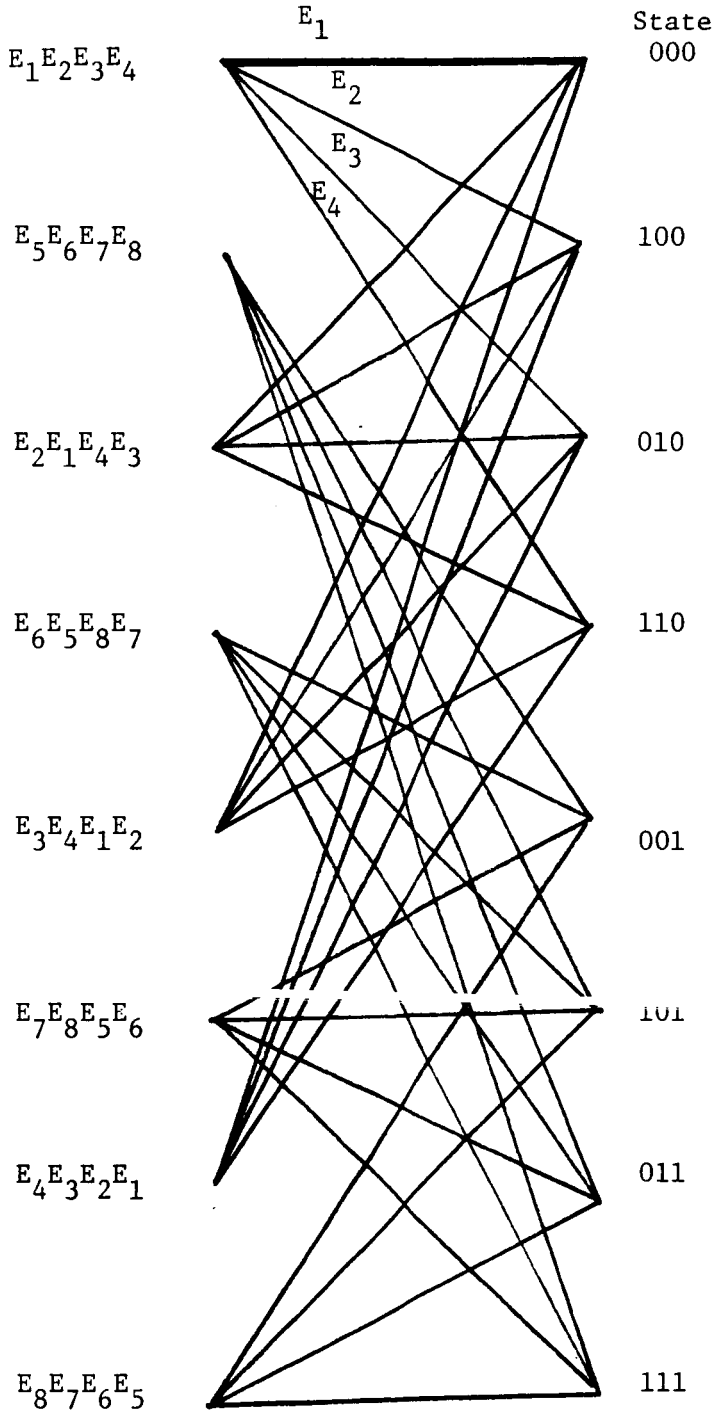


Fig. 11 Eight-state R = 9/12 coded 4*8PSK

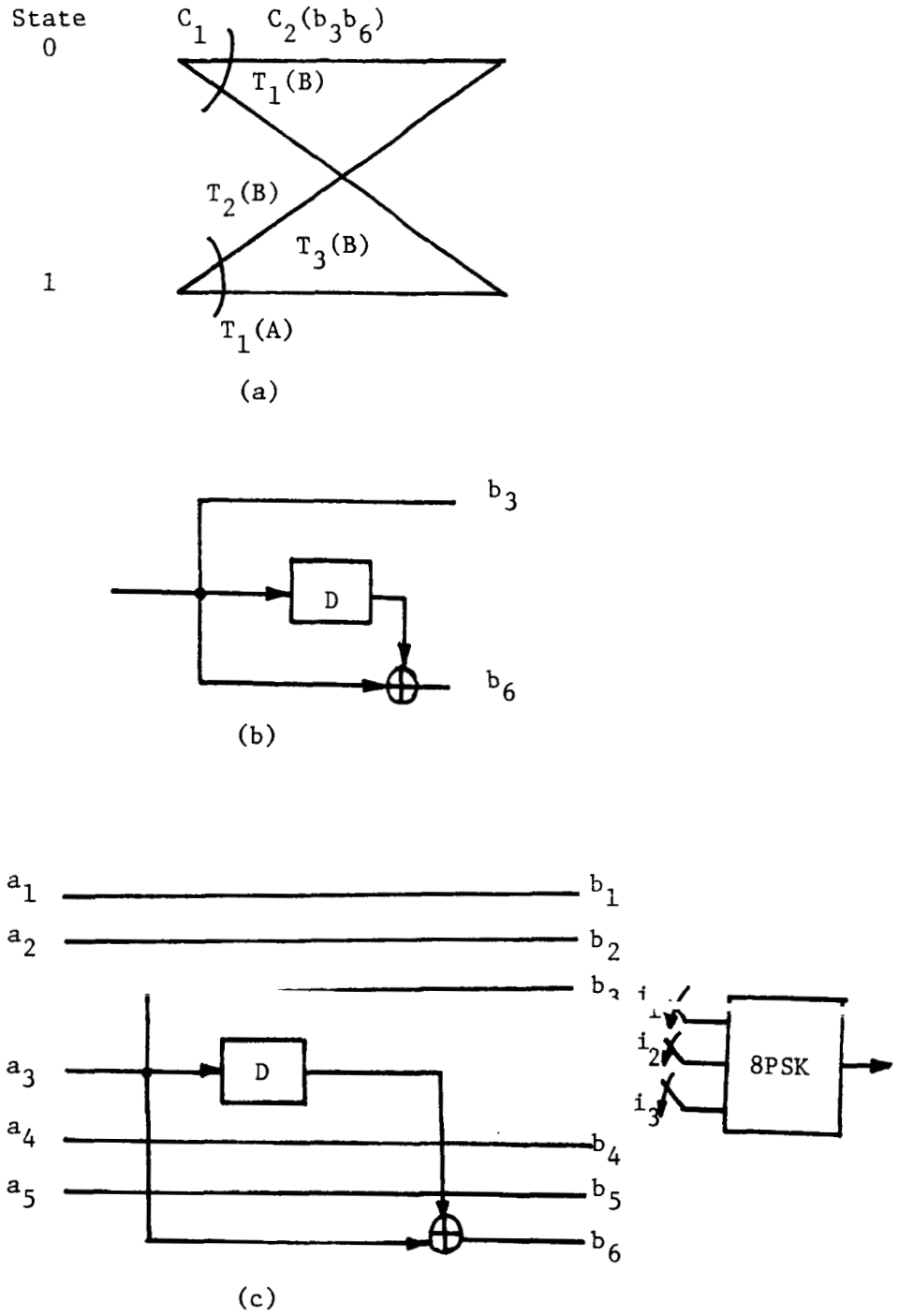


Fig. 12 Procedure for synthesizing a (6, 5, 1) encoder for the two-state R = 5/6 coded 2*8PSK of Example 1.

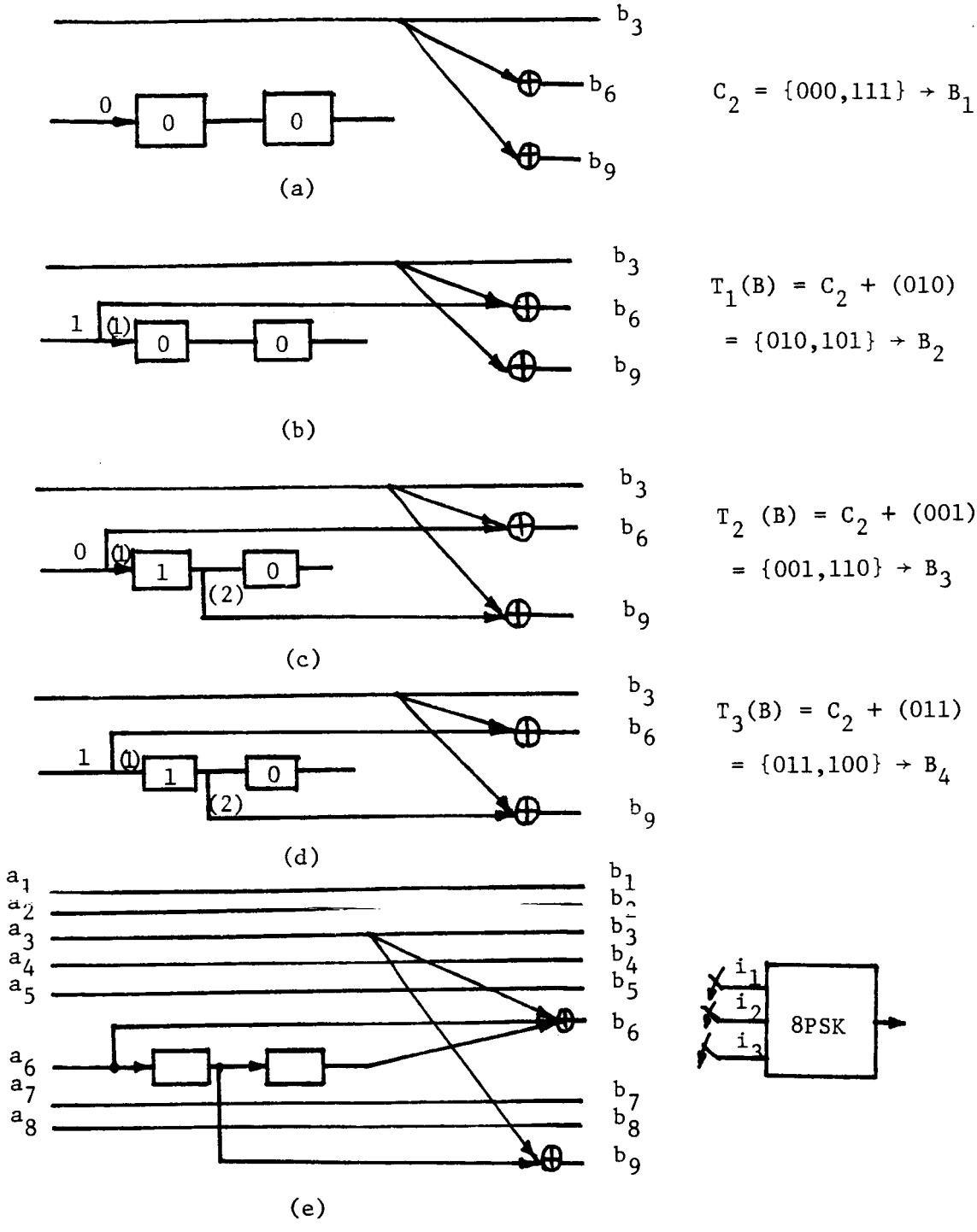


Fig. 13 Procedure for synthesizing a (9, 8, 2) encoder for the four-state R = 8/9 coded 3*8PSK of Example 2.

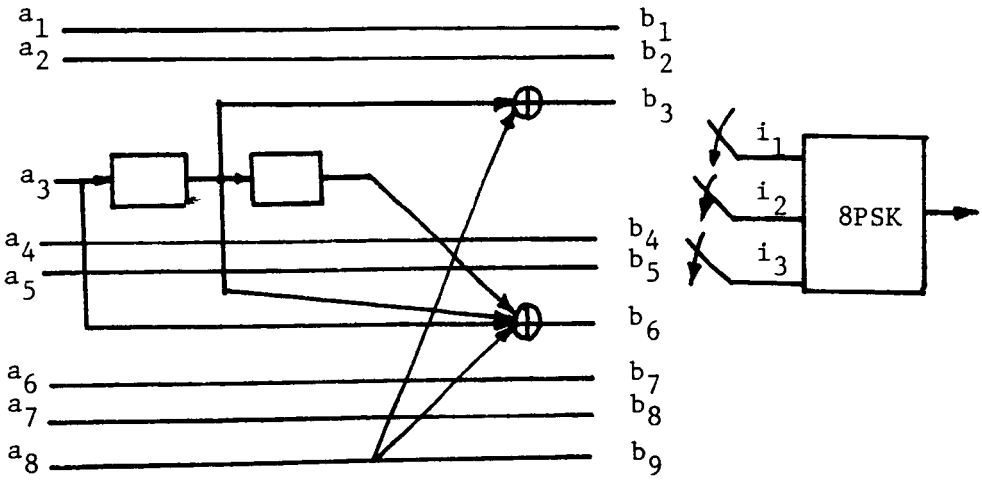


Fig. 14 Another (9, 8, 2) convolutional encoder for the four-state R = 8/9 coded 3*8PSK of Example 2.

Table 1. Coding gains obtained for R = k/n, with GCM(k, n) > 1, coded multi-dimensional QPSK modulation

Code rate R	Equivalent code rate	υ	UM	PUM	Improvement in HD	Coding gain γ (dB)
2/4		1		*	4/3	1.25
4/8		3		*	8/6	1.25
4/8	1/2	4	*		8/7	0.58
5/10		5	*		9/8	0.51
2/6		1		*	6/5	0.79
5/15		4		*	14/12	0.67
5/15	1/3	5	*		15/13	0.62
6/18		6	*		16/15	0.28
2/8		1		*	8/7	0.58
3/12		2		*	12/10	0.79
4/16		3		*	16/13	0.90
5/20	1/4	4		*	18/16	0.51
5/20		5	*		20/18	0.46
6/24		6	*		24/20	0.79
4/6		2		*	4/3	1.25
4/6	2/3	4	*		6/5	0.79

Table 2. The 4-dimensional 8PSK modulation signal space partition

Partition level	# of subsets	N(·)	Partition rule	$ED^2_{\min}(\cdot)$
S_4	1	64	--	$\Delta_0^2 = 0.586$
A	2	32	$\mathbf{C}/\mathbf{C}_1 : \mathbf{G}_1 = [1\ 1]$	$2\Delta_0^2 = 1.172$
B	4	16	$\mathbf{C}/\mathbf{C}_2 : \mathbf{C}_2 = \{0\ 0\}$	$\Delta_1^2 = 2$
C	8	8	$4\times[\mathbf{C}/\mathbf{C}_1] : \mathbf{G}_1 = [1\ 1]$	$2\Delta_1^2 = 4$
D	16	4	$4\times[\mathbf{C}/\mathbf{C}_2] : \mathbf{C}_2 = \{0\ 0\}$	$2\Delta_1^2 = 4$

Table 3(a). The 6-dimensional 8PSK modulation signal space partition I.

Partition level	# of subsets	$N(\cdot)$	Partition rule	$ED^2_{\min}(\cdot)$
S_6	1	512	--	$\Delta_0^2 = 0.586$
A	2	256	$\mathbf{C}/\mathbf{C}_1 : \mathbf{G}_1 = \begin{bmatrix}101\\011\end{bmatrix}$	$2\Delta_0^2 = 1.172$
B	4	128	$\mathbf{C}/\mathbf{C}_2 : \mathbf{G}_2 = [011]$	$2\Delta_0^2 = 1.172$
C	8	64	$\mathbf{C}/\mathbf{C}_3 : \mathbf{C}_3 = \{000\}$	$\Delta_1^2 = 2$
D	16	32	$8\times[\mathbf{C}/\mathbf{C}_4] : \mathbf{G}_4 = \begin{bmatrix}101\\011\end{bmatrix}$	$2\Delta_1^2 = 4$
E	32	16	$8\times[\mathbf{C}/\mathbf{C}_5] : \mathbf{G}_5 = [011]$	$2\Delta_1^2 = 4$

Table 3(b). The 6-dimensional 8PSK modulation signal space partition II.

Partition level	# of subsets	$N(\cdot)$	Partition rule	$ED^2_{\min}(\cdot)$
S_6	1	512	--	$\Delta_0^2 = 0.586$
A	2	256	$\mathbf{C/C_1} : \mathbf{G_1} = \begin{bmatrix} 101 \\ 010 \end{bmatrix}$	$\Delta_0^2 = 0.586$
B	4	128	$\mathbf{C/C_2} : \mathbf{G_2} = [111]$	$3\Delta_0^2 = 1.757$
C	8	64	$\mathbf{C/C_3} : \mathbf{C_3} = \{000\}$	$\Delta_1^2 = 2$
D	16	32	$8\times[\mathbf{C/C_4}] : \mathbf{G_4} = \begin{bmatrix} 101 \\ 011 \end{bmatrix}$	$2\Delta_1^2 = 4$
E	32	16	$8\times[\mathbf{C/C_5}] : \mathbf{G_5} = [011]$	$2\Delta_1^2 = 4$

Table 4. The 8-dimensional 8PSK modulation signal space partition				
Partition level	# of subsets	N(·)	Partition rule	$ED^2_{\min}(\cdot)$
S_8	1	4096	--	$\Delta_0^2 = 0.586$
A	2	2048	$\mathbf{C/C_1} : \mathbf{G_1} = \begin{bmatrix}1001\\0101\\0011\end{bmatrix}$	$2\Delta_0^2 = 1.176$
B	4	1024	$\mathbf{C/C_2} : \mathbf{G_2} = \begin{bmatrix}1010\\0101\end{bmatrix}$	$2\Delta_0^2 = 1.176$
C	8	512	$\mathbf{C/C_3} : \mathbf{G_3} = [1111]$	$\Delta_1^2 = 2$
D	16	256	$8\times\{2[\mathbf{C/C_4}]\} : \mathbf{G_4} = \mathbf{G_1}$	$4\Delta_0^2 = 2.343$
E	32	128	$16\times[\mathbf{C/C_5}] : \mathbf{G_5} = \mathbf{G_1}$	$2\Delta_1^2 = 4$
F	64	64	$16\times[\mathbf{C/C_6}] : \mathbf{G_6} = \mathbf{G_2}$	$2\Delta_1^2 = 4$
G	128	32	$16\times[\mathbf{C/C_7}] : \mathbf{G_7} = \mathbf{G_3}$	$2\Delta_1^2 = 4$

Table 5. Signal subsets and encoder output mapping for Example 1

	b_1	b_2	b_4	b_5	b_3	b_6	
B_1	*	*	*	*	(0	0) = C_2	} C_1
B_2	*	*	*	*	(1	1) = $T_1(B)$	
B_3	*	*	*	*	(0	1) = $T_2(B)$	} $T_1(A)$
B_4	*	*	*	*	(1	1) = $T_3(B)$	

Table 6. Signal subsets and encoder output mapping for Example 2

	b_1	b_2	b_4	b_5	b_7	b_8	b_3	b_6	b_9		
B_1= $\begin{Bmatrix} P_0 \times P_0 \times P_0 \\ P_1 \times P_1 \times P_1 \end{Bmatrix}$	*	*	*	*	*	*	$\begin{Bmatrix} 0 \\ 1 \end{Bmatrix}$	$\begin{Bmatrix} 0 \\ 1 \end{Bmatrix}$	$\begin{Bmatrix} 0 \\ 1 \end{Bmatrix}$	C_2	C_1
B_2= $\begin{Bmatrix} P_0 \times P_1 \times P_0 \\ P_1 \times P_0 \times P_1 \end{Bmatrix}$	*	*	*	*	*	*	$\begin{Bmatrix} 0 \\ 1 \end{Bmatrix}$	$\begin{Bmatrix} 1 \\ 0 \end{Bmatrix}$	$\begin{Bmatrix} 0 \\ 1 \end{Bmatrix}$	$T_1(B)$	
B_3= $\begin{Bmatrix} P_0 \times P_0 \times P_1 \\ P_1 \times P_1 \times P_0 \end{Bmatrix}$	*	*	*	*	*	*	$\begin{Bmatrix} 0 \\ 1 \end{Bmatrix}$	$\begin{Bmatrix} 0 \\ 1 \end{Bmatrix}$	$\begin{Bmatrix} 1 \\ 0 \end{Bmatrix}$	$T_2(B)$	$T_1(A)$
B_4= $\begin{Bmatrix} P_0 \times P_1 \times P_1 \\ P_1 \times P_0 \times P_0 \end{Bmatrix}$	*	*	*	*	*	*	$\begin{Bmatrix} 0 \\ 1 \end{Bmatrix}$	$\begin{Bmatrix} 1 \\ 0 \end{Bmatrix}$	$\begin{Bmatrix} 1 \\ 0 \end{Bmatrix}$	$T_3(B)$	

Table 7(a). R = 5/6 trellis coded 2*8PSK[1]
(R_{eff} = 1.25 bits/dimension)

υ	d_f^2	$M(d_f^2)$	γ_{8PSK} (dB)	γ_{QPSK} (dB)	Generators Matrices[2] G_o	G_1	G_2
1	1.757	8	3.98	0.41	$\begin{bmatrix} 4 & 0 \\ 2 & 0 \\ 1 & 1 \\ 0 & 4 \\ 0 & 2 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$	
2	2.0	4	4.54	0.97	$\begin{bmatrix} 4 & 0 \\ 2 & 0 \\ 1 & 1 \\ 0 & 4 \\ 0 & 2 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$
3	2.929	16	6.20	2.63	$\begin{bmatrix} 4 & 0 \\ 2 & 0 \\ 1 & 1 \\ 0 & 4 \\ 2 & 2 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 2 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$

1: Equivalent codes have been found independently by Costello and Lafanechere [5] and by Wilson [6].
2: Generator matrices are given in octal notation.

Table 7(b). R = 4/6 trellis coded 2*8PSK
(R_{eff} = 1 bit/dimension)

υ	d_f^2	$M(d_f^2)$	γ_{8PSK} (dB)	γ_{QPSK} (dB)	Generators Matrices G_o	G_1	G_2
1	3.172	8	5.57	2.00	$\begin{bmatrix} 4 & 0 \\ 2 & 0 \\ 0 & 4 \\ 2 & 2 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$	
2	4.0	6	6.58	3.01	$\begin{bmatrix} 4 & 0 \\ 2 & 0 \\ 0 & 4 \\ 2 & 2 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 \\ 2 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$
3	4.0	2	6.58	3.01	$\begin{bmatrix} 4 & 0 \\ 2 & 2 \\ 0 & 2 \\ 0 & 4 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 2 & 2 \\ 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 \\ 0 & 2 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$

Table 8(a). R = 8/9 trellis coded 3*8PSK
(R_{eff} = 1.33 bits/dimension)

υ	d_f^2	$M(d_f^2)$	γ_{8PSK} (dB)	γ_{QPSK} (dB)	Generators Matrices G_o	G_1	G_2
1	1.172	4	2.49	-1.08	$\begin{bmatrix}4&0&0\\2&0&0\\0&1&0\\0&4&0\\0&2&0\\0&0&4\\0&0&2\\1&1&1\end{bmatrix}$	$\begin{bmatrix}0&0&0\\0&0&0\\1&1&0\\0&0&0\\0&0&0\\0&0&0\\0&0&0\\0&0&0\end{bmatrix}$	(II)
2	1.757	16	4.26	0.69	$\begin{bmatrix}4&0&0\\2&0&0\\1&1&1\\0&4&0\\0&2&0\\0&1&0\\0&0&4\\0&0&2\end{bmatrix}$	$\begin{bmatrix}0&0&0\\0&0&0\\0&0&0\\0&0&0\\0&0&0\\0&0&1\\0&0&0\\0&0&0\end{bmatrix}$	$\begin{bmatrix}0&0&0\\0&0&0\\0&0&0\\0&0&0\\0&0&0\\0&1&0\\0&0&0\\0&0&0\end{bmatrix}$ (II)
3	2.0	6	4.82	1.25	$\begin{bmatrix}4&0&0\\2&0&0\\0&1&1\\1&0&1\\0&4&0\\0&2&0\\0&0&4\\0&0&2\end{bmatrix}$	$\begin{bmatrix}0&0&0\\0&0&0\\0&0&1\\0&1&1\\0&0&0\\0&0&0\\0&0&0\\0&0&0\end{bmatrix}$	$\begin{bmatrix}0&0&0\\0&0&0\\1&0&1\\0&0&0\\0&0&0\\0&0&0\\0&0&0\\0&0&0\end{bmatrix}$ (I)

II: Partition II
I : Partition I

Table 8(b). R = 7/9 trellis coded 3*8PSK
(R_{eff} = 1.167 bits/dimension)

υ	d_f^2	$M(d_f^2)$	γ_{8PSK} (dB)	γ_{QPSK} (dB)	Generators Matrices G_o	G_1	G_2
1	2.0	6	4.24	0.67	$\begin{bmatrix}4&0&0\\2&0&0\\0&1&1\\0&4&0\\0&2&0\\0&0&4\\0&0&2\end{bmatrix}$	$\begin{bmatrix}0&0&0\\0&0&0\\1&0&1\\0&0&0\\0&0&0\\0&0&0\\0&0&0\end{bmatrix}$	(I)
2	2.342	8	4.93	1.36	$\begin{bmatrix}4&0&0\\2&0&2\\0&0&2\\1&1&1\\0&4&0\\0&2&2\\0&0&4\end{bmatrix}$	$\begin{bmatrix}0&0&0\\0&0&0\\0&1&0\\0&0&2\\0&0&0\\0&0&0\\0&0&0\end{bmatrix}$	(II)
3	3.757	24	6.98	3.41	$\begin{bmatrix}4&0&0\\2&0&2\\0&0&2\\1&1&1\\0&4&0\\0&2&2\\0&0&4\end{bmatrix}$	$\begin{bmatrix}0&0&0\\0&0&0\\0&1&0\\0&0&2\\0&0&0\\0&0&0\\0&0&0\end{bmatrix}$	$\begin{bmatrix}0&0&0\\0&0&0\\1&1&1\\0&0&0\\0&0&0\\0&0&0\\0&0&0\end{bmatrix}$ (II)

Table 8(c). R = 6/9 trellis coded 3*8PSK
(R_{eff} = 1 bit/dimension)

υ	d_f^2	$M(d_f^2)$	γ_{8PSK} (dB)	γ_{QPSK} (dB)	Generators Matrices G_o	G_1	G_2
1	3.757	24	6.31	2.74	$\begin{bmatrix} 4&0&0 \\ 2&0&2 \\ 0&0&2 \\ 0&4&0 \\ 0&2&2 \\ 0&0&4 \end{bmatrix}$	$\begin{bmatrix} 0&0&0 \\ 0&0&0 \\ 1&1&1 \\ 0&0&0 \\ 0&0&0 \\ 0&0&0 \end{bmatrix}$	(II)
2	4.0	15	6.58	3.01	$\begin{bmatrix} 4&0&0 \\ 2&0&2 \\ 0&0&2 \\ 0&4&0 \\ 0&2&2 \\ 0&0&4 \end{bmatrix}$	$\begin{bmatrix} 0&0&0 \\ 0&0&0 \\ 1&1&1 \\ 0&0&0 \\ 0&0&0 \\ 0&0&0 \end{bmatrix}$	$\begin{bmatrix} 0&0&0 \\ 0&0&0 \\ 0&0&2 \\ 0&0&0 \\ 0&0&0 \\ 0&0&0 \end{bmatrix}$ (II)
3	4.0	7	6.58	3.01	$\begin{bmatrix} 4&0&0 \\ 2&0&2 \\ 0&0&2 \\ 0&4&0 \\ 0&2&2 \\ 0&0&4 \end{bmatrix}$	$\begin{bmatrix} 0&0&0 \\ 1&1&1 \\ 2&0&2 \\ 0&0&0 \\ 0&0&0 \\ 0&0&0 \end{bmatrix}$	$\begin{bmatrix} 0&0&0 \\ 0&0&2 \\ 0&0&0 \\ 0&0&0 \\ 0&0&0 \\ 0&0&0 \end{bmatrix}$ (II)

Table 9(a) R = 11/12 trellis coded 4*8PSK
(R_{eff} = 1.375 bits/dimension)

υ	d_f^2	$M(d_f^2)$	γ_{8PSK} (dB)	γ_{QPSK} (dB)	Generators Matrices G_o	G_1	G_2
1	1.172	8	2.63	-0.94	$\begin{bmatrix}4&0&0&0\\2&0&0&0\\1&0&1&0\\0&4&0&0\\0&2&0&0\\0&1&1&0\\0&0&4&0\\0&0&2&0\\0&0&0&4\\0&0&0&2\\0&1&0&1\end{bmatrix}$	$\begin{bmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\\0&1&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{bmatrix}$	
2	1.757	48	4.39	0.82	$\begin{bmatrix}4&0&0&0\\2&0&0&0\\0&4&0&0\\0&2&0&0\\1&0&1&0\\1&1&0&0\\0&0&4&0\\0&0&2&0\\0&0&0&4\\0&0&0&2\\1&1&1&1\end{bmatrix}$	$\begin{bmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\\1&1&1&0\\1&0&1&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{bmatrix}$	
3	2.0	8	4.95	1.38	$\begin{bmatrix}4&0&0&0\\2&0&0&0\\0&4&0&0\\0&2&0&0\\1&0&1&0\\1&1&0&0\\0&0&4&0\\0&0&2&0\\0&0&0&4\\0&0&0&2\\1&1&1&1\end{bmatrix}$	$\begin{bmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\\1&1&1&0\\1&0&1&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{bmatrix}$	$\begin{bmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\\1&1&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{bmatrix}$

Table 9(b). R = 10/12 trellis coded 4*8PSK
(R_{eff} = 1.25 bits/dimension)

υ	d_f^2	$M(d_f^2)$	γ_{8PSK} (dB)	γ_{QPSK} (dB)	Generators Matrices G_o	G_1	G_2
1	2.0	8	4.54	0.97	$\begin{bmatrix}4&0&0&0\\2&0&0&0\\0&4&0&0\\0&2&0&0\\1&0&1&0\\0&0&4&0\\0&0&2&0\\0&0&0&4\\0&0&0&2\\1&1&1&1\end{bmatrix}$	$\begin{bmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\\1&1&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{bmatrix}$	
2	2.343	72	5.23	1.66	$\begin{bmatrix}4&0&0&0\\2&0&0&2\\0&0&0&2\\1&0&1&0\\0&4&0&0\\0&2&0&2\\0&0&4&0\\0&0&2&2\\0&0&0&4\\1&1&1&1\end{bmatrix}$	$\begin{bmatrix}0&0&0&0\\0&0&0&0\\1&1&0&0\\0&0&0&2\\0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{bmatrix}$	
3	2.343	8	5.23	1.66	$\begin{bmatrix}4&0&0&0\\2&0&0&2\\0&0&0&2\\1&0&1&0\\0&4&0&0\\0&2&0&2\\0&0&4&0\\0&0&2&2\\0&0&0&4\\1&1&1&1\end{bmatrix}$	$\begin{bmatrix}0&0&0&0\\0&0&0&0\\1&1&0&0\\0&0&0&2\\0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{bmatrix}$	$\begin{bmatrix}0&0&0&0\\0&0&0&0\\1&0&1&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{bmatrix}$

Table 9(c). R = 9/12 trellis coded 4*8PSK
(R_{eff} = 1.125 bits/dimension)

υ	d_f^2	$M(d_f^2)$	γ_{8PSK} (dB)	γ_{QPSK} (dB)	Generators Matrices G_o	G_1	G_2
1	2.343	8	4.77	1.20	$\begin{bmatrix}4&0&0&0\\2&0&0&0\\0&0&0&2\\0&4&0&0\\0&2&0&2\\0&0&4&0\\0&0&2&2\\0&0&0&4\\1&1&1&1\end{bmatrix}$	$\begin{bmatrix}0&0&0&0\\0&0&0&0\\1&0&1&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{bmatrix}$	
2	3.172	16	6.08	2.51	$\begin{bmatrix}4&0&0&0\\2&0&0&2\\1&1&1&1\\0&0&0&2\\0&4&0&0\\0&2&0&2\\0&0&4&0\\0&0&2&2\\0&0&0&4\end{bmatrix}$	$\begin{bmatrix}0&0&0&0\\0&0&0&0\\0&1&0&1\\1&1&1&1\\0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{bmatrix}$	
3	4.0	28	7.09	3.52	$\begin{bmatrix}4&0&0&0\\2&0&0&2\\1&1&1&1\\0&0&0&2\\0&4&0&0\\0&2&0&2\\0&0&4&0\\0&0&2&2\\0&0&0&4\end{bmatrix}$	$\begin{bmatrix}0&0&0&0\\0&0&0&0\\0&1&0&1\\1&1&1&1\\0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{bmatrix}$	$\begin{bmatrix}0&0&0&0\\0&0&0&0\\0&0&0&2\\0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{bmatrix}$

Table 9(d). R = 8/12 trellis coded 4*8PSK
(R_{eff} = 1 bit/dimension)

υ	d_f^2	$M(d_f^2)$	γ_{8PSK} (dB)	γ_{QPSK} (dB)	Generators Matrices G_o	G_1
1	4.0	28	6.58	3.01	$\begin{bmatrix}4&0&0&0\\2&0&0&2\\1&1&1&1\\0&4&0&0\\0&2&0&2\\0&0&4&0\\0&0&2&2\\0&0&0&4\end{bmatrix}$	$\begin{bmatrix}0&0&0&0\\0&0&0&0\\0&0&0&2\\0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{bmatrix}$
2	4.0	12	6.58	3.01	$\begin{bmatrix}4&0&0&0\\2&0&2&0\\0&0&2&2\\1&1&1&1\\0&4&0&0\\0&2&0&2\\0&0&4&0\\0&0&0&4\end{bmatrix}$	$\begin{bmatrix}0&0&0&0\\0&0&0&0\\0&0&0&2\\0&0&2&2\\0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{bmatrix}$
3	4.0	4	6.58	3.01	$\begin{bmatrix}4&0&0&0\\2&2&2&2\\0&2&0&2\\0&4&0&0\\0&0&2&2\\1&1&1&1\\0&0&4&0\\0&0&0&4\end{bmatrix}$	$\begin{bmatrix}0&0&0&0\\0&0&0&0\\0&0&0&2\\0&0&0&0\\0&2&0&2\\0&0&2&2\\0&0&0&0\\0&0&0&0\end{bmatrix}$