Multidimensional Trellis Coded Phase Modulation Using a Multilevel Concatenation Approach Part I: Code Design

Sandeep Rajpal, *Member, IEEE*, Do Jun Rhee, *Member, IEEE*, and Shu Lin, *Fellow, IEEE*

***Abstract*—The first part of this paper presents a simple and systematic technique for constructing multidimensional M-ary phase shift keying (MPSK) trellis coded modulation (TCM) codes. The construction is based on a multilevel concatenation approach, in which binary convolutional codes with good free branch distances are used as the outer codes and block MPSK modulation codes are used as the inner codes (or the signal spaces). Conditions on phase invariance of these codes are derived and a multistage decoding scheme for these codes is proposed. The proposed technique can be used to construct good codes for both the additive white Gaussian noise (AWGN) and fading channels as is shown in the second part of this paper.**

***Index Terms*—Concatenation, MPSK modulation, multidimensional trellis coded modulation, multistage decoding.**

I. Introduction

SINCE the publication of the celebrated paper by Ungerboeck on trellis coded modulation (TCM) [1], there has been a boom of research in this area. Over the last fourteen years, researchers have proposed various techniques of constructing modulation codes using both convolutional codes (TCM) [1]–[7] and block codes [block coded modulation (BCM)] [8]–[14]. Almost all existing techniques for constructing TCM codes rely heavily on computer searches to find good TCM codes. These techniques work very well for small code complexities and rates. However, for large code complexities and high rates, the search becomes extremely time consuming (if not impossible) and a more systematic technique of construction is required. Most of the problems associated with the algebraic construction of TCM codes arise due to the lack of in-depth knowledge of convolutional codes. In addition, the nonlinearity of the mapping function (true for most signal constellations) which maps the coded output bits of the convolutional encoder onto the signal set, complicates the problem further. BCM codes on the other hand, have the advantage of being extremely rich in algebraic structure and phase symmetry, as has been shown in [10]–[13]. BCM codes however, have the disadvantage of being slightly poor in performance for low signal-to-noise ratio (SNR), as compared to TCM codes of the same decoding complexity, due to the large number of nearest neighbors.

Pietrobon *et al.* extended Ungerboeck's results to multidimensional MPSK signal constellations [3]. They proposed a set partitioning technique for multidimensional MPSK signal constellations similar to Ungerboeck's set partitioning technique and then used computer search to design multidimensional MPSK TCM codes. However, due to the limitations of computer search, as were outlined above, they restricted themselves to 4 × 2 dimensions. In addition, to reduce the search complexity, they placed some other restrictions on the computer search. Multidimensional MPSK TCM codes have various advantages over two-dimensional (2-D) Ungerboeck TCM codes, the main ones being: 1) higher spectral efficiencies can be achieved, 2) codes constructed over multidimensional MPSK signal constellations have better phase invariance properties than that of 2-D Ungerboeck MPSK codes, and 3) lower average decoding complexities to achieve the same performance.

A common point to be noted among all the construction techniques available in literature (whether TCM or BCM) is that the modulation codes constructed by these techniques require large decoding complexity to achieve large coding gains. The large decoding complexity of these codes makes them impractical for applications where high reliability and high data rates are required. As such, what is required is a *multistage* decoding technique which reduces the decoding complexity, while maintaining good performance.

This paper presents a simple and systematic technique for designing multidimensional MPSK TCM codes with minimal computer search. The technique will be used to construct good codes for both the AWGN and fading channels. Though the main emphasis has been to construct codes for the MPSK signal constellation, the results are applicable to other signal constellations as well and modifying the existing construction for other signal constellations is straight forward. This paper is organized as follows: Section II of the paper presents a new concept, branch distance of convolutional codes, which will be used extensively in the later sections. Section III outlines

Paper approved by D. Divsalar, the Editor for Coding Theory and Applications of the IEEE Communications Society. Manuscript received March 3, 1995; revised May 3, 1996. This work was supported by the National Science Foundation under Grants NCR-9115400 and NCR-9415374, and by NASA under Grant NAG 5-931. This paper was presented in part at the 1993 International Symposium on Communications, Taiwan, R.O.C., December 7–10, 1993.

S. Rajpal is with the Rockwell International Corporation, Multimedia Communications Division, Newport Beach, CA 92660 USA.

D. J. Rhee is with the LSI LOGIC Corporation, Milpitas, CA 95035 USA.

S. Lin is with the Department of Electrical Engineering, University of Hawaii at Manoa, Honolulu, HI 96822 USA.

Publisher Item Identifier S 0090-6778(97)00717-4.

the basic construction technique of the proposed codes, and, in addition, shows that the codes constructed in [3] turn out to be a special case of the proposed construction. Section IV discusses phase invariance. In Section V, a *multistage* decoding algorithm for the proposed codes is presented and it's decoding complexity is discussed. Section VI concludes by discussing the design rules for constructing good codes using the proposed technique.

II. Branch Distance of Convolutional Codes

For two code sequences $\boldsymbol{u}$ and $\boldsymbol{v}$ in a binary linear convolutional code, the *branch distance* between them, denoted $d_b(\boldsymbol{u},\boldsymbol{v})$, is defined as the number of branches in which $\boldsymbol{u}$ and $\boldsymbol{v}$ differ (or equivalently, this is simply equal to the number of nonzero branches in $\boldsymbol{u}\oplus\boldsymbol{v}$, where $\oplus$ denotes binary addition). For a code sequence $\boldsymbol{u}$ in a binary linear convolutional code, the *branch weight* of $\boldsymbol{u}$ denoted $w_b(\boldsymbol{u})$ is simply the number of nonzero branches in $\boldsymbol{u}$ (or equivalently $w_b(\boldsymbol{u})$ is the *branch distance* between $\boldsymbol{u}$ and $\boldsymbol{0}$, where $\boldsymbol{0}$ refers to the all-zero code sequence, i.e., $w_b(\boldsymbol{u}) = d_b(\boldsymbol{u},\boldsymbol{0})$). The minimum free branch distance of a convolutional code C, denoted $d_{\text{B-free}}$, is the minimum branch distance between any two code sequences, i.e.,

$$d_{\text{B-free}} \triangleq \min\{d_b(\boldsymbol{u},\boldsymbol{v})\colon \boldsymbol{u},\boldsymbol{v}\in C \text{ and } \boldsymbol{u}\neq\boldsymbol{v}\}. \tag{2.1}$$

Theorem 1: For a rate k/n feedforward binary linear convolutional code of total encoder memory γ, its minimum free branch distance, $d_{\text{B-free}}$, is upper bounded by $1+\lfloor\gamma/k\rfloor$.

Proof: Let the k inputs to the encoder be denoted as $I_1, I_2, \cdots, I_k$ and let the encoder memories associated with input I_i be γ_i for $1\le i\le k$. Let $\min\{w_b(\boldsymbol{u})\}$ denote the minimum branch weight among all the code sequences associated with the binary linear convolutional code. Let $\min_{i=1}^{k}\gamma_i = \gamma_j$. Consider that the binary sequence $(1,0,0,\cdots)$ is fed into the input I_j and the all zero sequence $(0,0,0,\cdots)$ is fed into the remaining inputs. The branch weight of the resulting code sequence is upper bounded by $1+\gamma_j$. Hence, $\min\{w_b(\boldsymbol{u})\}\le 1+\gamma_j$. Since the code is linear, this also corresponds to an upper bound on the minimum free branch distance, i.e., $d_{\text{B-free}}\le\{1+\min_{i=1}^{k}\gamma_i\}$. Given any γ and k, the idea is to maximize $d_{\text{B-free}}$. Hence, $\max_{\gamma,k}(d_{\text{B-free}})\le \max_{\gamma,k}\{1+\min_{i=1}^{k}\gamma_i\}$, i.e., the best $d_{\text{B-free}}$ for a given γ and k is $\le\{1+\max_{\gamma,k}\{\min_{i=1}^{k}\gamma_i\}\}$. It is readily seen that the value of $\max_{\gamma,k}\{\min_{i=1}^{k}\gamma_i\}$ is $\lfloor\gamma/k\rfloor$. △△

Theorem 2: If $d_{\text{B-free}} = 1+\lfloor\gamma/k\rfloor$, then $N_{\text{B-free}}$, the number of codewords with branch weight $d_{\text{B-free}}$, is lower bounded by (2^p-1) where p is the number of inputs of the convolutional encoder which have an encoder memory of $\lfloor\gamma/k\rfloor$ associated with it.

Proof: Let $\boldsymbol{e}_1$ denote the binary sequence $(1,0,0,\cdots)$ i.e., 1 followed by the all zero sequence and let $\boldsymbol{e}_0$ denote the all zero binary sequence $(0,0,0,\cdots)$. Consider any nonzero code sequence $\boldsymbol{u}$. Then $w_b(\boldsymbol{u})\ge 1+\lfloor\gamma/k\rfloor$. Let the p inputs which have an encoder memory of $\lfloor\gamma/k\rfloor$ associated with it be I_j for $1\le j\le p$. Consider that $\boldsymbol{e}_0$ is fed into the inputs I_j for $p+1\le j\le k$. Also, consider that the inputs I_j for $1\le j\le p$ can take only one of the two sequences $\boldsymbol{e}_0$ or $\boldsymbol{e}_1$. Then the convolutional encoder under this constraint has (2^p-1) distinct nonzero input sequences. Each of the (2^p-1) sequences will have branch weight $\le 1+\lfloor\gamma/k\rfloor$. Since $d_{\text{B-free}} = 1+\lfloor\gamma/k\rfloor$, each of the (2^p-1) sequences thus has branch weight $1+\lfloor\gamma/k\rfloor$. Hence, $N_{\text{B-free}}\ge(2^p-1)$. △△

TABLE I
Optimum Branch Distance Rate 1/2 Codes

$\gamma^{\dagger}$	G	$d_{\text{B-free}}$ ‡	$N_{\text{B-free}}$ △	$d_{\text{H-free}}$ ⊔	$N_{\text{H-free}}$ *
1	$\binom{4}{2}_8$	2	1	2	1
2	$\binom{5}{2}_8$	3	1	3	1
3	$\binom{5}{64}_8$	4	1	5	1
4	$\binom{44}{32}_8$	5	2	5	1
5	$\binom{62}{35}_8$	6	2	7	3
6	$\binom{51}{664}_8$	7	4	8	2
7	$\binom{344}{532}_8$	8	6	9	2
8	$\binom{622}{575}_8$	8	1	10	4
9	$\binom{355}{6244}_8$	9	1	11	2
10	$\binom{3576}{6322}_8$	10	3	12	2

† Total encoder memory.
‡ Minimum free branch distance.
△ Number of codewords with branch distance $d_{\text{B-free}}$.
⊔ Free Hamming distance.
* Number of codewords with Hamming distance $d_{\text{H-free}}$.
Note: The code generators have been listed in octal, where the octal representation of xyz is $4\cdot x + 2\cdot y + z$ and x, y and z denote three binary bits.

A binary linear feed-forward convolutional code is said to be *optimal* in terms of branch distance if it achieves the upper bound as stated in Theorem 1 for a given γ and k. Also, a code is said to be optimal in terms of the free Hamming distance, $d_{\text{H-free}}$, if it achieves the maximum $d_{\text{H-free}}$ possible for a given γ, k and n as specified in [15]. Note, from Theorem 1, for a given $d_{\text{B-free}}$, higher encoder memory is required to achieve the same $d_{\text{B-free}}$ as k increases, i.e., given a certain fixed $d_{\text{B-free}}$, there is a tradeoff between complexity and rate. In addition, as is shown in Theorem 2, $N_{\text{B-free}}$ also increases as the rate increases and hence there is also a tradeoff between rate and performance. A search has been performed on rate-1/2, -2/3 and -3/4 codes to find the best ones in terms of $d_{\text{B-free}}$ and $N_{\text{B-free}}$. The results are given in Tables I–III.

An important point to note is that codes optimum in terms of branch distance may not be optimum in terms of the free Hamming distance $d_{\text{H-free}}$ and vice-versa. For small values of γ, it has been observed that codes optimum in terms of branch distance are also optimum in terms of $d_{\text{H-free}}$, however, the same does not hold for higher values of γ. From Table I, we notice that up to $\gamma = 7$, the search yields codes which meet the upper bound in terms of $d_{\text{B-free}}$, however from that point on, the best codes start falling short of the

TABLE II
OPTIMUM BRANCH DISTANCE RATE 2/3 CODES

$\gamma^\dagger$	G	$d_{B\text{-free}}$ ‡	$N_{B\text{-free}}$ △	$d_{H\text{-free}}$ ⊔	$N_{H\text{-free}}$ *
2	$\begin{pmatrix} 6 & 2 & 6 \\ 2 & 4 & 4 \end{pmatrix}_8$	2	4	3	2
4	$\begin{pmatrix} 0 & 4 & 3 \\ 7 & 5 & 0 \end{pmatrix}_8$	3	5	3	1
6	$\begin{pmatrix} 0 & 54 & 64 \\ 54 & 74 & 14 \end{pmatrix}_8$	4	7	6	3
8	$\begin{pmatrix} 76 & 26 & 46 \\ 64 & 0 & 36 \end{pmatrix}_8$	5	14	6	1
10	$\begin{pmatrix} 75 & 57 & 0 \\ 66 & 64 & 55 \end{pmatrix}_8$	6	30	6	1

† Total encoder memory.
‡ Minimum free branch distance.
△ Number of codewords with branch distance $d_{B\text{-free}}$.
⊔ Free Hamming distance.
* Number of codewords with Hamming distance $d_{H\text{-free}}$.
Note: The code generators have been listed in octal, where the octal representation of xyz is $4 \cdot x + 2 \cdot y + z$ and x, y and z denote three binary bits.

TABLE III
OPTIMUM BRANCH DISTANCE RATE 3/4 CODES

$\gamma^\dagger$	G	$d_{B\text{-free}}$ ‡	$N_{B\text{-free}}$ △	$d_{H\text{-free}}$ ⊔	$N_{H\text{-free}}$ *
3	$\begin{pmatrix} 0 & 6 & 6 & 2 \\ 6 & 6 & 2 & 4 \\ 6 & 2 & 2 & 2 \end{pmatrix}_8$	2	11	3	3
6	$\begin{pmatrix} 7 & 1 & 0 & 4 \\ 5 & 7 & 1 & 7 \\ 0 & 5 & 6 & 7 \end{pmatrix}_8$	3	16	5	8
9	$\begin{pmatrix} 74 & 2 & 34 & 0 \\ 44 & 7 & 74 & 74 \\ 54 & 0 & 4 & 74 \end{pmatrix}_8$	4	30	5	1

† Total encoder memory.
‡ Minimum free branch distance.
△ Number of codewords with branch distance $d_{B\text{-free}}$.
⊔ Free Hamming distance.
* Number of codewords with Hamming distance $d_{H\text{-free}}$.
Note: The code generators have been listed in octal, where the octal representation of xyz is $4 \cdot x + 2 \cdot y + z$ and x, y and z denote three binary bits.

upper bound by one. Codes shown in Tables II and III meet the upper bound, however as the complexity increases, $N_{B\text{-free}}$ also starts increasing. Also listed in the tables is the $d_{H\text{-free}}$ and $N_{H\text{-free}}$, the number of codewords with $d_{H\text{-free}}$. The code generators in the tables have been listed in octal with the lowest degree on the left and the highest on the right, e.g., $(622)_8 \equiv 1 + D + D^4 + D^7$. As an example, consider the eighth code listed in Table I. This is a rate-1/2 convolutional code with generators $1 + D + D^4 + D^7$ and $1 + D^2 + D^3 + D^4 + D^5 + D^6 + D^8$ and $d_{B\text{-free}} = 8$.

III. CONSTRUCTION OF MULTIDIMENSIONAL MPSK CODES

The proposed multidimensional MPSK codes are constructed using a q level concatenation approach as shown in Fig. 1. Outer codes in the multilevel concatenation may be either block or convolutional, binary or nonbinary. However, in this paper we will focus on binary convolutional codes as the outer codes.

Outer Codes: The outer code, C_i, at the ith level for $1 \le i \le q$ is chosen to be a convolutional code of rate k_i/n_i with *optimum* branch distance for the given rate and state-complexity. The parameters k_i and n_i depend upon the choice of the inner codes, as will be clear after the discussion of inner codes. Each outer code is selected from the tables mentioned in Section II. The reasons for selecting an *optimum* branch distance convolutional code will be clear when discussing Theorems 4–6.

Inner Codes: Let S denote the two-dimensional MPSK signal constellation which consists of 2^ℓ signal points. Let S^m denote the set of all m-tuples over S, where m is a positive integer. Since S is a two-dimensional signal space, S^m is an $m \times 2$-dimensional signal space in which each signal point is a sequence of m MPSK signals. To construct the proposed codes, the signal space is chosen as a *subspace* of S^m, denoted Λ_0. In this paper, Λ_0 is constructed using the multilevel coding method proposed by Imai and Hirakawa [8].

Using the set partitioning approach proposed by Ungerboeck in [1], each signal point in the set S is labeled by a string of symbols from GF(2). Since S contains 2^ℓ signal points, we shall consider a labeling whose set of label strings is of the following form: $L \triangleq \{a_1 a_2 \cdots a_\ell \colon a_i \in \mathrm{GF}(2) \text{ for } 1 \le i \le \ell\}$. Let λ denote the one-to-one mapping from L to S. If $a_1 a_2 \cdots a_\ell$ is the label for a signal point s, then $s = \lambda(a_1 a_2 \cdots a_\ell)$. Define an addition "+" on the label set L as follows: For two labels, $a_1 a_2 \cdots a_\ell$ and $a'_1 a'_2 \cdots a'_\ell$, in L, $a_1 a_2 \cdots a_\ell + a'_1 a'_2 \cdots a'_\ell = a''_1 a''_2 \cdots a''_\ell$ where $a''_i = a_i \oplus a'_i$ for $1 \le i \le \ell$ and $\oplus$ is the modulo-2 addition. With this addition, L is simply the vector space of all ℓ-tuples over GF(2). We call L the *label space* for S.

For $1 \le i \le \ell$, let $C_{0,i}$ be a binary $(m, k_{0,i}, \delta_{0,i})$ linear block code of length m, dimension $k_{0,i}$ and minimum Hamming distance $\delta_{0,i}$. Let

$$\boldsymbol{V_i} = (v_{i,1}, v_{i,2}, \cdots v_{i,m}) \tag{3.1}$$

be a code word in $C_{0,i}$ for $1 \le i \le \ell$. We form the following sequence:

$$\begin{aligned} &\boldsymbol{V_1} * \boldsymbol{V_2} * \cdots * \boldsymbol{V_\ell} \\ &\quad \triangleq (v_{1,1} v_{2,1} \cdots v_{\ell,1}, v_{1,2} v_{2,2} \cdots v_{\ell,2}, \\ &\qquad \cdots, v_{1,m} v_{2,m} \cdots v_{\ell,m}). \end{aligned} \tag{3.2}$$

For $1 \le j \le m$, we regard $v_{1,j} v_{2,j} \cdots v_{\ell,j}$ as the label for a signal point s_j in the MPSK signal set S. Then $\boldsymbol{V_1} * \boldsymbol{V_2} \cdots * \boldsymbol{V_\ell}$ is simply an m-tuple over the label set L and

$$\begin{aligned} &\lambda(\boldsymbol{V_1} * \boldsymbol{V_2} \cdots * \boldsymbol{V_\ell}) \\ &\quad = (\lambda(v_{1,1} v_{2,1} \cdots v_{\ell,1}), \lambda(v_{1,2} v_{2,2} \cdots v_{\ell,2}), \\ &\qquad \cdots, \lambda(v_{1,m} v_{2,m} \cdots v_{\ell,m})) \\ &\quad = (s_1, s_2, \cdots, s_m) \end{aligned} \tag{3.3}$$

is an m-tuple over the MPSK signal set S (a sequence of

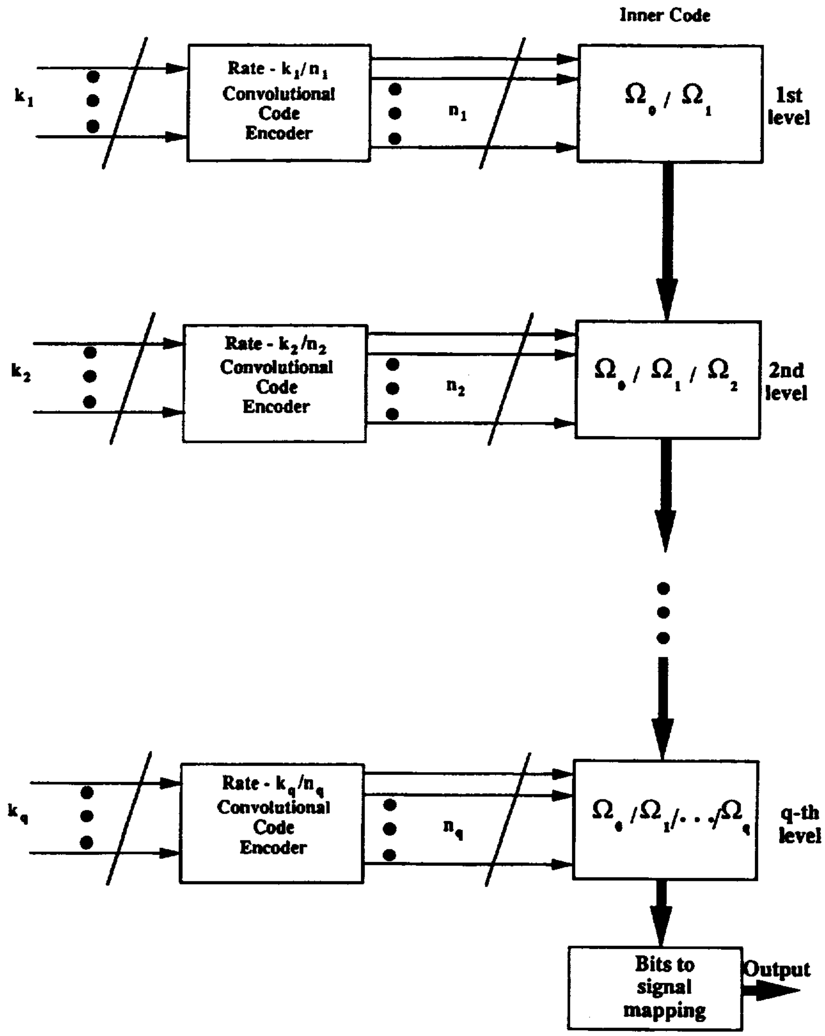


Fig. 1. A multilevel concatenated TCM system.

m MPSK signals) which is a signal point in the $m \times 2$-dimensional signal space S^m. From codes $C_{0,i}$ for $1 \le i \le \ell$, we form the following set of m-tuples over the label set L:

$$\begin{aligned} &C_{0,1} * C_{0,2} * \cdots * C_{0,\ell} \\ &\quad = \{\boldsymbol{V_1} * \boldsymbol{V_2} * \cdots * \boldsymbol{V_\ell}: \boldsymbol{V_1} \in C_{0,1}, \boldsymbol{V_2} \in C_{0,2}, \\ &\quad \cdots \boldsymbol{V_\ell} \in C_{0,\ell}\}. \end{aligned} \tag{3.4}$$

We will denote $C_{0,1} * C_{0,2} * \cdots * C_{0,\ell}$ by Ω_0. Then, Ω_0 is a vector space (or a linear code) over L (a subspace of the vector space of all the m tuples over L, denoted L^m). Ω_0 has $2^{k_{0,1}+k_{0,2}+\cdots+k_{0,\ell}}$ vectors. Hence, the dimension of Ω_0 is $\sigma_0 = k_{0,1} + k_{0,2} + \cdots + k_{0,\ell}$. Recall, that for $1 \le i \le q, n_i$ denotes the number of output coded bits of the convolutional encoder at the ith stage of encoding. Choose

$$n_1 + n_2 + \cdots + n_q = k_{0,1} + k_{0,2} + \cdots + k_{0,\ell} = \sigma_0 \tag{3.5}$$

Suppose each m-tuple in Ω_0 is mapped into an m-tuple over the MPSK signal set S by the mapping $\lambda(\cdot)$. Then, we obtain the following subset of signal points in S^m:

$$\begin{aligned} \Lambda_0 &\triangleq \lambda(\Omega_0) \\ &= \{\lambda(\boldsymbol{V_1} * \boldsymbol{V_2} \cdots * \boldsymbol{V_\ell}): \boldsymbol{V_1} \in C_{0,1}, \boldsymbol{V_2} \in C_{0,2} \\ &\quad \cdots \boldsymbol{V_\ell} \in C_{0,\ell}\}. \end{aligned}$$

The set Λ_0 is a subspace of S^m with dimension σ_0. This subspace Λ_0 is actually a *basic* ℓ-level block MPSK modulation code of length m [8]–[14].

The performance of Λ_0 over the AWGN channel depends upon the minimum squared Euclidean distance and the number of nearest neighbors. The minimum squared Euclidean distance of Λ_0 can be calculated using results of [12]. On the other hand, the performance of Λ_0 over fading channels depends upon the minimum symbol distance, product distance, number of nearest neighbors and the squared Euclidean distance to a lesser extent [17]. The minimum symbol distance of Λ_0 is given by [17] $\delta_H^0 = \min_{i=1}^{\ell} \delta_{0,i}$. Suppose, Λ_0 has minimum squared Euclidean distance Δ_0^2 and minimum symbol distance δ_H^0.

In the following, the subspace Λ_0 of S^m will be used as the *signal space* for constructing multidimensional trellis MPSK codes. Before presenting the code construction, we need to define a subspace of Ω_0 for partitioning Ω_0. For $1 \leq j \leq \ell$, let $C_{1,j}, C_{2,j}, \cdots, C_{q,j}$ be a sequence of linear subcodes of $C_{0,j}$ such that

$$C_{q,j} \subseteq C_{q-1,j} \subseteq \cdots \subseteq C_{1,j} \subseteq C_{0,j}. \quad (3.6)$$

Let $k_{i,j}$ be the dimension and $\delta_{i,j}$ be the minimum Hamming distance of $C_{i,j}$ for $1 \leq i \leq q$. Then $C_{i,j}$ is an $(m, k_{i,j}, \delta_{i,j})$ code. For $1 \leq i \leq q$, we form the following linear code over the labeling space L: $\Omega_i = C_{i,1} * C_{i,2} * \cdots * C_{i,\ell}$. The dimension of this code is $\sigma_i = k_{i,1} + k_{i,2} + \cdots + k_{i,\ell}$. It is clear that for $1 \leq i \leq q$,

$$\Omega_i \subseteq \Omega_{i-1}. \quad (3.7)$$

It follows from (3.7) that $\Omega_1, \Omega_2, \cdots, \Omega_q$ form a sequence of subspaces of Ω_0 and

$$\Omega_q \subseteq \Omega_{q-1} \subseteq \cdots \subseteq \Omega_1 \subseteq \Omega_0. \quad (3.8)$$

For $1 \leq i \leq q$, let

$$\Lambda_i \triangleq \lambda(\Omega_i). \quad (3.9)$$

Then, Λ_i is a subspace of S^m with dimension $\dim(\Lambda_i) = \sigma_i$. Let the minimum squared Euclidean distance of Λ_i be Δ_i^2 and minimum symbol distance be δ_H^i. Equations (3.8) and (3.9) imply that $\Lambda_1, \Lambda_2, \cdots, \Lambda_q$ form a sequence of subspaces of Λ_0 and

$$\Lambda_q \subseteq \Lambda_{q-1} \subseteq \cdots \subseteq \Lambda_1 \subseteq \Lambda_0. \quad (3.10)$$

Suppose the binary codes, $C_{i,j}$ with $1 \leq i \leq q$ and $1 \leq j \leq \ell$, are chosen such that

$$n_i = \sigma_{i-1} - \sigma_i. \quad (3.11)$$

It follows from (3.5) and (3.11) that

$$\begin{aligned} \sigma_1 &= n_2 + n_3 + \cdots + n_q \\ \sigma_2 &= n_3 + \cdots + n_q \\ &\vdots \\ \sigma_{q-1} &= n_q \\ \sigma_q &= 0 \end{aligned}$$

Ω_0 and its subcodes $\Omega_1, \Omega_2, \cdots, \Omega_q$ are used to form a sequence of coset codes [7]. Let $\boldsymbol{U_1} * \boldsymbol{U_2} * \cdots * \boldsymbol{U_\ell}$ be a vector in Ω_0 but not in Ω_1. Then $\boldsymbol{U_1} * \boldsymbol{U_2} \cdots * \boldsymbol{U_\ell} + \Omega_1$ is a *coset* of Ω_1 in Ω_0 and $\boldsymbol{U_1} * \boldsymbol{U_2} * \cdots * \boldsymbol{U_\ell}$ is called the *coset representative*. Recall $n_1 = \Sigma_{i=1}^{\ell}(k_{0,i} - k_{1,i})$. Hence, there are 2^{n_1} cosets of Ω_1 in Ω_0. These 2^{n_1} cosets of Ω_1 form a *partition* of Ω_0. Let Ω_0/Ω_1 denote the set of cosets in Ω_0 modulo Ω_1. Ω_0/Ω_1 is called a *coset code*. Let $[\Omega_0/\Omega_1]$ denote the set of coset representatives of the coset code Ω_0/Ω_1. Hence $\Omega_0/\Omega_1 = [\Omega_0/\Omega_1] + \Omega_1$. Ω_1 can be further partitioned using Ω_2, in the same way as is outlined above. Partitioning each coset of Ω_1 in Ω_0 on the basis of Ω_2, we form the coset code $\Omega_0/\Omega_1/\Omega_2$. Let $[\Omega_1/\Omega_2]$ denote the set of coset representatives in the partition Ω_1/Ω_2. Hence each coset in the coset code $\Omega_0/\Omega_1/\Omega_2$ can be written in the form $[\Omega_0/\Omega_1] + [\Omega_1/\Omega_2] + \Omega_2$. Proceeding in this manner, we form the following sequence of coset codes:

$$\begin{aligned} B_1 &= \Omega_0/\Omega_1 \\ B_2 &= \Omega_0/\Omega_1/\Omega_2 \\ &\vdots \\ B_q &= \Omega_0/\Omega_1/\Omega_2/\cdots/\Omega_q. \end{aligned}$$

For $1 \leq i \leq q$, each coset in $B_{i-1} = \Omega_0/\Omega_1/\cdots/\Omega_{i-1}$ consists of 2^{n_i} cosets modulo Ω_i. These coset codes are used as the inner codes in the multilevel concatenation in which B_1 is used at the first level and B_q at the qth level.

Let ω_0 and ω_0' be two distinct points in Ω_0. If these two points are in two distinct cosets of B_1 then the squared Euclidean distance between $s = \lambda(\omega_0)$ and $s' = \lambda(\omega_0')$ is at least Δ_0^2. If the two points ω_0 and ω_0' are in the same coset of B_1 but distinct cosets of B_2, then the squared Euclidean distance between s and s' is at least Δ_1^2. Generalizing in this manner, it is easy to see that if the two points ω_0 and ω_0' have identical coset representatives in B_j for $1 \leq j < i$, but distinct coset representatives for B_i then s and s' have a squared Euclidean distance of at least Δ_{i-1}^2. Hence, B_1 is the least powerful and B_q is the most powerful coset code in terms of Euclidean distance.

The same arguments as above will also hold if the minimum squared Euclidean distance at each stage is replaced by the corresponding minimum symbol distance.

Encoding of $m \times 2$–Dimensional TCM Code: Encoding is accomplished in q stages, as shown in Fig. 1, and for $1 \leq i \leq q$, the ith level encoding is accomplished in two steps: 1) at any time instant t, a message of k_i bits is encoded based on the convolutional outer code C_i into an n_i-bit coded block; and 2) the n_i-bit code block then selects a coset from the coset code $B_i = \Omega_0/\Omega_1/\cdots/\Omega_i$.

The output at the ith level encoder is a sequence of cosets from B_i. All the possible coset sequences at the ith level form a trellis, and each branch in the trellis corresponds to a coset in B_i, and this trellis is isomorphic to the trellis of C_i. Let $\boldsymbol{v_i}$ denote a code sequence in the convolutional code C_i and let ϕ_i denote the mapping from the n_i coded output bits of the convolutional code to the 2^{n_i} cosets. Hence, $\phi_i(\boldsymbol{v_i})$ denotes the sequence of coset representatives at the ith stage of encoding, corresponding to the code sequence $\boldsymbol{v_i}$. Hence, any code sequence in the $m \times 2$-dimensional TCM code can be written in the form

$$\lambda(\phi_1(\boldsymbol{v_1}) + \phi_2(\boldsymbol{v_2}) + \cdots + \phi_q(\boldsymbol{v_q})). \quad (3.12)$$

At every time instant t, the encoder puts out m MPSK signals.

A very interesting and special case of the proposed codes occurs when $q = 2$ and the second level outer code is left uncoded, as shown in Fig. 2. This structure is *equivalent* to the structure used for the construction of the multidimensional codes in [3]. A computer search was used in [3] to find the convolutional code to be used at the first level. The computer search selected a convolutional code which optimized the

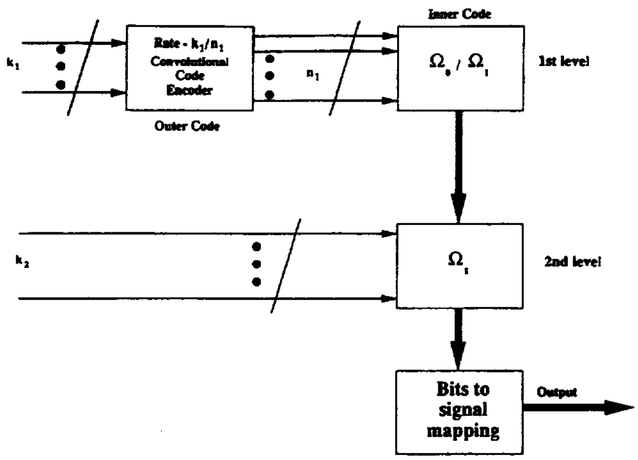


Fig. 2. A two-level concatenated TCM system.

multidimensional code both in terms of Euclidean distance and number of nearest neighbors.

A multidimensional code is said to be *linear* with respect to binary addition, if for any two code sequences in the multidimensional code, $\boldsymbol{U} = \lambda(\phi_1(\boldsymbol{u_1}) + \phi_2(\boldsymbol{u_2}) + \cdots + \phi_q(\boldsymbol{u_q}))$ and $\boldsymbol{V} = \lambda(\phi_1(\boldsymbol{v_1}) + \phi_2(\boldsymbol{v_2}) + \cdots + \phi_q(\boldsymbol{v_q}))$

$$\boldsymbol{U} \oplus \boldsymbol{V} \triangleq \lambda((\phi_1(\boldsymbol{u_1}) + \phi_2(\boldsymbol{u_2}) + \cdots + \phi_q(\boldsymbol{u_q})) + (\phi_1(\boldsymbol{v_1}) + \phi_2(\boldsymbol{v_2}) + \cdots + \phi_q(\boldsymbol{v_q})))$$

is also a code sequence, where $\boldsymbol{u_i}$ and $\boldsymbol{v_i}$ for $1 \leq i \leq q$ denote output code sequences of the convolutional code encoder C_i at the ith level. Linearity of the code (in terms of binary addition) simplifies the error analysis and in addition leads to a simpler encoder and decoder. The linear structure leads to the following theorems on the linearity, minimum squared Euclidean distance, and minimum symbol distance of the proposed codes.

Theorem 3: A multidimensional code is linear with respect to binary addition, if all the mappings ϕ_i, for $1 \leq i \leq q$ are linear.

Proof: Recall, that any code sequence in a multidimensional TCM code can be written in the form $\lambda(\phi_1(\boldsymbol{v_1}) + \phi_2(\boldsymbol{v_2}) + \cdots + \phi_q(\boldsymbol{v_q}))$ where $\boldsymbol{v_i}$ for $1 \leq i \leq q$ denotes the output code sequence of the convolutional code C_i at the ith level. The proof then follows trivially from the definition of linearity. △△

Theorem 4: The minimum free squared Euclidean distance of a coset trellis code at the jth level, for $1 \leq j \leq q$ is lower bounded by $D^2_{(j),\text{free}} \geq \Delta^2_{j-1} \cdot d^{(j)}_{\text{B-free}}$, where $d^{(j)}_{\text{B-free}}$ denotes the minimum free branch distance of the convolutional code at the jth level, C_j.

Proof: Consider two distinct code sequences, $\boldsymbol{U} = \lambda(\phi_1(\boldsymbol{u_1}) + \phi_2(\boldsymbol{u_2}) + \cdots + \phi_q(\boldsymbol{u_q}))$ and $\boldsymbol{V} = \lambda(\phi_1(\boldsymbol{v_1}) + \phi_2(\boldsymbol{v_2}) + \cdots + \phi_q(\boldsymbol{v_q}))$, where $\boldsymbol{u_i}$ and $\boldsymbol{v_i}$ for $1 \leq i \leq q$ denotes two output code sequences of the convolutional code C_i at the ith level. Assume that $\boldsymbol{u_i} = \boldsymbol{v_i}$ for $1 \leq i < j$ and $\boldsymbol{u_j} \neq \boldsymbol{v_j}$. At a particular time instant t, let $\lambda(\omega)$ and $\lambda(\omega')$ be the corresponding transmitted signal points for $\boldsymbol{U}$ and $\boldsymbol{V}$, respectively, where ω and $\omega' \in \Omega_0$. Since $\boldsymbol{u_i} = \boldsymbol{v_i}$ for $1 \leq i < j$ and $\boldsymbol{u_j} \neq \boldsymbol{v_j}$, hence ω and ω' have identical coset representatives in B_i for $1 \leq i < j$ and hence the minimum squared Euclidean distance between $\lambda(\omega)$ and $\lambda(\omega')$ is at least Δ^2_{j-1}. Since C_j has minimum free branch distance $d^{(j)}_{\text{B-free}}$, hence the two sequences $\boldsymbol{u_j}$ and $\boldsymbol{v_j}$ are distinct in at least $d^{(j)}_{\text{B-free}}$ branches. Therefore, the squared Euclidean distance between $\boldsymbol{U}$ and $\boldsymbol{V}$ is at least $\Delta^2_{j-1} \cdot d^{(j)}_{\text{B-free}}$. △△

Theorem 5: The minimum free squared Euclidean distance of the overall TCM code is lower bounded by $D^2_{\text{free}} \geq \min_{1 \leq j \leq q}\{\Delta^2_{j-1} \cdot d^{(j)}_{\text{B-free}}\}$.

Proof: Consider two distinct code sequences $\boldsymbol{U}$ and $\boldsymbol{V}$. Using the same notation as developed in Theorem 4, consider that $\boldsymbol{u_i} = \boldsymbol{v_i}$ for $1 \leq i < j$ and that $\boldsymbol{u_j} \neq \boldsymbol{v_j}$. Then, Theorem 4 gives us the minimum squared Euclidean distance between the two sequences. Since j is arbitrary, the minimum squared Euclidean distance between the two sequences is obtained by taking the minimum over all the q levels, i.e., if $D^2(\boldsymbol{U}, \boldsymbol{V})$ denotes the squared Euclidean distance between the two sequences $\boldsymbol{U}$ and $\boldsymbol{V}$, then $D^2(\boldsymbol{U}, \boldsymbol{V}) \geq \min_{1 \leq j \leq q}\{\Delta^2_{j-1} \cdot d^{(j)}_{\text{B-free}}\}$. Since $\boldsymbol{U}$ and $\boldsymbol{V}$ are any two sequences, the Theorem follows. △△

Theorem 6: The minimum symbol distance of the overall TCM code is lower bounded by $\delta_H \geq \min_{1 \leq j \leq q}\{\delta^j_H \cdot d^{(j)}_{\text{B-free}}\}$.

Proof: The proof is similar to that in Theorem 5, with the only difference that instead of minimum squared Euclidean distance we now consider minimum symbol distance. △△

IV. Code Properties

A. Spectral Efficiency

At each encoding time instant, $k_1 + k_2 + \cdots + k_q$ bits are fed into the encoder (Fig. 1), and the corresponding output is m MPSK signals. Hence the spectral efficiency of the $m \times 2$-dimensional TCM code is $(k_1 + k_2 + \cdots + k_q)/m$ bits/symbol.

B. Phase Invariance

Phase symmetry of a code is important in resolving carrier-phase ambiguity and ensuring rapid carrier-phase resynchronization after temporary loss of synchronization [2]. It is desirable for a modulation code to have as many phase symmetries as possible. Recall, that the proposed multidimensional modulation codes are constructed using q convolutional codes and $q + 1$ basic ℓ-level block modulation codes (Fig. 1). The phase invariance of the proposed codes is a function of both the inner codes and the outer codes. If convolutional codes are used *at all the q levels*, the phase invariance of the constructed modulation codes would depend upon the structure of the convolutional codes used, and for *most* cases the constructed modulation codes would have no phase invariance. A special case of the proposed codes occurs when the outer code at the qth level is left uncoded (Fig. 2 shows this special case for $q = 2$). Most of the codes constructed using this special case *do have* phase invariance. Kasami *et al.* derived conditions on phase invariance of basic ℓ-level block modulation codes [16]. A slightly modified form of the conditions proposed in [16] will be applicable to the proposed codes.

The following theorem gives the conditions for the proposed modulation codes to be phase invariant under rotation for this special case.

Theorem 7: Let $\Lambda_0 = \lambda(C_{0,1} * C_{0,2} * \cdots * C_{0,\ell})$ and let $\Lambda_{q-1} = \lambda(C_{q-1,1} * C_{q-1,2} * \cdots * C_{q-1,\ell})$, where $C_{0,i}$ and $C_{q-1,i}$ for $1 \leq i \leq \ell$ are binary linear block codes of length m. For $1 \leq h \leq \ell$, the multidimensional MPSK TCM code is phase invariant under $180°/2^{\ell-h}$ phase shifts if the multidimensional TCM code is linear with respect to binary addition and

$$1 \in C_{q-1,h} \quad \text{and} \tag{4.1}$$

$$C_{0,h} \cdot C_{0,h+1} \cdots C_{0,j-1} \subseteq C_{q-1,j} \quad \text{for} \quad h < j \leq \ell \tag{4.2}$$

where 1 denotes the all-one binary sequence of length m, and for two-binary m-tuples $\boldsymbol{a} = (a_1, a_2, \cdots a_m)$ and $\boldsymbol{b} = (b_1, b_2, \cdots b_m), \boldsymbol{a} \cdot \boldsymbol{b} \triangleq (a_1 \cdot b_1, a_2 \cdot b_2, \cdots, a_m \cdot b_m)$, where $a_i \cdot b_i$, for $1 \leq i \leq m$ denotes the logical product of a_i and b_i.

Proof Appendix:

If the outer code at the qth level is left uncoded, sequences of signal points from Λ_{q-1} are valid code sequences. The best phase invariance that can be achieved for the overall multidimensional code in this case is equal to the phase invariance of Λ_{q-1}. The conditions as stated in Theorem 7 provide a set of conditions which guarantee a certain phase invariance for the overall multidimensional MPSK TCM code independent of the convolutional codes chosen. Most codes designed using the proposed technique, do achieve the best possible phase invariance (i.e., of Λ_{q-1}).

V. Multistage Decoding Algorithm

One obvious way of decoding a TCM code proposed in Section III, is to form a super trellis for the code, which is obtained by taking the direct product of the trellises of the convolutional codes at the q levels. The complexity associated with this technique (for most cases) would be tremendous. We will focus on a multistage decoding scheme, in which the decoding is carried out in q stages, corresponding to the q levels of the multidimensional TCM code. Let $\boldsymbol{V} = (s_1, s_2, s_3, \cdots)$ be the transmitted code sequence, where s_i for $1 \leq i \leq \infty$ denotes a signal point in the MPSK signal constellation and let $\boldsymbol{R} = (r_1, r_2, r_3, \cdots)$ denote the corresponding received sequence. Using (3.12), $\boldsymbol{V}$ can be written in the form $\boldsymbol{V} = \lambda(\phi_1(\boldsymbol{v_1}) + \phi_2(\boldsymbol{v_2}) + \cdots + \phi_q(\boldsymbol{v_q}))$ where $\boldsymbol{v_i}$ for $1 \leq i \leq q$ denotes a code sequence in the convolutional code C_i.

First Stage of Decoding: At the first stage, $\boldsymbol{v_1}$ is estimated using the received sequence $\boldsymbol{R}$. Recall, that at the first stage of encoding, the trellis is *isomorphic* to the trellis of the convolutional code C_1 used at the first level, with each branch of the trellis corresponding to a coset in B_1. Each coset in B_1 can be written in the general form $\omega_0 + \Omega_1$, where $\omega_0 \in [\Omega_0/\Omega_1]$. Let us call this *isomorphic* trellis $\tilde{C}_1$. Hence, each branch of $\tilde{C}_1$ consists of 2^{σ_1} points, corresponding to the 2^{σ_1} points in Ω_1. The trellis $\tilde{C}_1$ is used to form the trellis $\lambda(\tilde{C}_1)$, where

$$\lambda(\tilde{C}_1) \triangleq \{\lambda(\boldsymbol{v}) : \boldsymbol{v} \in \tilde{C}_1\}. \tag{5.1}$$

The trellis $\lambda(\tilde{C}_1)$ will be used for decoding at the first stage. Any code sequence in $\lambda(\tilde{C}_1)$ can be written in the form

$$\lambda(\phi_1(\boldsymbol{u_1}) + \boldsymbol{\omega}_1) \tag{5.2}$$

where $\boldsymbol{u_1}$ is a code sequence in C_1 and $\boldsymbol{\omega}_1$ is a sequence of points from Ω_1, i.e., $\boldsymbol{\omega}_1 = \{(\omega_{1,1}, \omega_{1,2}, \omega_{1,3}, \cdots) : \omega_{1,i} \in \Omega_1 \text{ for } 1 \leq i \leq \infty\}$. Standard soft-decision Viterbi decoding[1] is performed on $\boldsymbol{R}$ using the trellis $\lambda(\tilde{C}_1)$. This yields a code sequence $\lambda(\phi_1(\hat{\boldsymbol{v}}_1) + \hat{\boldsymbol{\omega}}_1)$ in $\lambda(\tilde{C}_1)$ which is closest to the received sequence $\boldsymbol{R}$ in terms of squared Euclidean distance. The code sequence $\hat{\boldsymbol{v}}_1$ forms an estimate of the sequence $\boldsymbol{v_1}$. $\hat{\boldsymbol{\omega}}_1$ denotes a sequence of points from Ω_1. Since $\hat{\boldsymbol{v}}_1$ is a code sequence in C_1, the estimate of the information sequence associated with the first level can be obtained from $\hat{\boldsymbol{v}}_1$.

The ith Stage of Decoding: The second and subsequent stages of decoding are very similar to the first stage of decoding. For $2 \leq i \leq q$, let us consider the ith stage of decoding. The previous $i-1$ stages of decoding give us estimates of $\boldsymbol{v_j}$, denoted by $\hat{\boldsymbol{v}}_j$ for $1 \leq j \leq (i-1)$. Using arguments similar to that given above, we form the *isomorphic* trellis $\tilde{C}_i$, where any code sequence in $\tilde{C}_i$ can be expressed in the general form

$$\phi_1(\hat{\boldsymbol{v}}_1) + \phi_2(\hat{\boldsymbol{v}}_2) + \cdots + \phi_{i-1}(\hat{\boldsymbol{v}}_{i-1}) + \phi_i(\boldsymbol{u_i}) + \boldsymbol{\omega}_i \tag{5.3}$$

where $\boldsymbol{u_i}$ is a code sequence in the convolutional code at the ith level, C_i and $\boldsymbol{\omega}_i$ is sequence of points from Ω_i. Each branch of $\tilde{C}_i$ consists of 2^{σ_i} points, corresponding to the number of points in Ω_i. The trellis $\tilde{C}_i$ is used to form the trellis $\lambda(\tilde{C}_i)$, where

$$\lambda(\tilde{C}_i) \triangleq \{\lambda(\boldsymbol{v}) : \boldsymbol{v} \in \tilde{C}_i\}. \tag{5.4}$$

The trellis $\lambda(\tilde{C}_i)$ will be used for decoding at the ith stage. Standard soft-decision Viterbi decoding is performed on $\boldsymbol{R}$ using the trellis $\lambda(\tilde{C}_i)$. This yields a code sequence

$$\lambda(\phi_1(\hat{\boldsymbol{v}}_1) + \phi_2(\hat{\boldsymbol{v}}_2) + \cdots + \phi_{i-1}(\hat{\boldsymbol{v}}_{i-1}) + \phi_i(\hat{\boldsymbol{v}}_i) + \hat{\boldsymbol{\omega}}_i) \tag{5.5}$$

in $\lambda(\tilde{C}_i)$ which is closest to the received sequence $\boldsymbol{R}$ in terms of squared Euclidean distance, where $\hat{\boldsymbol{v}}_i$ is a code sequence in the convolutional code used at the ith level, C_i, and $\hat{\boldsymbol{\omega}}_i$ is a sequence of points from Ω_i. The code sequence $\hat{\boldsymbol{v}}_i$ forms an estimate of the sequence $\boldsymbol{v}_i$. Since $\hat{\boldsymbol{v}}_i$ is a code sequence in C_i, the information sequence associated with the ith level can be obtained from $\hat{\boldsymbol{v}}_i$.

The branch metric (squared Euclidean distance) for each branch in $\lambda(\tilde{C}_i), 1 \leq i \leq q$, is calculated by taking the m received signals corresponding to that branch and finding the element in the coset corresponding to that branch, which is closest to the m received signals in terms of Euclidean distance. This process of finding the closest element in the coset is termed as *closest coset decoding*. The Euclidean distance corresponding to the closest element in the coset becomes the branch metric. If m is small, calculation of the

[1] We will use minimum squared Euclidean distance as the decoding metric for both the AWGN and fading channels.

branch metric does not represent a formidable task, however if m is large and if $\Omega_i, 1 \leq i \leq q$, has trellis structure then a trellis can be used to calculate the branch metric. In addition, if the number of states associated with the trellis structure of Ω_i is big, multistage decoding for Ω_i can be used to further reduce the decoding complexity. Multistage decoding of Ω_i would be carried out in the same way as proposed in [10] and [11].

Another way of reducing the decoding complexity associated with *closest coset decoding* would be as follows: Consider a trellis $\tilde{C}_i^{\text{sup}}$, where any code sequence in the trellis $\tilde{C}_i^{\text{sup}}$ can be written in the following form:

$$\phi_1(\hat{\boldsymbol{v}}_1) + \phi_2(\hat{\boldsymbol{v}}_2) + \cdots + \phi_{i-1}(\hat{\boldsymbol{v}}_{i-1}) + \phi_i(\boldsymbol{u_i}) + \boldsymbol{\omega}_i^{\text{sup}} \quad (5.6)$$

where $\boldsymbol{\omega}_i^{\text{sup}}$ is a sequence of points from $\boldsymbol{\Omega}_i^{\text{sup}}$, and the rest of the sequences are as before. If $\Omega_i \subset \Omega_i^{\text{sup}}$ then the trellis $\tilde{C}_i$ is a subcode of the trellis $\tilde{C}_i^{\text{sup}}$. As such, instead of using $\tilde{C}_i$ we can use $\tilde{C}_i^{\text{sup}}$ at the ith stage of decoding. Ω_i^{sup} can be chosen to have a simpler trellis structure as compared to that of Ω_i. This would reduce the complexity associated with *closest coset decoding* and hence reduce the decoding complexity associated with the ith stage of decoding.

Multistage decoding leads to error propagation. To reduce the effect of error propagation, the first couple of decoding stages should be powerful. A special case of the decoding algorithm occurs for $q = 2$ and $k_2 = n_2$. If *closest coset decoding* at the first stage is carried out in a single-stage, then the overall decoding of the multidimensional code is also one-stage. If m is small, then one-stage *closest coset decoding* is feasible, however if m is large, multistage *closest coset decoding* could be adopted to reduce the decoding complexity. The overall decoding in the latter case would then be multistage.

Decoding Complexity of the Proposed Decoding Algorithm: The complexity of the proposed schemes will be measured in terms of the number of computations required for the decoder to produce an estimate of each 2-D PSK signal. For $1 \leq i \leq q$, let γ_i be the total encoder memory of the convolutional code used at the ith level in the proposed scheme. Consider the ith stage of decoding. Then, due to the Viterbi algorithm alone, the complexity is $2^{\gamma_i + k_i}$ additions and $2^{\gamma_i}(2^{k_i} - 1)$ comparisons, per $m \times 2$-dimensions (since each branch has m MPSK signals). The branch metric calculation forms an additional complexity and depends upon the choice of the inner codes. Let us call this complexity B_{C_i}. Hence the total complexity per $m \times 2$ dimensions is: 1) $\Sigma_{i=1}^{q} 2^{\gamma_i + k_i}$ additions, 2) $\Sigma_{i=1}^{q} 2^{\gamma_i}(2^{k_i} - 1)$ comparisons, and 3) $\Sigma_{i=1}^{q} B_{C_i}$. Dividing this total complexity by m would give us the number of computations required per two dimensions (i.e., the number of computations required to decode a single MPSK point).

VI. Design Rules For Good Codes

The performance of codes designed using the proposed technique depends upon various factors. If all the design considerations are followed strictly, the codes usually would achieve good performance and in some cases, with reduced decoding complexity. Some of the most important design considerations are: 1) the number of levels q, in the multilevel concatenation should be kept as low as possible. The advantages of this are twofold. First, reducing the number of encoding levels, would reduce the number of decoding stages and in most cases reduce the decoding complexity. Second, reducing the number of decoding levels also decreases the amount of error propagation which occurs as a result of the multistage decoding. To reduce the error propagation due to multistage decoding, the first few levels should be chosen extremely powerful, so that the amount of error propagation is decreased. This however leads to higher decoding complexity for the first few levels. 2) The number of dimensions, i.e., $m \times 2$, should be kept as low as possible. As m increases the number of nearest neighbors associated with the code also start increasing, which limits the performance of the code. On the other hand, increasing m usually helps in decreasing the normalized decoding complexity associated with the code. 3) Theorem 5 gives us the minimum squared Euclidean distance of the overall multidimensional TCM code. For a given minimum squared Euclidean distance of the TCM code, $d_{\text{B-free}}$ of the convolutional codes chosen to form the multidimensional TCM code should be chosen to be as small as possible. Lower $d_{\text{B-free}}$ would imply lower decoding complexity associated with the convolutional code decoding. The above also holds for Theorem 6. 4) The branch computation complexity B_{C_i} at the ith stage of decoding depends upon Λ_i. If Λ_i is chosen to have a simple trellis structure, the corresponding branch computation complexity will be minimal. If on the other hand, the trellis for Λ_i is sufficiently complex, techniques described in Section V can be used to reduce the computation complexity. These techniques however, usually lead to degraded performance. 5) Construction of codes with good phase invariance, places restrictions on codes as per Theorem 7 and hence in most cases this would limit either the performance and/or the achievable spectral efficiency.

Most design considerations mentioned above lead to conflicting requirements. Hence, there is a tradeoff involved between performance, decoding complexity, spectral efficiency and phase invariance.

Appendix

Proof of Theorem 7: The proof follows very closely the derivation of the phase invariance conditions in [16]. For the code to be phase invariant by $180°/2^{\ell-h}$, any code sequence in the multidimensional code when rotated by $180°/2^{\ell-h}$ should produce another code sequence. Let $\boldsymbol{V}$ be the transmitted code sequence. Let $\boldsymbol{V}^{\text{rot}}$ denote the code sequence $\boldsymbol{V}$ rotated by $180°/2^{\ell-h}$. Recall from Section III, that the basic building block of the proposed multidimensional codes is Λ_0, hence any valid code sequence in the multidimensional code can be considered to be a sequence of points from Λ_0. Consider the jth time instant. Let $\boldsymbol{V_j} = \lambda(\boldsymbol{V_{1,j}} * \boldsymbol{V_{2,j}} * \cdots * \boldsymbol{V_{\ell,j}})$ be the transmitted sequence of m MPSK signals at the jth time instant, where, $\boldsymbol{V_{i,j}} \in C_{0,i}$ for $1 \leq i \leq \ell$. Also, let $\boldsymbol{V_j^{\text{rot}}} = \lambda(\boldsymbol{V_{1,j}^{\text{rot}}} * \boldsymbol{V_{2,j}^{\text{rot}}} * \cdots * \boldsymbol{V_{\ell,j}^{\text{rot}}})$ be the sequence of m MPSK signals for $\boldsymbol{V}^{\text{rot}}$ at the jth time instant, where, $\boldsymbol{V_{i,j}^{\text{rot}}} \in C_{0,i}$

for $1 \le i \le \ell$. Using results of [16], V_j^{rot} can be written in the following form:

$$V_j^{\text{rot}} = \lambda((V_{1,j} + V'_{1,j}) * (V_{2,j} + V'_{2,j}) * \cdots * (V_{\ell,j} + V'_{\ell,j})) \quad \text{(A.1)}$$

where $V'_{i,j} = 0$ for $1 \le i < h$, $V'_{h,j} = 1$ and $V'_{i,j} = V_{h,j} \cdot V_{h+1,j} \cdots V_{i-1,j}$ for $h < i \le \ell$ and $\mathbf{0}$ denotes the all-zero sequence of length m. Form the sequence V', such that the jth time instant of V' is

$$V'_j = \lambda(V'_{1,j} * V'_{2,j} * \cdots * V'_{\ell,j}). \quad \text{(A.2)}$$

Then, for the code to be phase invariant under rotations of $180°/2^{\ell-h}, V'$ should also be a valid code sequence. Sequences of signal points from Λ_{q-1} form a valid code sequence. Hence, if $V'_j \in \Lambda_{q-1}$ then V is phase invariant under rotations of $180°/2^{\ell-h}$, i.e., if $\mathbf{1} \in C_{q-1,h}$ and

$$V_{h,j} \cdot V_{h+1,j} \cdots V_{i-1,j} \in C_{q-1,i} \quad \text{for} \quad h < i \le \ell \quad \text{(A.3)}$$

then V is phase invariant under phase rotations of $180°/2^{\ell-h}$. Since the above should hold for any transmitted sequence V, the Theorem follows.

References

[1] G. Ungerboeck, "Channel coding with multilevel/phase signals," *IEEE Trans. Inform. Theory*, vol. IT-28, no. 1, pp. 55–67, Jan. 1982.

[2] ——, "Trellis-coded modulation with redundant signal sets: Parts I and II," *IEEE Commun. Mag.*, vol. 25, pp. 5–21, Feb. 1987.

[3] S. S. Pietrobon *et al.*, "Trellis coded multi-dimensional phase modulation," *IEEE Trans. Inform. Theory*, vol. IT-36, no. 1, pp. 63–89, Jan. 1990.

[4] A. R. Calderbank and N. J. A. Sloane, "New trellis codes based on lattices and cosets," *IEEE Trans. Inform. Theory*, vol. IT-33, pp. 177–195, Mar. 1987.

[5] A. R. Calderbank and J. E. Mazo, "A new description of trellis codes," *IEEE Trans. Inform. Theory*, vol. IT-30, pp. 784–791, Nov. 1984.

[6] L. F. Wei, "Trellis-coded modulation with multi-dimensional constellations," *IEEE Trans. Inform. Theory*, vol. IT-33, pp. 483–501, July 1987.

[7] G. D. Forney, Jr., "Coset codes—Part I: Introduction and geometrical classification," *IEEE Trans. Inform. Theory*, vol. 34, pp. 1123–1151, Sept. 1988.

[8] H. Imai and S. Hirakawa, "A new multilevel coding method using error correcting codes," *IEEE Trans. Inform. Theory*, vol. IT-23, pp. 371–376, May 1977.

[9] V. V. Ginzburg, "Multidimensional signals for a continuous channel," *Problm. Pered. Inform.*, vol. 20, no. 1, pp. 28–46, 1984.

[10] S. I. Sayegh, "A class of optimum block codes in signal space," *IEEE Trans. Commun.*, vol. COM-30, no. 10, pp. 1043–1045, Oct. 1986.

[11] R. M. Tanner, "Algebraic construction of large euclidean distance combined coding modulation systems," in *Abstract of Papers, 1986 IEEE International Symposium on Information Theory*, Ann Harbor, Oct. 6–9, 1986.

[12] T. Kasami *et al.*, "On multi-level block modulation codes," *IEEE Trans. Inform. Theory*, vol. 37, no. 4 , pp. 965–975, July 1991.

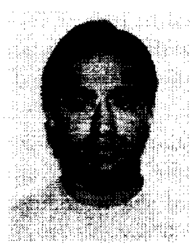
[13] A. R. Calderbank, "Multi-level codes and multi-stage decoding," *IEEE Trans. Commun.*, vol. 37, no. 3, pp. 222–229, Mar. 1989.

[14] T. Kasami *et al.*, "A concatenated coded modulation scheme for error control," *IEEE Trans. Commun.*, vol. 38, no. 6, pp. 752–763, June 1990.

[15] S. Lin and D. J. Costello, Jr., *Error Control Coding: Fundamentals and Applications*. Englewood Cliffs, NJ: Prentice-Hall, 1983.

[16] T. Kasami *et al.*, "On linear structure and phase rotation invariant properties of block 2^l—PSK modulation codes," *IEEE Trans. Inform. Theory*, vol. 37, no. 1 , pp. 164–167, Jan. 1991.

[17] D. Divsalar and M. K. Simon, "The design of trellis coded MPSK for fading channels : Performance criteria," *IEEE Trans. Commun.*, vol. 36, no. 9, pp. 1004–1012, Sept. 1988.



Sandeep Rajpal (S'88–M'95) received the B.Tech. degree in electrical engineering from the Indian Institute of Technology, Kanpur, India, and the Ph.D. degree in electrical engineering from the University of Hawaii, Manoa.

He is with the Multimedia Communications Division, Rockwell International Corporation, Newport Beach, CA. His research interests include error-control coding, coded modulation, and source coding.

Do Jun Rhee (S'85–M'96) received the B.S.E.E. degree from Hanyang University, Seoul, Korea, in 1983, the M.S. and P.E. degrees in electrical engineering from George Washington University, Washington, D.C., in 1987 and 1989, respectively, and the Ph.D. degree in electrical engineering from the University of Hawaii, Manoa, in 1995.

He is with the Communication Division, LSI LOGIC Corporation, Milpitas, CA. His current research interests include: algebraic coding theory, coded modulation for the mobile satellite communication, VLSI implementation of channel decoders, and fully digital QPSK demodulator for satellite broadcast receiver.

Shu Lin, for photograph and biography, see this issue, p. 63.