PERFORMANCE STUDY AND IMPROVEMENT OF UNGERBOECK-TYPE TRELLIS CODES

by

Ying Li

B.S. National Tsing Hua University, Taiwan, R.O.C.

(1986)

SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE
DEGREE OF

MASTER OF SCIENCE
IN ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

December 1988

Signature of Author ______________________________
Department of Electrical Engineering and Computer Science
December 14, 1988

Certified by ______________ ______________
Robert G. Gallager
Thesis Supervisor

Accepted by ______________________________
Arthur C. Smith
Chairman, Departmental Commitee on Graduate Students

PERFORMANCE STUDY AND IMPROVEMENT OF UNGERBOECK-TYPE TRELLIS CODES

by

Ying Li

SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE
DEGREE OF
MASTER OF SCIENCE
IN ELECTRICAL ENGINEERING AND COMPUTER SCIENCE
DECEMBER 1988

Abstract

Trellis coded modulation schemes are designed for band-limited communication channels to reduce errors caused by noise. Applications include telephone channels, digital radio, and satellite channels. In this work, we first study "regular" trellis codes, for which the performance analysis is much simplified. It is shown that for m-dimensional rectangular constellations partitioned into more than 2^{2m} subsets, regular binary trellis codes do not exist. The general structure of regular labelings for rectangular constellations are discussed. Also, we search over one- and two-dimensional Ungerboeck-type codes with a performance measure taking into account the minimum distance and the first three error coefficients. Codes with improved performance are found.

Thesis Supervisor: Robert G. Gallager
Title: Professor of Electrical Engineering

ACKNOWLEDGEMENTS

I thank my parents, Professor Ke-Yang Li and Ching-Wen Shieh for their love.

I thank my advisor, Professor Robert Gallager for his teaching, guidance and tremendous patience.

I thank my supervisor, Dr. David Forney for his help and encouragement, and also for permission to use figures from his paper.

I thank my supervisor, Dr. Vedat Eyüboğlu, for his guidance and support during and after my work in Codex Corporation.

I thank Professor Pierre Humblet, for his help and support.

I thank my officemates Diane Ho, Jay Kuo, Whay Lee, Rajesh Pankaj, John Spinelli, and Emre Telatar. They are wonderful.

This thesis has been supported in part by the National Science Foundation (NSF 8802991-NCR). I thank my sponsor.

Last but not least, I thank my special friend, Chin-Hsun Yeh, for everything.

Contents

List of Figures

Chapter 1

Introduction and Outline

On a communication channel, noise limits the performance by producing errors at the receiver, and thus makes the transmission unreliable. Shannon defined *channel capacity*, C, as the maximum rate at which data can be transmitted reliably on a noisy channel. Channel coding, or error control coding, is a technique used to combat noise such that data can be transmitted at higher rates reliably. The goal is to approach channel capacity with modest complexity. Coding schemes for power-limited channels have been designed successfully since the Sixties. However, for band-limited channels such as telephone channels, coding used to be considered impractical. The reasons were: first, the major channel impairments used to be dispersion and phase jitter, and second, it cost too much to do the signal processing for coding. The situation changed in the Seventies. Development of adaptive equalization techniques(adaptive filtering) eliminated most dispersion, and newer lines reduced phase jitter. Thus, additive noise became a major cause of errors. Modern VLSI technology also reduced the system cost for coding. Coding then became a practical and promising way to improve error performance on band-limited channels.

For band-limited channels with Added White Gaussian Noise(AWGN), channel capacity C is given by $W \log(1 + S/N)$, where S/N is the signal to noise ratio(SNR), and W is the allowable signal bandwidth[3]. On band-limited channels with high S/N, sending data reliably at rate C with simple uncoded pulse amplitude modulation requires 9 dB more power than the power S in the above formula. This 9 dB gap between the theoretical limit and PAM can be partly closed by coded modulation, i.e. combining channel coding and bandwidth-efficient modulation techniques. Ungerboeck proposed some coded modulation schemes in the late Seventies[1][2]. His results created great interest in both research and practical applications. This work also originated from a study of Ungerboeck codes. However, the study of regular labelings for rectangular constellations in section 4.1 turns out to be more general.

In chapter two, some principles of digital transmission over band-limited channels are reviewed, including bandwidth-efficient modulation methods, coded and uncoded. The basics of Ungerboeck coding are reviewed in chapter three with an example; previous improvements on Ungerboeck codes are also described. In chapter four, the performance of trellis codes is studied. In particular, we study the requirements for trellis codes to be "regular"[6][9]. "Regularity" largely simplifies the code design. The error coefficient effect and a method to find error coefficients are discussed. In chapter five, some new Ungerboeck-type codes are presented and compared with previous results.

Chapter 2

Background

When transmitting digital data over analog channels, the following events take place: the source generates digital data bits; the modulator maps each consecutive set of n bits to a signal waveform; the signal waveform is transmitted through the noisy channel, and at the receiving end the demodulator converts the signal back to the most likely digital data bits; the destination receives these bits. A communication system is shown below.

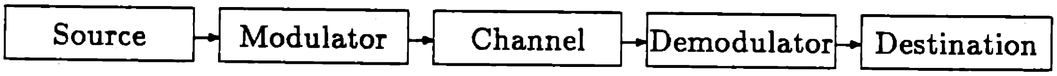
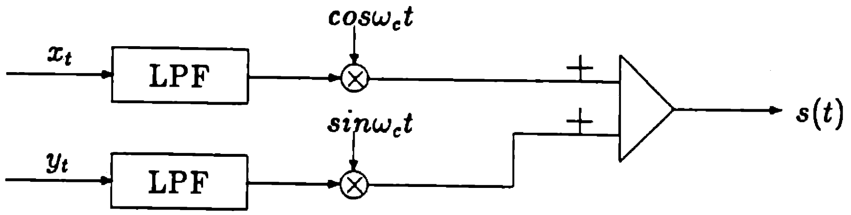


Figure 2.1: Model of a Communication System.

2.1 Bandwidth-efficient modulation methods

When channels are band-limited, bandwidth instead of signal power is often the expensive resource. An example is the telephone channel, which is band-limited between 300-3000 Hz, with high signal-to-noise ratios at 28 dB or more. On these channels, efficient modulation schemes that trade signal power for bandwidth are implemented. Modulation schemes for band-limited channels, coded or uncoded, are summerized in a tutorial paper by Forney, et al[3].

Bandwidth efficient modulation schemes can be implemented using a quadrature amplitude modulator(QAM). In QAM, the quadrature components of the channel signal waveform (sine and cosine waves at carrier frequency) are amplitude-modulated, as shown in Fig. 2.2. Techniques such as amplitude modulation, phase modulation and phase/amplitude modulation can be viewed as special cases of QAM.



x_t, y_t: pulse sequences
LPF: low pass filter
ω_c: carrier frequency
$s(t)$: line signal

Figure 2.2: QAM modulator.

Assuming the only channel impairment is Gaussian noise and the receiver achieves perfect timing, the channel can be modeled as a discrete time channel as shown in Fig. 2.3. Discrete time signals are specified by pairs (x_t, y_t), where each pair can be thought of as a symbol or a "signal point" lying on a two-dimensional space. The two coordinates x_t and y_t are sent independently and perturbed by Gaussian noise variables (n_{xt}, n_{yt}).

The "signal constellation" for QAM schemes is the collection of all possible signal points. An important and popular class is that of rectangular constellations. These constellations are composed of signal points drawn from the rectangular lattice. Others such as hexagonal constellations are discussed in [3]. It is shown that when

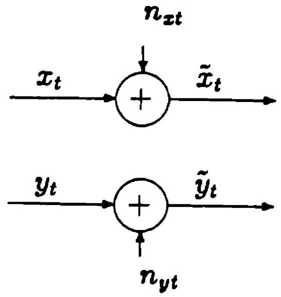


Figure 2.3: QAM channel model.

coding is used, gain that comes from choice of signal constellation is relatively small compared to coding gain. In this work our attention is restricted to rectangular constellations.

Digital signaling through QAM can be done by using one-dimensional pulse amplitude modulation(PAM) independently for each signal coordinate. In PAM, to send n bits, the signal point coordinate takes on one of 2^n equi-spaced levels. Therefore, $2n$ data bits are mapped to one of 2^{2n} points in the two-dimensional QAM constellation. The resultant constellation is a square. Alternatively, one can select signal points from the two-dimensional plane keeping in mind that a constellation with circular boundary is more desirable due to a smaller average power, and a smaller peak to average power ratio. The "cross" constellation, for example, as shown in Fig. 2.4 for 32-QAM, has a more circular boundary, and is better than the square. Some PAM and QAM rectangular signal constellations are shown in Fig. 2.4.

By using a large signal constellation, transmission rate is increased without bandwidth expansion. For example, doubling the size of the signal constellation while signaling rate is fixed means an additional 1 bit/symbol is sent. The price

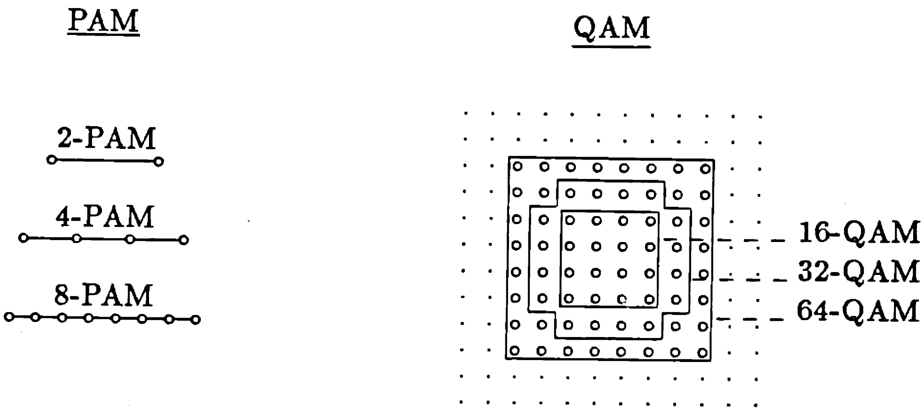


Figure 2.4: PAM/QAM signal constellations

for this increased throughput is a larger signal power. Therefore, we trade signal power(larger signal constellation) for bandwidth, or signaling rate. For rectangular constellations, approximately 4 times as much power(or 6 dB) is required to send an additional 1 bit/dimension[3]. As a result, doubling the size of signal constellations requires an additional average power of 6 dB for PAM and 3 dB for QAM.

The operation of the receiver is to decide from the received signal which of the possible signals was actually used. The best strategy is the Maximum Aposteriori Probability Receiver(MAP), and it answers the question: "Given the received signal, what is the most likely signal to have been sent?" However, provided that all the signals are equally likely to have been used, we can answer this question instead: "Which of the possible transmitted signals makes the signal that was received the most likely?" The Maximum Likelihood Receiver(ML) answers this question and has the advantage of making the receiver independent of the signal probabilities. Assuming the minimum Euclidean distance between any two signals is l_0, the probability of a wrong decision is upper bounded and approximated by the probability that the Gaussian noise vector(n_{xt}, n_{yt}) lies outside a circle of radius $l_0/2$, which is $P_e = exp(-l_0^2/2\delta^2)$, where δ^2 is the noise variance per degree of freedom[3]. If the spacing between signals is increased, P_e decreases. Thus, the larger the "distance" between signal points is, the less likely that a decision error will occur. Therefore, it is desirable to have the minimum distance between signals or sequences of signals as large as possible, while not violating the power constraint. One way to achieve this is by coding. In the following whenever "distance" is mentioned, *squared Euclidean distance* is implied. Notice that this "distance" is named for convenience of discussing these coding problems; it's properties are different from those of the distances we are familiar with.

2.2 Coded Modulation

For systems with uncoded modulation, to send n bits/symbol, a 2^n-point constellation is used. We can think of a sequence of two-dimensional symbols as a point in a higher dimension, lying on the lattice defined as the Cartesian product of two-dimensional rectangular lattices. The minimum distance between points in that higher dimension is the same as that in two dimensions. Therefore, decisions made based on sequences of received signals are no better than decisions made for each received signal independently.

However, channel coding techniques can be used to add redundancy to the signaling, and introduce interdependencies between sequences of signal points such that not all sequences are possible. One can then choose a code that generates only a set of "good sequences" where the minimum distance d_{min} between any two sequences is large. Therefore, the maximum likelihood receiver can make decisions by selecting the coded signal sequence that makes the received sequence most likely.

The combination of efficient modulation and coding gives rise to "coded modulation". The general structure of a coded modulation scheme is given in Fig. 2.5. To send n bits/symbol, a redundant 2^{n+r}-point signal constellation is used, partitioned into subsets. The basic process as pointed out by Forney, Gallager,et al. is the

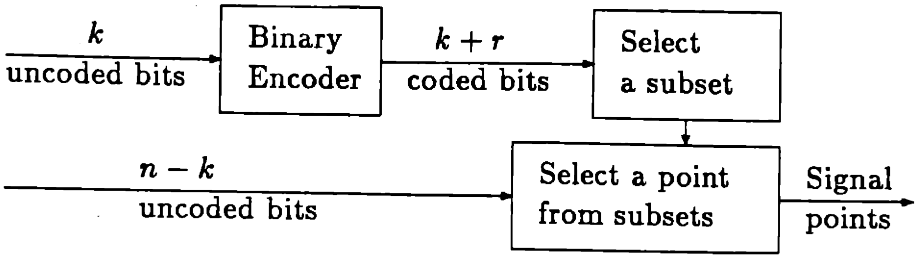


Figure 2.5: General coded modulation scheme.

following:

1. A rate $k/(k+r)$ binary encoder encodes k bits of the incoming data into $k+r$ coded bits.

2. The $k+r$ coded bits select one of the 2^{k+r} subsets of the partitioned signal constellation.

3. The remaining $n-k$ uncoded data bits select one signal point from the selected subset.

When the binary encoder used is a convolutional encoder[4], the scheme is a "trellis coded modulation" scheme. The set of all possible sequences of signal points generated by such a scheme is a "trellis code". Ungerboeck codes are a class of trellis codes. These codes can be described by the code trellis(Fig. 2.6) in much the same way as conventional convolutional codes. However, transitions in this trellis represent subsets, and each transition actually implies 2^{n-k} parallel transitions for all signals in the same subset. A "codeword" is a sequence of coded signals that compose a "path" in the code trellis. In conventional convolutional coding schemes, the signal constellations are the same as for the uncoded schemes; the coded bits with redundancy from coding are used to transmit signals for more times, which means that a higher data rate or a larger bandwidth is required to

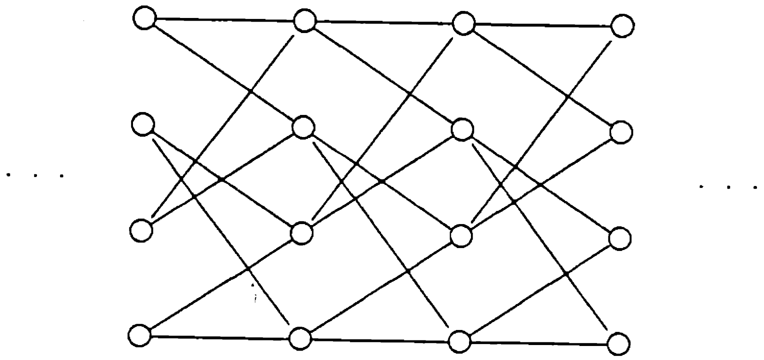


Figure 2.6: Code trellis diagram(4 state, rate 1/2 code)

achieve the same throughput as the uncoded schemes. In coded modulation schemes where bandwidth is considered expensive, coded bits are used to select signals from expanded signal constellations, which contain more signal points than those for the uncoded schemes; the data rate and bandwidth are kept the same.

Let the received signals be $r_n{=}a_n{+}w_n$, where the a_n are the discrete signals sent by the modulator, and the w_n represent samples of an additive white Gaussian noise process. The decision rule is to choose, among the set C of all possible coded signal sequences, the sequence $\{\hat{a}_n\}$ which satisfies

$$\sum_n |r_n - \hat{a}_n|^2 = \min_{\{a_n\}\in C} \sum_n |r_n - a_n|^2$$

The "soft decision" ML decoder determines the sequence $\{\hat{a}_n\}$ closest to the unquantized received sequence $\{r_n\}$ in terms of distance, i.e., squared Euclidean distance. The distance between two sequences $[a_n, a_{n+1}, \ldots]$ and $[r_n, r_{n+1}, \ldots]$ is the sum of the distances between symbols $[a_n,\ r_n]$, $[a_{n+1}, r_{n+1}]$, ... The Viterbi algorithm[5] can be used in the decoder to find the "nearest" sequence to the received sequence.

The error event probability P_e characterizes the performance of a code. It is the probability that at any given time the decoder either makes a wrong decision among the signals within the same subset, or starts to make a sequence of wrong decisions along some path diverging for more than one transition from the correct path. This will be discussed in more detail in section 4.2.

The following are important parameters for error event probability:

d_{min} : minimum squared Euclidean distance between codewords. The most probable errors made by the optimum soft-decision decoder occur between signals or sequences of signals $\{a_n\}$ and $\{b_n\}$, one transmitted and the other decoded, that are closest together. The minimum distance of a code is:

$$d_{min} = \min_{\{a_n\}\neq\{b_n\}} \sum_n |a_n - b_n|^2; \quad \{a_n\}, \{b_n\} \in C.$$

For a "distance invariant" code[9] where each codeword has the same distance properties as any other one, the all zero codeword can be chosen as the reference; therefore, d_{min} is equal to the minimum distance between the all zero sequence and any coded signal sequence; in other words, d_{min} is equal to the minimum "norm" of all codewords of a trellis code. Ungerboeck codes are distance invariant.

Error Coefficient N_0 : number of coded signal sequences that start with a nonzero signal and have the minimum norm d_{min}. When the all zero sequence is transmitted, and assuming the receiver is in the correct state, an error event occurs when the receiver chooses a sequence that starts with a nonzero signal. A large N_0 implies a large number of possibilities of error.

At high signal to noise ratio , P_e can be approximated by

$$P_e \simeq N_0 Q[\frac{\sqrt{d_{min}}}{2\delta}]$$

where δ is the Gaussian noise standard deviation in each dimension, and

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty \exp(-y^2/2)\, dy.$$

To achieve the same error event probability for coded and uncoded modulation, coded schemes have a power saving known as "coding gain". Coding gain is another way to measure performance for a trellis code, and it is also a function of signal to noise ratio. At high SNR, the "asymptotic coding gain" γ can be evaluated in dB by:

$$\gamma = 10\log_{10}[(d_{min}/d)/E_c/E_u)],$$

where d_{min} and d are minimum distances of the coded and uncoded schemes, and E_c and E_u are average signal energies of the coded and uncoded schemes, respectively.

At moderate SNR, coding gain may be lost due to a large number of nearest neighbors N_0. Define N_1, N_2 as the number of codewords with Euclidean weight $d_{min} + 1$, and $d_{min} + 2$, respectively. If N_1 and N_2 are very large, they will also increase P_e subsequently, and thus reduce coding gain. The "error coefficient effect" is considered in this work. It is shown that by a slight modification of the Viterbi algorithm, N_1 and N_2 can be evaluated easily. Therefore, they can be taken into account in the search for good codes.

Chapter 3

Ungerboeck codes

3.1 Working Principles

Ungerboeck's trellis coded modulation schemes[1][2] were proposed in the late Seventies. Using one-dimensional PAM, two-dimensional QAM , and PSK[1] signal constellations, coding gains of 3 to 6 dB can be achieved for digital transmission over band-limited channels without compromising bandwidth efficiency. Later schemes such as higher dimensional codes and codes based on lattices and cosets were proposed in [3][6][7][8]. Given all the later results, Ungerboeck's codes still stand out as a performance benchmark in terms of coding gain versus complexity[9].

In Ungerboeck's schemes, to send n bits in each signaling interval, a one- or two-dimensional constellation of 2^{n+1} points is used. The constellation is partitioned into 2^{k+1} subsets with enlarged intra-subset minimum Euclidean distance. Out of the n bits that arrive in each signaling interval, k bits enter a rate $k/k+1$ convolutional encoder, and the resulting $k+1$ coded bits specify which subset is to be used. The remaining $n-k$ data bits specify which point from the selected subset is to be transmitted.

The mapping from encoder output to subsets is called a "labeling", and the coded $k+1$ tuple is a "label"[9]. Ungerboeck's labeling comes from"mapping by set partitioning". The signal constellation is partitioned into subsets by a sequence

[1] Ungerboeck codes for PSK signals, with similar working principles, are not discussed in this work.

of 2-way partitions. This is done in such a way that points in the same subset are placed as far apart as possible, and the minimum intra-subset distance grows as the number of partitions increases. The 2^{k+1} subsets are labeled, bit by bit, by results of these 2-way partitionings . Therefore, starting from the last bit, which indicates the result of the first 2-way partitioning, the more bits on which two labels agree, the larger the minimum distance between these two subsets is. Ungerboeck used one-dimensional 4-way partitioned PAM and two-dimensional 8-way partitioned QAM signal constellations for his codes, where the minimum distance between any two subsets can be determined by the number of bits two labels agree on. However, this method to find minimum distance between subsets does not work in one dimension for more than 4-way partitioned constellations and in two dimensions for more than 8-way partitioned constellations. Ungerboeck labelings for one-dimensional 4-way partitioned PAM and two-dimensional 8-way partitioned QAM constellations are shown in Fig. 3.1.

When the signal constellation is finite, problems of "outer points" that lie close to the boundaries arise. Comparing with the inner points, outer points have fewer "near neighbors" and thus have a smaller chance of being in error. When designing trellis codes, this means that points in the same subset are "different", and all pairs of codewords need to be considered in order to find the "real" d_{min}, N_0,...etc. A huge amount of work is thus required in the code design. In the following we shall assume the signal constellations to be *infinitely large*. This assumption is reasonable when the constellation is large. It separates the choice of constellation size from the code design, and largely simplifies the code design. The error coefficients assuming the constellation is infinite will be larger than those for finite constellations.

In Ungerboeck's schemes, if labels of two subsets agree in the last q positions but not in the $q+1$th bit, then the minimum distance between signal points from these two subsets is independent of the particular subsets and will be denoted Δ_q. For one-

1D 4-Way Partitioned Constellation

⋯	○	○	○	○	○	○	○	○	○	⋯
					↓ 2-way partition					
⋯	0	1	0	1	0	1	0	1	0	⋯
					↓ 2-way partition					
⋯	00	01	10	11	00	01	10	11	00	⋯

2D 8-Way Partitioned Constellation

⋯	110	011	010	111	110	011	010	111	⋯
⋯	101	000	001	100	101	000	001	100	⋯
⋯	010	111	110	011	010	111	110	011	⋯
⋯	001	100	101	000	001	100	101	000	⋯
⋯	110	011	010	111	110	011	010	111	⋯
⋯	101	000	001	100	101	000	001	100	⋯
⋯	010	111	110	011	010	111	110	011	⋯

Figure 3.1: Ungerboeck Labelings

dimensional PAM and two-dimensional QAM [2], Δ_q is 2^{2q} and 2^q, respectively; and Δ_{k+1} is set to zero. For subsets corresponding to a and a', the minimum squared distance is a function only of the number of trailing zero's of $a \oplus a'$. Therefore, Ungerboeck codes have the "distance invariant" property. That is, the distribution of distances from any given code sequence to all other code sequences is the same as the weight distribution of the code. Therefore, minimum distance and error coefficients can be found using the all zero sequence as the reference, and the code design is largely simplified.

3.2 Example: Ungerboeck 4-state 1D code

This code uses a 2^{n+1}-point PAM constellation, divided into 4 subsets of 2^{n-1} points each. A rate 1/2 convolution code is used to select the subsets. The scheme is shown in Fig. 3.2. We shall find the minimum distance d_{min} and error coefficient N_0 for this code.

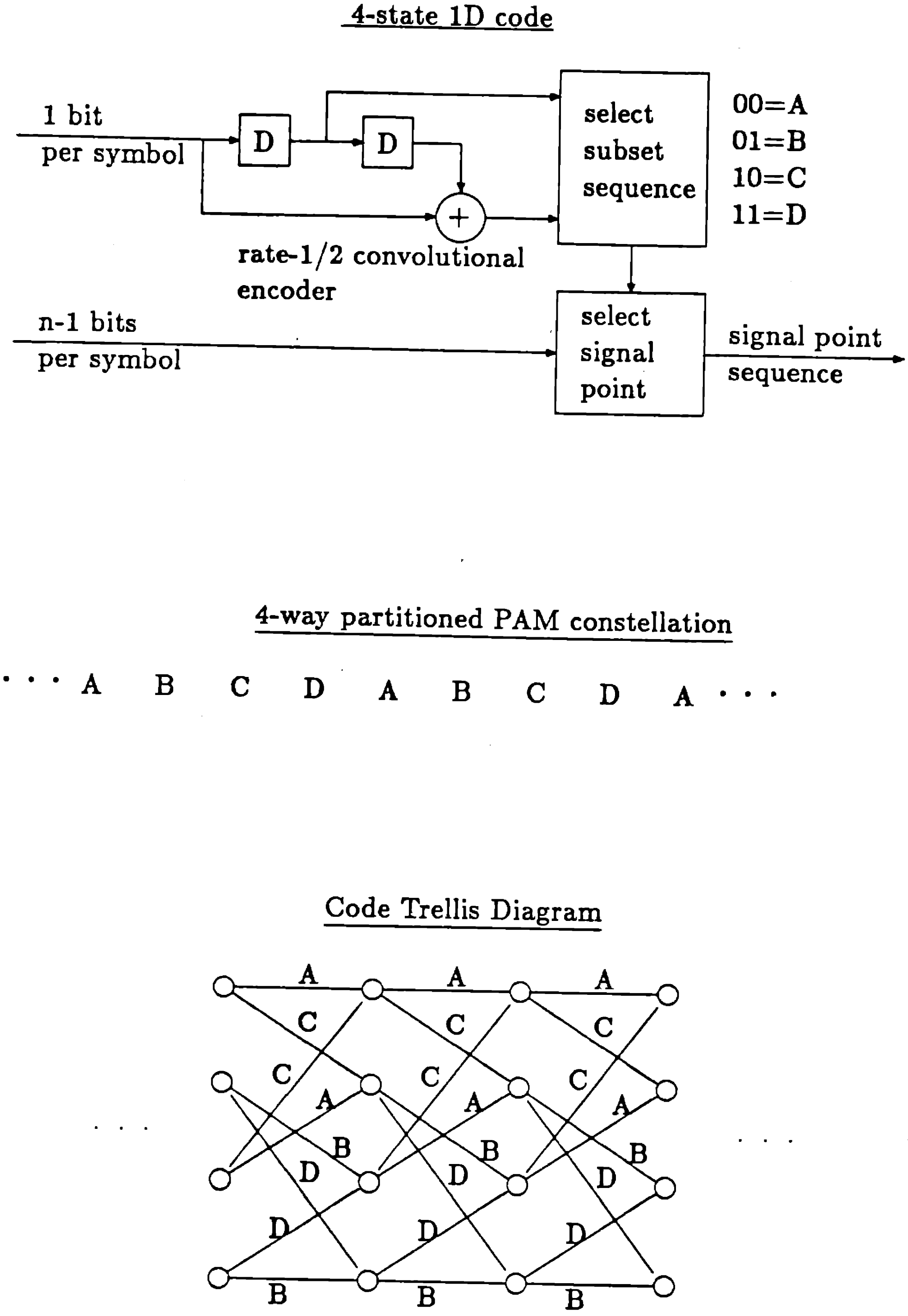
The minimum distance can be expressed as

$$d_{min} = \min[d_1, d_2]$$

where d_1 is the minimum distance between points in the same subset, which corresponds to parallel transitions in the code trellis; d_2 denotes the minimum distance between nonparallel paths in the code trellis diagram. N_0 is the number of paths at distance d_{min} away from a given path on the code trellis, assuming an infinite constellation. Ungerboeck codes are regular; therefore, the all zero path can be taken as the reference.

The minimum squared Euclidean distance between different points in the same subset, for example, between different points labeled A, is $d_1 = 16$ as seen in Fig. 3.1. Define the minimum squared distance between subset i and subset A to be $d(i)$, then $d(A) = 0$, $d(B) = 1$, $d(C) = 4$, $d(D) = 1$. Also define $n(i)$ as the number of points

[2] The distance between neighboring points is set to one.



4-state 1D code

4-way partitioned PAM constellation

Code Trellis Diagram

Figure 3.2: One-dimensional 4-state Ungerboeck code

in subset i at the minimum distance away from the zero subset; therefore, $n(A) = 2$, $n(B) = 1$, $n(C) = 2$, $n(D) = 1$. The minimum distance path away from the all zero path is the path {C,B,C}. The distance is $d(C)+d(B)+d(C) = 9 = d_2$, and number of those paths is $n(C) * n(B) * n(C) = 4$. Therefore, $d_{min} = \min[d_1, d_2] = d_2 = 9$, and $N_0 = 4$.

3.3 Improved Ungerboeck-type codes

While some simple Ungerboeck codes were hand-designed, most were found by a nearly exhaustive computer search. The performance measure in the search was the asymptotic coding gain. Thus, codes with the largest possible d_{min} were considered best. Error coefficients were not taken into account in the search, although their significance was recognized. [3] It is therefore possible to find better codes with the same d_{min} and smaller error coefficients.

Recently, Honig[10], and Pottie and Taylor[11] proposed improved Ungerboeck-type one- and two-dimensional codes for PAM and QAM signals, respectively. Taking into account error coefficients, they both found codes with the same d_{min} but smaller N_0 than Ungerboeck codes. Their approaches are the following:

- Honig[10] improved Ungerboeck-type codes for one-dimensional 4-PAM and 8-PAM constellations. A feedback-free(feedforward) encoder was used. The performance measure in the search was an upper bound for P_e. Codes with maximum possible d_{min} were chosen first. For each of these codes the upper bound of P_e was computed. Codes that minimize the upper bound were considered best.

- Pottie and Taylor[11] improved Ungerboeck-type codes for two-dimensional QAM constellations, assuming the signal constellations are very large. A

[3] The error coefficients N_0 for Ungerboeck codes were later computed and appeared in [2].

feedback-free encoder was used. Codes with good d_{min} were found, and among them the ones with minimum N_0 were selected as best. After the codes were found, the first five error coefficients[4] were computed and used in an approximate upper bound of P_e; the coding gain was evaluated accordingly. They recognized that codes with slightly smaller d_{min} might have a significantly smaller N_0, and thus perform better. This idea was applied when searching for complicated codes with large N_0. They found a 128-state code in this way which performs better at moderate signal to noise ratio where P_e is at the range of 10^{-5} to 10^{-6}.

These improvements demonstrated the importance of error coefficients in code design. When searching for good codes, it will be desirable to take more error coefficients into account in a reasonable way. The upper bound for P_e that Honig used requires knowledge of all error coefficients, and is not very good as a performance measure. This will be discussed in section 4.2. Although Pottie and Taylor cosidered several error coefficients at the performance evaluation after the code search, they only cosidered N_0 in the code search. In this work, the first three error coefficients are considered in the code search. Instead of bounding or approximating P_e, d_{min}, N_0, N_1, and N_2 are used to compute an "effective coding gain" as defined by Forney[9], which is a simple and yet realistic performance measure.

While Ungerboeck searched over codes implemented by a systematic feedback encoder, Honig, and Pottie and Taylor looked at feedforward codes. There should be no difference using one kind over the other. However, the heuristic rejection rules used to save search time are not the same for feedback and feedforward codes and might lead to different results. Searching over systematic feedback codes has several benefits: firstly, catastrophic codes are ruled out; secondly, Ungerboeck's heuristic to guarantee large d_{min} can be used, which cuts down search time by a

[4] The way they computed error coefficients is correct for finding the first four error coefficients, but not the fifth.

factor of 4 for 1D and 8 for 2D codes; thirdly, there is no need to worry about how to divide memory elements into two queues, which must be done for feedforward codes. As a result, the search in this work is done over systematic feedback codes.

Chapter 4

Performance of Trellis Codes

4.1 Regular Labelings for Rectangular Constellations

4.1.1 Introduction

In coded modulation schemes, at each time interval some of the input bits (k bits) enter the encoder, and the coded $k+r$ bits are used to select subsets from a partitioned signal constellation. The 2^{k+r} binary $k+r$ tuples are called "labels", and the mapping from labels to subsets is called a "labeling" according to Forney in "Coset Codes I" [9].

Forney defined a labeling to be "regular" if the minimum squared Euclidean distance between points in two subsets is a function of the mod-2 sum of their labels only, independent of the individual labels. To understand the definition of "coset codes" let's look at a few terms. An m-dimensional "lattice" is a discrete set of m-dimensional vectors(points) that forms a group under ordinary vector addition. A "coset" is a translation of a lattice. A lattice can be partitioned into subsets that are cosets of some lattice. A coset code is one where the signal constellation is a finite set of points taken from an infinite lattice, and the partitioning of the constellation into subsets corresponds to the partitioning of that lattice into a sublattice and its cosets. Practically all known good constructive coding techniques for band-limited channels[9] are coset codes, including Ungerboeck-type codes.

In this work, we do not restrict ourselves initially to cosets of a given lattice; rather, we study methods to build sensible labeling schemes for partitioned rectangular signal constellations. Since the labels are binary $k+r$ tuples, the number of subsets must be a power of two. We restrict ourselves to partitions that are structured in such a way that for each point in a given subset a, the minimum distance to points in subset b is the same for all points in a, denoted as $d(a,b)$. $d(a,a)=0$ $\forall a$. Also, each point in subset a has the same number of points in b at distance $d(a,b)$ away, and we define $n(a,b)$ to be this number. These conditions are true when the subsets are cosets of a lattice.

Define $D(l) = d(0,l)$ to be the norm of the subset labeled l, or the minimum distance between the subset labeled 0 and subset l; define $N(l) = n(0,l)$ to be the multiplicity of subset l, or the number of points in subset l at distance $D(l)$ away from a given point in subset zero. According to Forney, a labeling is regular if

$$d(a,b) = D(a \oplus b), \quad \forall a,b \in L,$$

where L is the set of all possible labels, or equivalently, the set of 2^{k+r} binary $k+r$ tuples. The minimum distance between subsets a, b, instead of being determined by the pair of two labels a and b, is determined by the label-difference $a \oplus b$, which is itself a label. Since the labels are binary strings where sum is equal to difference, the mod-2 sum of labels will also be referred to as label-difference later on. We use Forney's definition of regular labeling, but, as mentioned above, we do not restrict ourselves to cosets of a lattice.

A trellis code is regular, as defined by Calderbank and Sloane[6], if the squared Euclidean distance between two coded signal sequences is a function of the mod-2 sum of the input sequences only, independent of the individual sequences. For a regular trellis code, the distribution of distances from any given code sequence to all other code sequences is the same as the norm distribution of code sequences. Therefore, regular trellis codes are distance-invariant[9]. When a code is distance invariant, d_{min} and N_0, N_1 and N_2 can be found using the all zero sequence as the reference; it is unnecessary to look at all pairs of codewords . In section 4.3 we will show that regardless of the size of the signal constellation, only a small portion of the signal constellation (a basic set) needs to be considered to find d_{min}, N_0, N_1 and N_2. This, together with the distance-invariant property, largely simplifies the design of a regular coset code, making it essentially no harder than designing a binary code using antipodal or QPSK signals. Therefore, as with conventional convolutional codes, regular coset codes are of special interest to code designers. A regular coset code must be based on a partition with a regular labeling[9]. Thus, it is important to understand structures of regular labelings.

In the following, we discuss structures for regular labelings when signal constellations are rectangular. Firstly, it is shown that for m-dimensional rectangular constellations, no regular binary labeling exists for more than 2^{2m}-way partitions. Actually, we shall see that if the label-differences of an m-dimensional point and it's $2m$ nearest neighbors are given and are linearly independent, this uniquely determines the labels for all points in the space. Ungerboeck's labelings[1] and those used by Calderbank and Sloane[6] follow this structure, and are "equivalent" labelings in the sense that there is a one to one linear relation between the two sets of labels. Secondly, it is shown that for less than 16-way partitions in two dimensions, the structure for regular labelings is not unique. In addition, it is shown that even though $d(a,b) = D(a \oplus b)$ $\forall a, b$, $n(a,b) = N(a \oplus b)$ is not necessarily true. Equivalently, a regular labeling is not sufficient to guarantee that the number of points in one subset at minimum distance away from a point in another subset is also a function only of the label difference of the two subsets. This is because there are many coded signal sequences corresponding to one label sequence. If $n(a,b) \neq N(a \oplus b)$ it will be difficult to find error coefficients. Thus "strong regularity" for labelings will be defined to be:

$$d(a,b) = D(a \oplus b) \quad n(a,b) = N(a \oplus b) \quad \forall a, b \in L$$

4.1.2 Regular Labelings for m-dimensional Partitions with 2^{2m} Subsets

Consider an m-dimensional rectangular constellation where the distance between neighboring points is one. Let L_1 be the set of label differences(binary strings) between a point labeled zero and it's $2m$ nearest neighbors at distance one. The same label differences are counted only once. Under these assumptions, the following results come from the definition of regular labeling.

Proposition 1 *For a regular labeling, each point has the same set of label differences between itself and it's $2m$ nearest neighbors as each other point.*

Proof: Let L_1^e be the set of label differences between an arbitrary point labeled e and it's nearest neighbors. L_1, as defined, is the set of label differences between zero and it's nearest neighbors. If there exists $c \in L_1^e$ but c not in L_1, then $d(e, e \oplus c) = 1 \neq D(c) = d(0, 0 \oplus c) > 1$. This contradicts the fact that the labeling is regular, since from the definition of regularity, $d(e, e \oplus c) = D(c)\ \forall e \in L$, where L is the set of all possible labels. As a result, all elements in L_1^e must be in L_1, or $L_1^e \subset L_1$. Conversely, if any element in L_1, say a_1, is not in L_1^e, then $d(e, e \oplus a_1) > 1$, while $d(0, 0 \oplus a_1) = D(a_1) = 1$ and thus $d(e, e \oplus a_1) \neq D(a_1)$, which violates regularity. Thus any element in L_1 must be in L_1^e, or $L_1 \subset L_1^e$. We therefore conclude that $L_1 = L_1^e$. Since e is arbitrary, our proof is complete.

Q.E.D.

Proposition 2 *All labels for a regular labeling scheme can be expressed as linear combinations of elements in L_1, where L_1 is the set of label differences between any point and it's nearest neighbors.*

Proof: According to Proposition 1, the set of label differences between any point and all it's distance one neighbors must be L_1. Therefore, the label difference between any two points at distance one apart must be in L_1. If we "label" the edge between any two points at distance one apart by the label difference of the points, labels of all edges have to be in L_1. If labels for all edges are determined, the label for any point is just the mod-2 sum of labels for all edges on the path from point zero to it. Labels for all points must therefore be expressed as linear combinations of elements in L_1.

Q.E.D.

Proposition 3 *For an m-dimensional rectangular constellation, binary regular labeling does not exist for more than 2^{2m}-way partitions.*

Proof: From Proposition 2, all labels can be expressed as linear combinations of elements in L_1. When the elements of L_1: a_i, b_i $i = 1 \dots m$ are linearly independent, the largest number of different labels can be generated. Since labels are binary strings, $2m$ linearly independent binary strings have 2^{2m} different linear combinations. Therefore, regular labelings do not exist for partitions with more than 2^{2m} subsets.

Q.E.D.

In this section we try to "build" regular labelings for m-dimensional rectangular constellations with 2^{2m}-way partition, $m = 1, 2, \dots$. From Proposition 3, the elements of L_1 for these labelings must be linearly independent. It turns out that there is a unique and simple structure. Once a_i, b_i are determined and are linearly independent, labels for all points are fixed. Although we do not restrict ourselves to partitions where subsets are cosets, and we intend to find regular labelings instead of strongly regular labelings, the labeling turns out to be strongly regular, and the subsets correspond to cosets of a magnified rectangular lattice.

Let's start from the one-dimensional PAM constellation (Fig. 4.1). Let $L_1 = \{a, b\}$, a, b are linearly independent using mod-2 operations, which means that

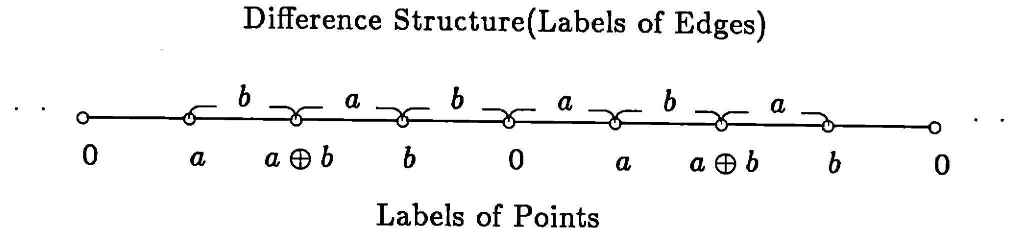


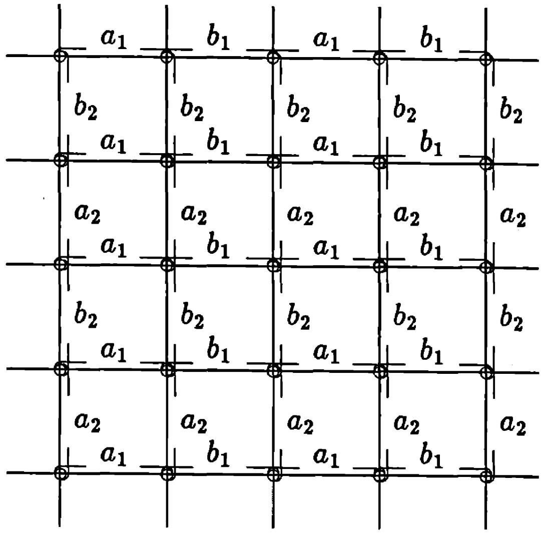
Figure 4.1: Structure of Regular Labelings for 1D 4-way Partition

$a \neq b$. We can arbitrarily select a point and label it zero, and then label it's two nearest neighbors a and b. From the three labeled points, we have the conditions $d(a,0) = D(a) = 1$, and $d(b,0) = D(b) = 1$. If we label each edge in Fig. 4.1 by the label difference of it's two end points, to satisfy $D(a) = D(b) = 1$, the sequence of label differences (the "difference structure") must be an alternating sequence of $\{a, b, a, b, \ldots\}$. The sequence of labels is therefore $\{0, a, a \oplus b, b, 0, a, a \oplus b, \ldots\}$, as seen in Fig. 4.1.

Checking the minimum inter-subset distances and multiplicities centered at any point, the above construction indeed leads to a regular labeling where $D(a) = D(b) = 1$, $N(a) = N(b) = 1$, $D(a \oplus b) = 4$, and $N(a \oplus b) = 2$. This can be explained as follows: looking out from any point, the sequences of label differences must be either $a, b, a, b..$ to the left and $b, a, b, a...$ to the right, or $a, b, a, b...$ to the right and $b, a, b, a...$ to the left. If from a point of subset zero one can go to exactly one point of subset c by moving x segments to the right, with a one to one correspondence one can go from any point of subset e to a point $e \oplus c$ by moving either x or $-x$ segments, with the same distance. Thus, $d(e, e \oplus c) = D(c) = x$, $n(e, e \oplus c) = N(c)$ for all e in L, and the labeling is not only regular, but strongly regular. In addition, points in the same subset form a magnified and shifted one-dimensional rectangular lattice with minimum distance 16.

Next, look at the two-dimensional QAM constellation. Let $L_1 = \{a_1, b_1, a_2, b_2\}$, a_1, b_1, a_2, b_2 are linearly independent. We can thus label point $(0,0)$ to be 0, and it's nearest four neighbors $(\pm 1, 0), (0, \pm 1)$ to be a_1, b_1, a_2, b_2 (see Fig. 4.2). $D(a_1) = D(b_1) = D(a_2) = D(b_2) = 1$. Consider the corner point $(1,1)$ adjacent to a_1, a_2(see Fig. 4.2). For a_1, this neighbor could be labeled by either $a_1 \oplus a_2, a_1 \oplus b_1$, or $a_1 \oplus b_2$, according to Proposition 1. Similarly, for a_2, a valid label for this point is either $a_2 \oplus a_1, a_2 \oplus b_1$, or $a_2 \oplus b_2$. Since a_1, b_1, a_2, b_2 are linearly independent, two linear combinations of a_i, b_i cannot be the same unless they contain identical components; therefore, $a_1 \oplus a_2 \neq a_2 \oplus b_1$, $a_1 \oplus a_2 \neq a_2 \oplus b_2$, $\ldots$, etc. The only possible label for this point is thus $a_1 \oplus a_2$. Labels for the other three corners can be found in the same way. Once this square is filled, proceed to label the four next nearest points $(\pm 2, 0), (0, \pm 2)$. These are the only unlabeled nearest neighbors of points a_1, b_1, a_2, b_2, and thus their labels are $a_1 \oplus b_1, a_2 \oplus b_2$ from Proposition 1. Up to now, we have learned that once the point at position (x, y) and it's nearest neighbors $(x+1, y)$, $(x, y+1)$, $(x-1, y)$, and $(x, y-1)$ are labeled, labels for the nearest neighbors of $(x+1, y)$, $(x, y+1)$, $(x-1, y)$, and $(x, y-1)$ are also fixed; by repeating the above procedure to new center points with four labeled nearest neighbors, labels for all points in the two-dimensional plane can be determined.

We can check the regularity by looking at the "difference structure" as in one dimension. Labels for the edges are ordered as seen in Fig. 4.2. Notice that moving horizontally from any point in the plane, one sees an alternating sequence of differences given by $a_1, b_1, a_1, b_1, \ldots$, and similarly moving vertically the sequence is $a_2, b_2, a_2, b_2, \ldots$. If one can go from a point of subset zero to the nearest point of subset c by moving a distance x horizontally and y vertically, then one can go from any point of subset e to the nearest point of subset $e \oplus c$ by moving either $(x, y), (-x, y), (-x, -y)$, or $(x, -y)$, all with the same distance $x^2 + y^2$. Therefore, $d(e, e \oplus c) = D(c) = x^2 + y^2$. Notice the one to one relation between paths from 0 to c and from e to $e \oplus c$. This says that $n(e, e \oplus c) = N(c)$. Thus the labeling is strongly regular. Also, each subset corresponds to a magnified and shifted rectangular lattice with distance 16 between neighboring points.


a_1 b_1 a_1 b_1
b_2 b_2 b_2 b_2 b_2
a_1 b_1 a_1 b_1
a_2 a_2 a_2 a_2 a_2
a_1 b_1 a_1 b_1
b_2 b_2 b_2 b_2 b_2
a_1 b_1 a_1 b_1
a_2 a_2 a_2 a_2 a_2
a_1 b_1 a_1 b_1

Difference Structure(Labels of Edges)

	⋮	⋮	⋮	⋮	⋮	
...	$a_1b_1a_2b_2$	$b_1a_2b_2$	a_2b_2	$a_1a_2b_2$	$a_1b_1a_2b_2$	...
...	$a_1b_1a_2$	b_1a_2	a_2	a_1a_2	$a_1b_1a_2$	...
...	a_1b_1	b_1	0	a_1	a_1b_1	...
...	$a_1b_1b_2$	b_1b_2	b_2	a_1b_2	$a_1b_1b_2$	...
...	$a_1b_1a_2b_2$	$b_1a_2b_2$	a_2b_2	$a_1a_2b_2$	$a_1b_1a_2b_2$	...
	⋮	⋮	⋮	⋮	⋮	

Labels of Points
note: $\oplus$ is omitted to save space

Figure 4.2: Structure of Regular Labelings for 2D 16-way Partition

The structure in Fig. 4.2 is, as seen by construction, the only way to regularly label a 16-way partitioned two-dimensional constellation when a_1, b_1, a_2, b_2 are linearly independent; when a_1, b_1, a_2, b_2 are dependent(i.e., when there are less than 16 subsets), this structure still works. Labeling schemes for 4-way, 8-way and 16-way partitions used by Ungerboeck[1] and Calderbank and Sloane[4] both follow this structure, although they associate different binary strings with a_1, b_1, a_2, b_2.

When generalized to m dimensions, similar procedures can still be used to find the structures of regular labeling. First label a point at (0,0,..0) to be zero, and label it's $2m$ nearest neighbors at (1,0,0,..0), (-1,0,0,..0), (0,1,0,..0), (0,-1,0,..0), ...(0,0,...1), (0,0,...-1) by $a_1, b_1, a_2, b_2, ...a_m, b_m$. Then $L_1 = \{a_i, b_i,\ i = 1 \ldots m\}$. All elements in L_1 are linearly independent. Let $\vec{e}_i$ be the unit vector of the ith coordinate, then we have labeled $\vec{e}_i$ to be a_i and $-\vec{e}_i$ to be b_i for $i = 1 \ldots m$.

Consider point (1,1,0,...0) adjacent to (1,0,0,...0) and (0,1,0,...0) that are labeled as a_1, a_2. From Proposition 1, when (1,1,0,...) is viewed as a neighbor of a_1, possible labels are $a_1 \oplus a_i$ or $a_1 \oplus b_j$, where $i \neq 1$, $j = 1 \ldots m$; and when viewed as a neighbor of a_2, possible labels are $a_2 \oplus a_i$ or $a_2 \oplus b_j$, where $i \neq 2$, $j = 1 \ldots m$. The valid label must be in the intersection of the two sets. Since all a_i and b_i are linearly independent, the only valid label is $a_1 \oplus a_2$. Similarly for point (1,0,1,0,...0), or, $\vec{e}_1 + \vec{e}_3$, which is a neighbor of both a_1 and a_3, the label can only be $a_1 \oplus a_3$. For point (1,1,1,0,...0), the label must be $a_1 \oplus a_2 \oplus a_3$ since it is adjacent to $a_1 \oplus a_2$, $a_2 \oplus a_3$ and $a_3 \oplus a_1$. In general, point $\vec{e}_i + \vec{e}_j - \vec{e}_k + \ldots$ is labeled by $a_i \oplus a_j \oplus b_k \oplus \ldots$. In this way, we can label all points whose coordinates are between ±1. Next proceed to label the points with one coordinate 2 and all the rest zero, or, $\pm 2\vec{e}_i$, $i = 1 \ldots m$. The point (2,0,...0), for example, is the only unlabeled nearest neighbor for point a_1. Since $a_1 \oplus b_1$ is the only possible label not yet used, this point is labeled as $a_1 \oplus b_1$. Similarly, (-2,0,0...0) is the only unlabeled nearest neighbor of b_1, it also must be labeled as $a_1 \oplus b_1$. In general, points $\pm 2\vec{e}_i$ are labeled by $a_i \oplus b_i$, $i = 1, \ldots m$. Now that all the nearest neighbors of points a_i, b_i are labeled, labels for their distance two neighbors can also be found, as we have done this for point zero. This can then be "propagated" to any point in the m-dimensional space, and labels for all points are fixed as a result.

The difference structure is still highly ordered. Looking out from any point, the sequences of differences along axis i must be either $\{a_i, b_i, a_i, b_i, \ldots\}$ in the positive direction and $\{b_i, a_i, b_i, a_i \ldots\}$ in the negative direction, or the other way round. This structure assures regularity, since if one can go from a point of subset zero to the nearest point of subset c by moving in each dimension $(x_1, x_2, \ldots x_m)$, one can surely go from any point of subset e to the nearest point in subset $e \oplus c$ by moving $(\pm x_1, \pm x_2, \ldots. \pm x_m)$, with the same distance $x_1^2 + x_2^2 + \ldots x_m^2$ in any way. The multiplicity is also preserved since there is a one to one relation between the shortest paths starting from the zero point and those starting from point e. Therefore $d(e, e \oplus c) = D(c) = x_1^2 + x_2^2 + \ldots x_m^2$, $n(e, e \oplus c) = N(c)$, and the labeling is indeed regular and strongly regular. Points in the same subset form a magnified and shifted rectangular lattice, where the neighboring points are at distance 16 apart.

4.1.3 The Ungerboeck Labelings

Ungerboeck's labelings[1] for one-dimensional PAM and two-dimensional QAM signals are generated by successive 2-way partitions of the constellation, and by use of one bit to represent the result of each partition. Forney[9] further explains Ungerboeck's labeling by a partition tower and a partition tree corresponding to a chain of coset decompositions, where each bit selects one of the two cosets in each level. Alternatively, Ungerboeck's labelings can be analyzed using the structures we found. The structures of Ungerboeck's labelings agree with the ones shown in Fig. 4.1 and Fig. 4.2. For the one-dimensional 4-way partition, $a = 01$, $b = 11$; for the two-dimensional 4-way partition, $a_1 = b_1 = 01$, $a_2 = b_2 = 11$; for the two-dimensional 8-way partition, $a_1 = 101$, $a_2 = 011$, $b_1 = 001$, $b_2 = 111$. Although never used, Ungerboeck proposed the labeling for the two-dimensional 16-way partition as well, which has $a_1 = 0101$, $a_2 = 1011$, $b_1 = 0001$, $b_2 = 0111$.(Fig. 4.3, Fig. 4.4)

The fact that for Ungerboeck's labeling, the "minimum distance between subsets is a function of only the number of trailing zeros in the label-difference" contributes to heuristic rules that reduce code search time. This "trailing zero method" breaks down, though, for the Ungerboeck labeling of the 2D 16-way partition[1]. Nevertheless, that labeling is still regular. Obviously we can find other regular labelings by associating a_i, b_i, $i = 1, ..m$ with different binary strings. For 2D 4-way, 8-way and 16-way partitions, both Ungerboeck's and Calderbank and Sloane's labeling schemes[4] used the same structure as in Fig. 4.2[1]. Labels in one scheme and the other can be related by a linear one to one mapping(Fig. 4.3, Fig. 4.4), and there is a one to one relation between codes with the same performance using one labeling and the other. Therefore, Ungerboeck's labeling and Calderbank/Sloane's labeling are essentially equivalent.

A careful choice of labeling could simplify the code design, as in the case of Ungerboeck. However, when trying to design a very complicated code, the time saved by choice of labeling is limited.

The complexity of a code is roughly proportional to $2^{\nu} * k$, where 2^{ν} is the number of encoder states, and k is the number of transitions entering each state in the code trellis. The largest gain achievable by coding is bounded by the minimum intra-subset distance. This gain can be achieved by increasing the complexity of coding. However, it is very hard to design a complicated code. It is also hard to design a code that is not regular, in which case all pairs of codewords must be considered to find d_{min} and error coefficients. For 8-way partition in two dimensions, the largest gain is achieved by a code so complicated that one would not consider going beyond the 16-way partition. A similar situation occurs in one dimension, where the

[1] Calderbank and Sloane's labeling for 2D 16-way partition is a regular binary labeling, and actually the product of two one-dimensional regular labelings. However, they used it as a quadrature(mod-4) linear labeling. Their labeling for two-dimensional 4-way partition corresponds to that for standard binary codes.

Relation between Ungerboeck and Calderbank/Sloane's labelings:

for any point point labeled as Y by Ungerboeck,
Z by Calderbank/Sloane,

$$YM = Z$$

where M is a nonsingular binary k by k matrix for 2^k-way partition.

2D 16-way partition

Note: labels are in decimal to save space

Ungerboeck(Y)

Calderbank/Sloane(Z)

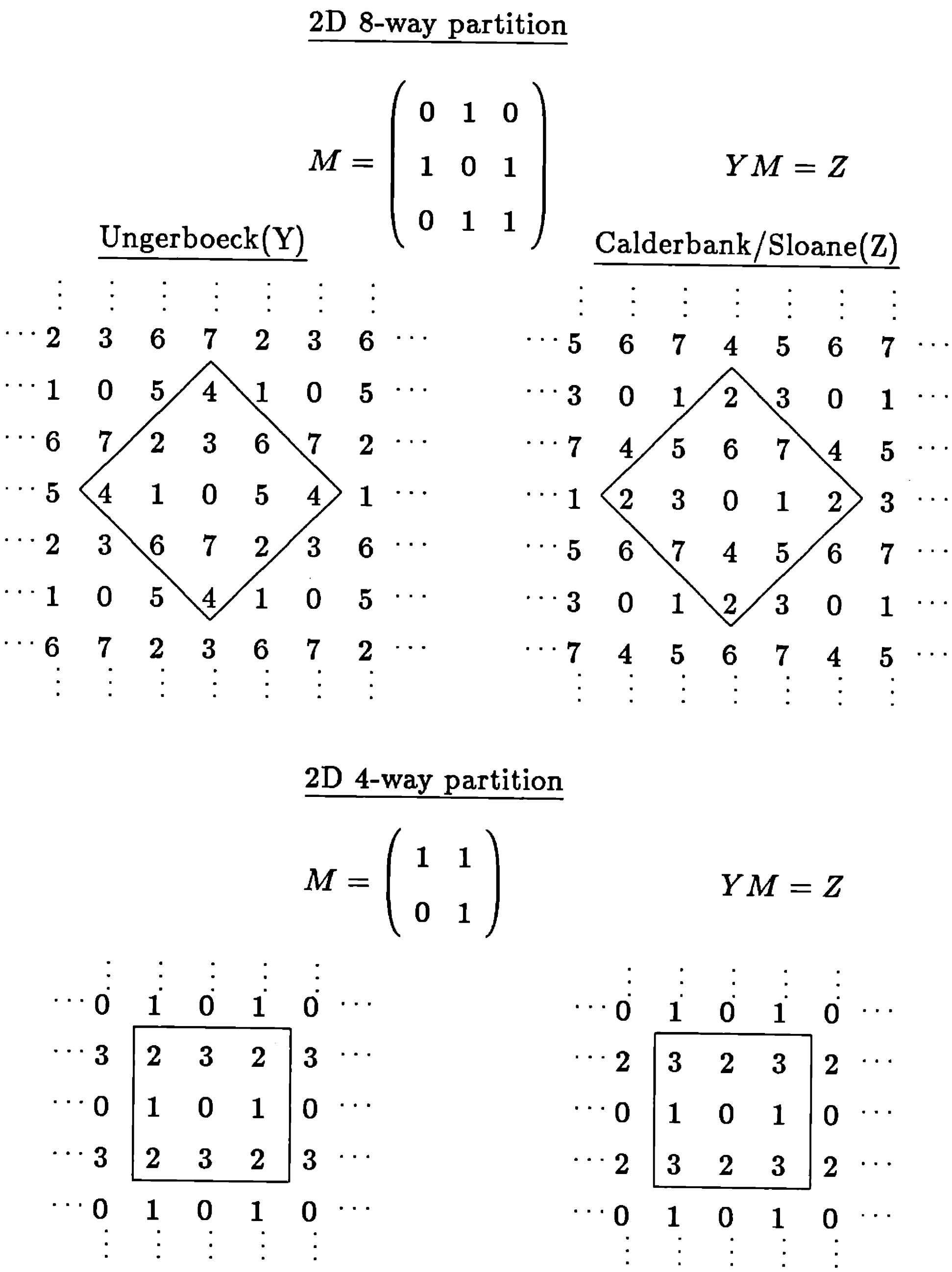
$$M = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$

$$YM = Z$$

	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
⋯	2	3	6	7	2	3	6	⋯
⋯	9	8	13	12	9	8	13	⋯
⋯	14	15	10	11	14	15	10	⋯
⋯	5	4	1	0	5	4	1	⋯
⋯	2	3	6	7	2	3	6	⋯
⋯	9	8	13	12	9	8	13	⋯
⋯	14	15	10	11	14	15	10	⋯
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	

	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
⋯	5	6	7	4	5	6	7	⋯
⋯	9	10	11	8	9	10	11	⋯
⋯	13	14	15	12	13	14	15	⋯
⋯	1	2	3	0	1	2	3	⋯
⋯	5	6	7	4	5	6	7	⋯
⋯	9	10	11	8	9	10	11	⋯
⋯	13	14	15	12	13	14	15	⋯
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	

Figure 4.3: Equivalent Ungerboeck and Calderbank/Sloane Labelings:1



2D 8-way partition

$$M = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix} \qquad YM = Z$$

Ungerboeck(Y)

$$\begin{array}{ccccccccc}
 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \\
\cdots & 2 & 3 & 6 & 7 & 2 & 3 & 6 & \cdots \\
\cdots & 1 & 0 & 5 & 4 & 1 & 0 & 5 & \cdots \\
\cdots & 6 & 7 & 2 & 3 & 6 & 7 & 2 & \cdots \\
\cdots & 5 & 4 & 1 & 0 & 5 & 4 & 1 & \cdots \\
\cdots & 2 & 3 & 6 & 7 & 2 & 3 & 6 & \cdots \\
\cdots & 1 & 0 & 5 & 4 & 1 & 0 & 5 & \cdots \\
\cdots & 6 & 7 & 2 & 3 & 6 & 7 & 2 & \cdots \\
 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots &
\end{array}$$

Calderbank/Sloane(Z)

$$\begin{array}{ccccccccc}
 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \\
\cdots & 5 & 6 & 7 & 4 & 5 & 6 & 7 & \cdots \\
\cdots & 3 & 0 & 1 & 2 & 3 & 0 & 1 & \cdots \\
\cdots & 7 & 4 & 5 & 6 & 7 & 4 & 5 & \cdots \\
\cdots & 1 & 2 & 3 & 0 & 1 & 2 & 3 & \cdots \\
\cdots & 5 & 6 & 7 & 4 & 5 & 6 & 7 & \cdots \\
\cdots & 3 & 0 & 1 & 2 & 3 & 0 & 1 & \cdots \\
\cdots & 7 & 4 & 5 & 6 & 7 & 4 & 5 & \cdots \\
 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots &
\end{array}$$

2D 4-way partition

$$M = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \qquad YM = Z$$

$$\begin{array}{ccccccc}
 & \vdots & \vdots & \vdots & \vdots & \vdots & \\
\cdots & 0 & 1 & 0 & 1 & 0 & \cdots \\
\cdots & 3 & 2 & 3 & 2 & 3 & \cdots \\
\cdots & 0 & 1 & 0 & 1 & 0 & \cdots \\
\cdots & 3 & 2 & 3 & 2 & 3 & \cdots \\
\cdots & 0 & 1 & 0 & 1 & 0 & \cdots \\
 & \vdots & \vdots & \vdots & \vdots & \vdots &
\end{array}
\qquad
\begin{array}{ccccccc}
 & \vdots & \vdots & \vdots & \vdots & \vdots & \\
\cdots & 0 & 1 & 0 & 1 & 0 & \cdots \\
\cdots & 2 & 3 & 2 & 3 & 2 & \cdots \\
\cdots & 0 & 1 & 0 & 1 & 0 & \cdots \\
\cdots & 2 & 3 & 2 & 3 & 2 & \cdots \\
\cdots & 0 & 1 & 0 & 1 & 0 & \cdots \\
 & \vdots & \vdots & \vdots & \vdots & \vdots &
\end{array}$$

Figure 4.4: Equivalent Ungerboeck and Calderbank/Sloane Labelings:2

code achieving the maximum gain for 4-way partition is already very complicated. As we have seen, regular labelings exist for up to one-dimensional 4-way and two-dimensional 16-way partitions only. Codes with further partitions cannot be regular. It is thus extraordinarily difficult to design m-dimensional codes for more than 2^{2m}-way partitions. The time spent for code search would be enormous since good codes are complicated, and no regular labeling exists.

4.1.4 Regular Labelings for m-dimensional Partitions with Fewer than 2^{2m} Subsets

In the previous section, we concluded that there is a unique structure for regular labeling for m-dimensional 2^{2m}-way partition, and that no regular labeling exists if the constellation is partitioned further. Actually, constellations for all existing coset codes are partitioned into 2^{2m} and fewer subsets in m dimensions. It is thus useful to also understand regular labelings for less than 2^{2m}-way partitions.

For m-dimensional partitions with fewer than 2^{2m} subsets, $L_1 = \{a_i, b_i,\ i = 1 \ldots m\}$ are not linearly independent anymore. However, Proposition 1 still holds, which says that the set of label differences between any point and it's nearest neighbors must be the same. Let $d_{intra}(a)$ be the minimum distance between points in subset a, then the minimum intra-subset distance $d_{intra} = \min_a d_{intra}(a)$. Since d_{intra} limits the largest d_{min} achievable by coding, structures where d_{intra} is maximized are primarily considered. In the following, we use the definition of regularity, Proposition 1, and the requirement that d_{intra} is to be maximized to build regular labelings. Structures for regular labelings of the one-dimensional 2-way partition, and two-dimensional 2-way, 4-way and 8-way partitions are found.

For a one-dimensional 2-way partition, let the two labels be $0, A$. In order to have d_{intra} larger than one, the same labels cannot be placed next to each other. Therefore, the only regular labeling is shown in Fig. 4.5. Since $n(e, e \oplus a) = N(a) = 2$, for all e in L, the labeling is strongly regular. This structure is the same as that in Fig. 4.1 where $a = b$.

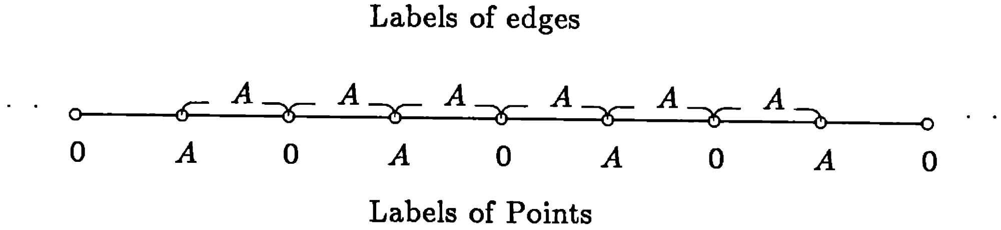


Figure 4.5: Structure of Regular Labeling for 1D 2-way Partition

The structure of regular labeling for two-dimensional 2-way partition is very similar to that in one dimension, as shown in Fig. 4.6. Again, the condition that the same labels cannot be placed next to each other is enough to determine the whole structure. This structure is also strongly regular.

For the two-dimensional 4-way partition, let's call the four different labels $0, A, B, C$. These labels correspond to binary strings "00, 01, 10, 11", and thus $A \oplus B \oplus C = 0$. While 0 corresponds to "00", we shall not specify the correspondence between A, B, C and "01, 10, 11". Later we will see that all choices of A, B, C work equally well.

Let's start from a point labeled as 0 located at $(0,0)$. Our first attempt is to label it's four distance one neighbors (±1,0), (0, ±1). To make d_{intra} larger than one, these points cannot be labeled zero. Therefore, the four points have to be labeled by A, B, C, and at least one label has to be used twice. The same labels should be placed as far apart as possible to achieve large d_{intra}. Therefore, (1,0), (-1,0) can share the same labels, as can (0,1), (0,-1). Since the distance between the two uses of the same label is 4, the largest d_{intra} for a 4-way partition is 4. We shall restrict ourselves to structures where $d_{intra} = 4$. Fig. 4.7 shows two ways to label the near neighbors of point 0. Other labeling choices can be converted to them by interchanging A, B, C(which are actually arbitrary) and perhaps by rotating by 90 degrees ("flipping" the structure with respect to the x or y axis).

```
  :   :   :   :   :   :   :   :   :   :
..D   A   D   A   D   A   D   A   D   A..

..A   D   A   D   A   D   A   D   A   D..

..D   A   D   A   D   A   D   A   D   A..

..A   D   A   D   A   D   A   D   A   D..

..D   A   D   A   D   A   D   A   D   A..

..A   D   A   D   A   D   A   D   A   D..

..D   A   D   A   D   A   D   A   D   A..

..A   D   A   D   A   D   A   D   A   D..

..D   A   D   A   D   A   D   A   D   A..

..A   D   A   D   A   D   A   D   A   D..
  :   :   :   :   :   :   :   :   :   :
```

Figure 4.6: Structure of Regular Labeling for 2D 2-way Partition

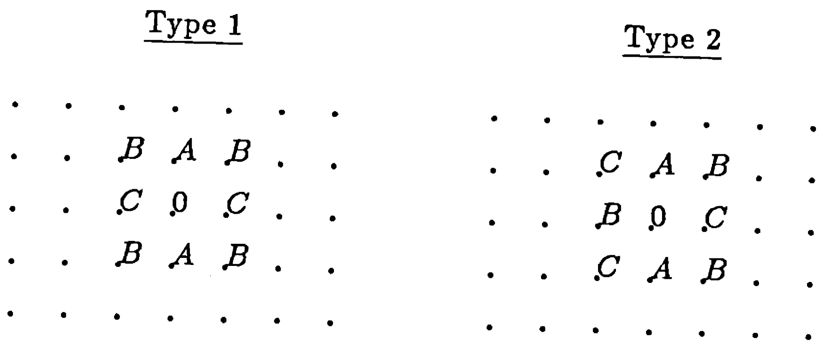


Figure 4.7: Partial Structures for 2D 4-way partition

In structure 1(Fig. 4.7), A and C are both used twice for the nearest neighbors; the four points at (±1,±1) must be labeled B, since otherwise d_{intra} will be less than 4. Actually, any four corner points of a square with unit sides must be labeled $A, B, C,$0 for d_{intra} to be greater than 2. In structure 2, $(0, \pm 1)$ are both labeled A, but (1,0), (-1,0) are labeled C, B. Labels for points at (±1,±1) are determined since each point is the only unlabeled corner for a side-one square. These partial structures are completed in the following.

For any point, say (x, y), in the type one structure, only two labels are used to label it's four neighboring points at distance 1: $(x \pm 1, y \pm 1)$. To make $d_{intra} = 4$, the label for $(x + 1, y)$ must be the same as that for $(x - 1, y)$. Also, labels for $(x, y + 1)$ and $(x, y - 1)$ must be the same. This says that the same label repeats every two steps along both x and y axis. As a result, given the partial structure in Fig. 4.7, the complete structure is found as in Fig. 4.8.

For the type 2 structure, we start from the partial structure in Fig. 4.7. Labels can be found from Proposition 1 and from the constraint that $d_{intra} = 4$. For example, point (1,0) has been labeled C, and the three neighbors of C at (0,0), (1,1), (1,-1) are labeled as 0, B, B, respectively. Since $L_1 = \{A, B, C\}$, from Proposition 1 the neighbors of (1,0) should be labeled by $C \oplus A = B$, $C \oplus B = A$, and $C \oplus C = 0$. Therefore, the point (2,0) must be labeled as A. Point (-2,0) is labeled A for the same reason. The points (0,±2) are both labeled 0, since any

Structure 1

```
   :  :  :  :  :  :  :  :  :  :
..0  C  0  C  0  C  0  C  0  C..
..A  B  A  B  A  B  A  B  A  B..
..0  C  0  C  0  C  0  C  0  C..
..A  B  A  B  A  B  A  B  A  B..
..0  C  0  C  0  C  0  C  0  C..
..A  B  A  B  A  B  A  B  A  B..
..0  C  0  C  0  C  0  C  0  C..
..A  B  A  B  A  B  A  B  A  B..
..0  C  0  C  0  C  0  C  0  C..
..A  B  A  B  A  B  A  B  A  B..
   :  :  :  :  :  :  :  :  :  :
```

Structure 2

```
   :  :  :  :  :  :  :  :  :  :
..0  C  A  B  0  C  A  B  0  C..
..A  B  0  C  A  B  0  C  A  B..
..0  C  A  B  0  C  A  B  0  C..
..A  B  0  C  A  B  0  C  A  B..
..0  C  A  B  0  C  A  B  0  C..
..A  B  0  C  A  B  0  C  A  B..
..0  C  A  B  0  C  A  B  0  C..
..A  B  0  C  A  B  0  C  A  B..
..0  C  A  B  0  C  A  B  0  C..
..A  B  0  C  A  B  0  C  A  B..
   :  :  :  :  :  :  :  :  :  :
```

Figure 4.8: Structures of Regular Labelings for 2D 4-way partition

other label will make $d_{intra} \leq 2$. Continuing labeling, we observe after a while that the same label repeats every two steps along the y axis and every four steps along the x axis. The complete type 2 structure is shown in Fig. 4.8.

Both structures in Fig. 4.8 can be shown to be strongly regular, regardless of the choice of A, B, C as $10, 01, 11$. Structure 1, where subsets correspond to cosets of a magnified rectangular lattice, is the one used by Ungerboeck and Calderbank/Sloane. It is a special case of Fig. 4.2 where $a_1 = b_1$, $a_2 = b_2$. For structure 2, the 0 points correspond to a lattice which is not rectangular; other subsets correspond to cosets of that lattice. Given a structure, the six choices of A, B, C as $01, 10, 11$ result in essentially the same labelings. Define n_{intra} as follows: for a point in a subset where the intra-subset distance is d_{intra}, n_{intra} is the number of points in the same subset at distance d_{intra} away from it. Thus structure 1 has $n_{intra} = 4$, structure 2 has $n_{intra} = 2$, and both structures have $d_{intra} = 4$. When coding complexity is so high that $d_{min} = d_{intra}$, $N_0 = n_{intra}$, and the signal to noise ratio is large enough, the code labeled by structure 2 will have a larger coding gain due to the smaller error coefficient. The trade-off is that for structure 2, all subsets have norms of no more than one, while structure 1 has one subset with norm two. This can lead to a smaller d_{min} given that the complexity of the codes using either structure are the same and are moderate.

The two-dimensional 8-way partition is of special interest since it is used in most two-dimensional codes, including those proposed by Ungerbock, Calderbank and Sloane. As discussed earlier, their labeling schemes are actually special cases of the structure in Fig. 4.2 where $a_1 \oplus a_2 \oplus b_1 \oplus b_2 = 0$. In the following, other structures of regular labelings are discussed, including structures that are not strongly regular, and structures in which subsets are not cosets of one lattice.

Label the four nearest neighbors of point zero to be a_1, a_2, b_1, b_2 as before. From Proposition 2 the eight labels for this 8-way partition are linear combinations of a_1, a_2, b_1, b_2. Therefore, out of a_1, a_2, b_1, b_2, three must be linearly independent, and the other is a linear combination of the first three. Without loss of generality, let a_1, a_2, b_1 be linearly independent. To find b_2, let's check d_{intra} for different choices of b_2 and select the one that maximizes d_{intra}. If b_2 is a_1, a_2, or b_1, d_{intra} is 2, 4, 2, respectively. If b_2 is $a_1 \oplus a_2$, from Proposition 1 it's four neighbors must be 0, a_2, a_1, $a_1 \oplus a_2 \oplus b_1$. This says that at least one point at distance 2 from point zero must be labeled a_1 or a_2, which make $d_{intra}(a_1)$, $d_{intra}(a_2)$, and thus d_{intra} no more than 5. Similarly, d_{intra} is no more than 5 when b_2 is $a_2 \oplus b_1$ or $a_1 \oplus b_1$. At last look at $b_2 = a_1 \oplus a_2 \oplus b_1$. The equivalent relation $a_1 \oplus a_2 \oplus b_1 \oplus b_2 = 0$ implies that the shortest path between two points with the same labels contains four edges, labeled a_1, a_2, b_1, b_2, respectively. Each edge has length one; therefore, d_{intra} is at least eight when the two point are different by ($\pm$2,$\pm$2). Thus, b_2 is chosen to be $a_1 \oplus a_2 \oplus b_1$, and $L_1 = \{a_1, a_2, b_1, b_2\}$. We will show later that the largest d_{intra} is actually 8.

Among the eight labels for the 8-way partition, apart from 0, a_1, a_2, b_1, b_2, there are three other labels, let's call them A, B, C. The set $\{A, B, C\}$ corresponds to $\{a_1 \oplus a_2, a_1 \oplus b_1, a_2 \oplus b_1\}$, but we shall not specify the correspondence for now. Notice that for any point labeled a_1, a_2, b_1 or b_2, from Proposition 1 it's four nearest neighbors must be $0, A, B, C$. Similarly, for any point labeled 0, A, B, or C, it's four nearest neighbors must be a_1, a_2, b_1, b_1. The eight labels are thus broken into two sets, $\{a_1, a_2, b_1, b_2\}$, $\{A, B, C, 0\}$. In these sets, $a_1 \oplus a_2 \oplus b_1 = b_2$, and $A \oplus B \oplus C = 0$.

Consider the four corner points at distance 2 away from point zero, which must be labeled by A, B, C. To label these points, at least one label must be used twice. Since the distance between these two uses of the same label is 8, the largest d_{intra} for an 8-way partition is 8. Similar to the 4-way partition, there are again two possible structures for labeling near neighbors of point zero, as shown in Fig. 4.9. The sets of norms for both structures contain $D(a_1) = D(a_2) = D(b_1) = D(b_2) = 1$, $D(a_1 \oplus a_2) = D(a_2 \oplus b_1) = 2$ and for type one structure, $D(A) = D(B) = 2$, $D(C) = 4$, while $D(A) = D(B) = D(C) = 2$ for type two structure. Given the norms, labels for all points in the space that satisfy the condition for regularity can

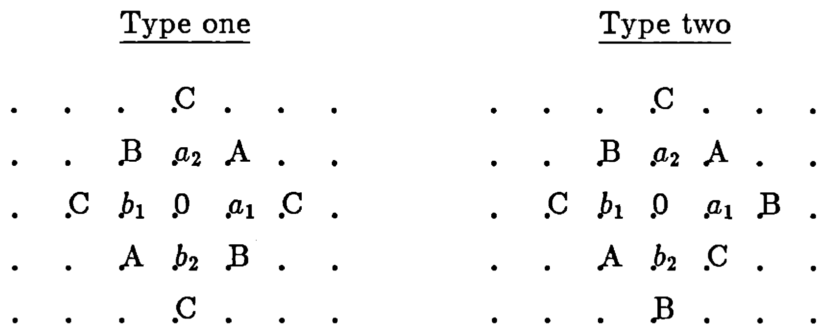


Figure 4.9: Partial Structures for 2D 8-way partition

be found. The complete structures are shown in Fig. 4.10 and Fig. 4.11.

The type one structure in Fig. 4.10 has the property that each subset looks like a magnified and shifted rectangular lattice, rotated by 45 degrees. Thus, each subset corresponds to one coset of a rotated rectangular lattice. $0, A, B, C$ actually follow structure 1 for the 4-way partition, magnified and rotated by 45 degrees, and so do a_1, a_2, b_1, b_2. For the structure to be completely determined, A, B, C have to be specified in terms of linear combinations of a_1, a_2, b_1. Since $D(C) = 4 = D(a_1 \oplus b_1)$ and there is only one subset with norm 4, C must be $a_1 \oplus b_1$. When $A = a_1 \oplus a_2$, and $B = a_2 \oplus b_1$, the labeling is the popular 8-way partition labeling used by Ungerboeck and Calderbank/Sloane. It is also a special case of the structure in Fig. 4.2, and is strongly regular. Alternatively, when $A = a_2 \oplus b_1$, and $B = a_1 \oplus a_2$, the labeling is also strongly regular. Since the sets of norms and multiplicities are the same in both cases, one cannot tell the difference in terms of d_{min} and error coefficients whether a code is using one labeling or the other. Therefore, these two choices of A, B result in equivalent labelings.

There are two ways to complete type two structures. The first one, on the top of Fig. 4.11, has the property that all subsets correspond to cosets of a lattice which is not rectangular. In this case, subsets $0, A, B, C$ and a_1, a_2, b_1, b_2 both follow structure 2 of the two-dimensional 4-way partition, magnified and rotated by 45 degrees. For this structure to be regular, all six choices of A, B, C in terms

$$
\begin{array}{cccccccccccc}
 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \\
\cdots & D & a_1 & C & b_1 & D & a_1 & C & b_1 & D & a_1 & \cdots \\
\cdots & b_2 & B & a_2 & A & b_2 & B & a_2 & A & b_2 & B & \cdots \\
\cdots & C & b_1 & D & a_1 & C & b_1 & D & a_1 & C & b_1 & \cdots \\
\cdots & a_2 & A & b_2 & B & a_2 & A & b_2 & B & a_2 & A & \cdots \\
\cdots & D & a_1 & C & b_1 & D & a_1 & C & b_1 & D & a_1 & \cdots \\
\cdots & b_2 & B & a_2 & A & b_2 & B & a_2 & A & b_2 & B & \cdots \\
\cdots & C & b_1 & D & a_1 & C & b_1 & D & a_1 & C & b_1 & \cdots \\
\cdots & a_2 & A & b_2 & B & a_2 & A & b_2 & B & a_2 & A & \cdots \\
\cdots & D & a_1 & C & b_1 & D & a_1 & C & b_1 & D & a_1 & \cdots \\
\cdots & b_2 & B & a_2 & A & b_2 & B & a_2 & A & b_2 & B & \cdots \\
 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots &
\end{array}
$$

Figure 4.10: Structure of Regular Labeling for 2D 8-way Partition: Type 1

of $a_1 \oplus a_2, a_2 \oplus b_1, a_1, b_1$ will do. However, for the labeling to be strongly regular, $N(a_1 \oplus a_2) = N(a_1 \oplus b_1) = 1 = N(B) = N(C)$, and $N(a_2 \oplus b_1) = 2 = N(A)$. Therefore, A must be $a_2 \oplus b_1$, while B, C can be either $a_1 \oplus a_2$, $a_1 \oplus b_1$ or the other way round. The other four choices of A, B, C result in structures that are regular but not strongly regular, while subsets correspond to cosets of a lattice.

Similarly, for the bottom structure in Fig. 4.11, $0, A, B, C$ follow structure 2 in Fig. 4.8, magnified and rotated. However, a_1, a_2, b_1, b_2 correspond to that structure flipped with respect to the y axis. Since structure 2 for the 4-way partition is not symmetrical with respect to the y axis, a_1, a_2, b_1, b_2 correspond to cosets of some lattice that is different from the lattice 0. In this case the multiplicity $N(A) = 2 = N(a_1 \oplus a_2)$. Therefore, there are two strongly regular labelings where $A = a_1 \oplus a_2$, and four other regular labelings that are regular but not strongly regular. We thus have found structures that are strongly regular, but in which subsets do not correspond to cosets of one lattice. The four strongly regular labelings for the top and bottom structures are equivalent in the sense that they all have four subsets with norm 1 and multiplicity 1, three subsets with norm 2, and among them two subsets have multiplicity 1 and the other one has multiplicity 2.

Similar to the 4-way partition, type two structures for 8-way partitions are special because of the reduced n_{intra}. While $d_{intra} = 8$ for both type one and type two structures, $n_{intra} = 4$ for type one and $n_{intra} = 2$ for type two structures. There is again a trade-off: while all subsets of the type two structures have norms of no more than two, the type one structure has one subset with norm four. All structures found so far have $d_{intra} = d_{intra}(a)$ and $n_{intra} = n_{intra}(a)$, $\forall a \in L$.

4.2 Error Event Probability

The performance of a trellis code is measured by its ability to reduce error rate due to noise. Therefore, probability of error is of key importance.

	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
..	0	a_1	B	a_2	A	b_2	C	b_1	0	a_1	..
..	b_2	C	b_1	0	a_1	B	a_2	A	b_2	C	..
..	B	a_2	A	b_2	C	b_1	0	a_1	B	a_2	..
..	b_1	0	a_1	B	a_2	A	b_2	C	b_1	0	..
..	A	b_2	C	b_1	0	a_1	B	a_2	A	b_2	..
..	a_1	B	a_2	A	b_2	C	b_1	0	a_1	B	..
..	C	b_1	0	a_1	B	a_2	A	b_2	C	b_1	..
..	a_2	A	b_2	C	b_1	0	a_1	B	a_2	A	..
..	0	a_1	B	a_2	A	b_2	C	b_1	0	a_1	..
..	b_2	C	b_1	0	a_1	B	a_2	A	b_2	C	..
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	

	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
..	0	a_1	B	b_2	A	a_2	C	b_1	0	a_1	..
..	b_2	C	a_2	0	b_1	B	a_1	A	b_2	C	..
..	B	b_1	A	a_1	C	b_2	0	a_2	B	b_1	..
..	a_1	0	b_2	B	a_2	A	b_1	C	a_1	0	..
..	A	a_2	C	b_1	0	a_1	B	b_2	A	a_2	..
..	b_1	B	a_1	A	b_2	C	a_2	0	b_1	B	..
..	C	b_2	0	a_2	B	b_1	A	a_1	C	b_2	..
..	a_2	A	b_1	C	a_1	0	b_2	B	a_2	A	..
..	0	a_1	B	b_2	A	a_2	C	b_1	0	a_1	..
..	b_2	C	a_2	0	b_1	B	a_1	A	b_2	C	..
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	

Figure 4.11: Structures of Regular Labelings for 2D 8-way Partition: Type 2

At each time interval, the Maximum Likelihood decoder conditionally selects a sequence of channel symbols, but only the first symbol of that sequence is sent out as the decision. Therefore, an "error event" occurs when a sequence with a wrong first symbol is selected. It is not considered an error if the selected sequence is not the correct sequence but has the correct first symbol. For conventional convolutional codes, the "error event probability" P_e is also called the "node error probability". It is the probability that, given the decoder is at the correct state(node) in time k, the decision it makes leads to an incorrect state at time $k+1$. For trellis codes with multiple transitions, the state sequence does not correspond one to one to the symbol sequence. An "error event" can either start with a node error or a "parallel transition " error where the decoder remains in the correct state at time $k+1$ but the first transition of the selected sequence is parallel to that of the correct sequence. Thus, P_e is the conditional probability that, given the decoder is in the correct state, the next symbol decoded will be incorrect. Equivalently, P_e is the probability that, given the correct state, the decoder will start to make a sequence of errors. Notice that this is not the same as the fraction of symbols received incorrectly. However, what we really care about is not the ratio of symbols received incorrectly, but how often do error events occur. This is because every error event, long or short, usually requires a retransmission. Therefore, P_e is important, and has been used as a performance measure when searching for trellis codes.

On the code trellis, a state transition that contains parallel transitions corresponds to a subset; each transition among the parallel transitions corresponds to a symbol or equivalently a signal point, and a path corresponds to a sequence of channel signals. For regular trellis codes, P_e can be analyzed without loss of generality assuming that the all zero path is the correct transmitted sequence. Define A as the set of infinitely long paths with the first transition equal to zero, including the all zero sequence, and also define A' as the set of infinitely long paths with an incorrect first transition, including the ones starting with a transition parallel to the correct transition. Then P_e is the probability that a sequence $a' \in A'$ is chosen by the decoder given that the all zero sequence is transmitted. The received sequence r is an independent Gaussian random vector with zero mean and standard deviation δ in each dimension. Define $\Delta M(a',a)$ to be $\sum_{k=j}^{\infty} |r^k - a'^k|^2 - \sum_{k=j}^{\infty} |r^k - a^k|^2$. $\Delta M(a',a) \leq 0$ means that the squared distance between r and a' is less than that between r and a; this means that r is more likely to be received if a' is the transmitted sequence than if a is transmitted.

$$\begin{aligned} P_e &= Pr[\bigcup_{a' \in A'} [\bigcap_{a \in A} \{\Delta M(a',a) \leq 0\}]] \\ &\leq Pr[\bigcup_{a' \in A'} \{\Delta M(a',0) \leq 0\}] \\ &\leq \sum_{a' \in A'} Pr[\Delta M(a',0) \leq 0] \\ &= \sum_{d_i \geq d_{min}} N_i Q(\frac{\sqrt{d_i}}{2\delta}) \end{aligned}$$

In general, A contains many more paths "close" to the correct path than A' does.

When upper bounding P_e as above, the probability of the event " a' is more likely than <u>all</u> a " is first upper bounded by that of the event " a' is more likely than the all zero sequence", then the union bound is applied for the probability of union of the later events. Therefore, the union bound for P_e is not tight.

At high SNR, P_e will be small and composed mostly of error probability from the nearest error events at distance d_{min}. P_e can be approximated by taking into account only the nearest error events and using the union bound on their probability:

$$P_e \simeq N_0 Q[\frac{\sqrt{d_{min}}}{2\delta}]$$

In practice, the communication system is often operated at a signal to noise ratio(SNR) where P_e is in the range of 10^{-5} to 10^{-6}. In this range, the error events with slightly larger distances come into the picture. If $N_1, N_2, \ldots$ are relatively large, they might contribute more to P_e than the nearest error events and dominate the performance. It is therefore useful to find a few terms of N_i's in the code search.

The complete sequence $[N_i]$ or the "weight distribution" for some trellis codes can be found from their generating functions[12][13]. They can be used to find the upper bound for P_e. However, in order to find $[N_i]$ for a 2^ν state code, a ν by ν matrix with symbolic entries is required to be inverted. In addition, the distance between any two signal points in the signal constellation must be known. This becomes quite difficult when ν is large or when the signal constellation is large. Fortunately, the leading terms N_0, N_1, N_2 can be found rather easily. This is because all the individual symbol errors comprising these "near" error events are between symbols that are close together. Thus, only a small portion of the signal constellation need to be considered, with a size determined by the number of subsets, independent of the actual size of the constellation. Methods to find N_0, N_1, N_2 will be discussed in section 4.4. Due to its simplicity, N_0, N_1, and N_2 can be computed when searching for good codes.

4.3 Coding Gain

Another way to characterize the performance of a trellis code is from its ability to save power over an uncoded system while achieving the same error probability. The "coding gain" is defined as the power saving when using a trellis code over an uncoded system. It is a function of P_e and SNR.

The "asymptotic coding gain" γ is the coding gain when P_e is very small, or, when SNR is large. γ can be evaluated in dB by

$$\gamma = 10\log_{10}[(d_{min}/d)/(E_c/E_u)],$$

which is the gain in signal power that comes from coding minus the power loss for signal set expansion. γ was used originally as the performance measure when searching for trellis codes. It takes into account d_{min} only, without considering any error coefficient. This has been shown to be too optimistic. Recent works have all included at least N_0 together with γ in code search[8][9][10][11].

From simulation[2] it is observed that if P_e is in the range of 10^{-6}, when the error coefficient is double that of uncoded modulation,(e.g., when N_0 is 8 for a 2D code using a rectangular signal constellation) the asymptotic coding gain is reduced by about 0.2 dB. From the approximation $P_e = N_0 Q(\sqrt{d_{min}}/2\delta)$, it is found that a doubling of N_0 from 4 to 8 reduces coding gain by about 0.269 dB when $P_e = 10^{-5}$ and 0.224 dB when $P_e = 10^{-6}$. The "effective coding gain" γ_{eff} is defined by Forney[9] as γ subtracted by 0.2 dB whenever the error coefficient doubles. If more than one error coefficient is known, $\gamma_{eff} = \min[\gamma_{eff}^0, \gamma_{eff}^1, \gamma_{eff}^2]$, where γ_{eff}^i is the effective coding gain for a code with $\hat{d}_{min} = d_{min} + i$, and $\hat{N}_0 = N_i$.

In this work, N_0, N_1, N_2 are computed in the code search. The effective coding gain γ_{eff} is used instead of asymptotic coding gain as a simple and more realistic performance measure.

4.4 Algorithm To Find Error Coefficients

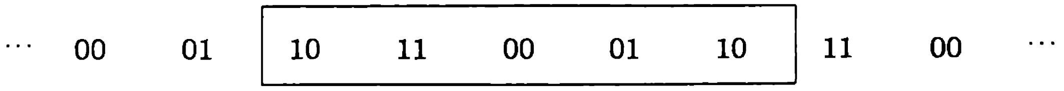
Before finding any error coefficient, we shall first show that only a small portion of the signal constellation needs to be considered if we want to find N_0, N_1, N_2.

In a signal constellation partitioned into subsets, a "basic set" contains a symbol at the center, and all symbols from other subsets that are nearest to the center symbol [2]. Basic sets for signal constellations used in Ungerboeck codes are shown in Fig. 4.12. Each signal is labeled by the corresponding subset. The labeling should be regular, which is the case for Ungerboeck's codes. Under the assumption of infinite constellation, basic sets centered at different symbols has the same size. From the basic set, norms and multiplicities of all subsets can be found.

Say $\{a\}$ is a transmitted coded signal sequence. The subset sequence corresponding to $\{a\}$ is $\{x\}$, where symbols $a^0, a^1, \ldots$ belong to subsets $x^0, x^1, \ldots$. An

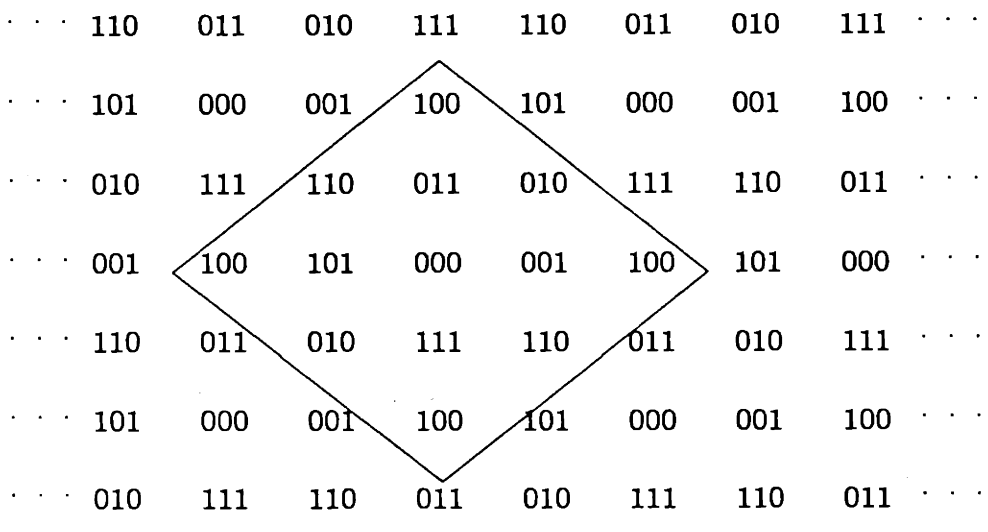
[2] Pottie and Taylor[11] defined the "basic set" to contain only one symbol from each subset, which has enough information for finding d_{min}. In order to find error coefficients, basic set here is defined to contain more symbols.

1D 4-Way Partitioned Constellation



···	00	01	10	11	00	01	10	11	00	···

2D 8-Way Partitioned Constellation



· · ·	110	011	010	111	110	011	010	111	· · ·
· · ·	101	000	001	100	101	000	001	100	· · ·
· · ·	010	111	110	011	010	111	110	011	· · ·
· · ·	001	100	101	000	001	100	101	000	· · ·
· · ·	110	011	010	111	110	011	010	111	· · ·
· · ·	101	000	001	100	101	000	001	100	· · ·
· · ·	010	111	110	011	010	111	110	011	· · ·

Figure 4.12: Basic Sets with Ungerboeck Labelings

error event occurs if the decoder selects a sequence $\{b\}$ instead of $\{a\}$ where $b^0 \neq a^0$. Let the subset sequence for $\{b\}$ be $\{y\}$. If $x^0 \neq y^0$, this error event starts with a node error; otherwise it starts with a parallel transition error. For the node error case, if it is a nearest error, i.e., $\sum_{i=0,1,..} |a^i - b^i|^2 = d_{min}$, and $x^0 \neq y^0$, the individual symbol errors that occur between (a^i, b^i) for all i must be those where b^i is a symbol within the basic sets centered at a^i. If this is not true, then there exists another error event where the decoder selects sequence $\{c\}$ that belongs to subset sequence $\{y\}$, and each symbol c^i is within the basic set centered at a^i. The squared distance between $\{a\}$ and $\{c\}$ will be less than that between $\{a\}$ and $\{b\}$, which is a contradiction. If the error event starts with a parallel transition error, $x^0 = y^0$, the distance between (a^0, b^0) must be $d_{intra}(x^0)$. Actually, all error events at distances $d_{min}, d_{min} + 1, ...d_{min} + S - 1$ are composed either of errors within the basic set or of intra-subset errors. S is the minimum, over all subsets, of the difference in energy of a nearest point outside the basic set and a point inside the basic set. An equivalent statement is that, given norms and multiplicities of all subsets(which can be found from the basic set), and d_{intra}, n_{intra}, error coefficients from N_0 to N_{S-1} can be determined. The one-dimensional 4-way partition has $S = 8$, and the two-dimensional 8-way partition has $S = 4$(see Fig. 4. 4). As a result, N_0, N_1, N_2 for one- and two-dimensional Ungerboeck-type codes can be found with knowledge of the basic set and d_{intra}, n_{intra}, regardless of the actual size of the signal constellation.

To find d_{min}, N_0, N_1, N_2, a modified Viterbi algorithm is used. It is described as the follows:

The trellis search starts from state zero at stage zero. The purpose is to find d_{min}, the minimum distance of paths that diverge from and merge back to state zero, and to find N_0, N_1, N_2, the numbers of those paths with distances d_{min}, $d_{min} + 1$, and $d_{min} + 2$. At stage k, each state i keep the minimum distance of paths from state zero to itself in k steps: $d^k_{min,i}$, and numbers of paths at $d^k_{min,i}$, $d^k_{min,i} + 1$ and $d^k_{min,i} + 2$: $n^k_{0,i}$, $n^k_{1,i}$, $n^k_{2,i}$. Initially $d^0_{min,i} = \infty$, $n^0_{0,i} = n^0_{1,i} = n^0_{2,i} = 0$ for all i. The search then proceeds to stage 1,2,3..., and stops when d_{min}, N_0, N_1, and N_2 are found. At each stage the algorithm iterates over i, j, where i is the "from state" at the previous stage, and j is the "to state" at the current stage.

At stage $k+1$, for each state j (starting from j=0), look at all states i that can be reached back from state j. Say the transition between state i and j corresponds to subset l. Let $D(l)$ be the norm, and $N(l)$ be the multiplicitiy of subset l. Let $d_{temp} = d^k_{min,i} + D(l)$; $d_{old} = d^k_{min,j}$. If $i = j = 0$ then $d_{temp} = \infty$. Let $n_{temp,r} = n^k_{r,i} * N(l)$, $n_{old,r} = n^k_{r,j}$, $r = 0,1,2$.
Comparing d_{temp} with d_{old}, if

1. $d_{temp} > d_{old} + 2 \quad \rightarrow \quad d_{new} = d_{old}, \quad n_{new,r} = n_{old,r},\ r = 0,1,2.$

2. $d_{temp} = d_{old} + 2 \quad \rightarrow \quad d_{new} = d_{old}, \quad n_{new,2} = n_{old,2} + n_{temp,0},$
   $n_{new,r} = n_{old,r},\ r = 0,1.$

3. $d_{temp} = d_{old} + 1 \quad \rightarrow \quad d_{new} = d_{old}, \quad n_{new,2} = n_{old,2} + n_{temp,1},$
   $n_{new,1} = n_{old,1} + n_{temp,0}, \quad n_{new,0} = n_{old,0}.$

4. $d_{temp} = d_{old} \quad \rightarrow \quad d_{new} = d_{old}, \quad n_{new,r} = n_{old,r} + n_{temp,r},\ r = 0,1,2.$

5. $d_{temp} = d_{old} - 1 \quad \rightarrow \quad d_{new} = d_{temp}, \quad n_{new,2} = n_{old,1} + n_{temp,2},$
   $n_{new,1} = n_{old,0} + n_{temp,1}, \quad n_{new,0} = n_{temp,0}.$

6. $d_{temp} = d_{old} - 2 \quad \rightarrow \quad d_{new} = d_{temp}, \quad n_{new,2} = n_{old,0} + n_{temp,2},$
   $n_{new,1} = n_{temp,1}, \quad n_{new,0} = n_{temp,0}.$

7. $d_{temp} < d_{old} - 2 \quad \rightarrow \quad d_{new} = d_{temp}, \quad n_{new,r} = n_{temp,r},\ r = 0,1,2.$

Let $d^{k+1}_{min,j} = d_{new}$, $n^{k+1}_{r,j} = n_{new,r}$, $r = 0,1,2$. Repeat the above procedures and update $d^{k+1}_{min,j}$, $n^{k+1}_{r,j}$, $r = 0,1,2$ for each state i, until all states i connected to state j are visited. Then, go to state $j+1$ and find $d^{k+1}_{min,j+1}$, $n^{k+1}_{r,j+1}$, $r = 0,1,2$. When all states j are visited, if for all $j \neq 0$, $d^{k+1}_{min,j} > d^{k+1}_{min,0} + 2 \rightarrow$ stop, since no shorter paths can be found if search further. Otherwise, go to stage $k+2$. In general, the iteration over k will stop for $k \leq 6 * \nu$ when searching for 2^{ν}-state codes. When the iteration stops at stage k_0, we find for this code:

- $d_{min} = d^{k_0}_{min,0}$,
- $N_2 = n^{k_0}_{2,0}$, $N_1 = n^{k_0}_{1,0}$, $N_0 = n^{k_0}_{0,0}$.

This algorithm is used to search for Ungerboeck-type codes with improved effective coding gain. Results of the search are presented in the next chapter.

Chapter 5

Code Search Results

Applying the algorithm for finding error coefficients N_0, N_1, N_2, a code search is conducted over Ungerboeck-type codes using one- and two-dimensional rectangular signal constellations.

The nearly exhaustive search procedure follows mostly that of Ungerboeck[1]. The encoder is in systematic feedback form. Search is carried out for one- and two-dimensional codes up to 256 states. In the search, for each code we build a code trellis according to it's parity check polynomials, and then find d_{min}, N_0, N_1 and N_2. We proceed to other codes by varying the parity check polynomials. The minimum distance for each code is compared with the largest value found earlier; if the new d_{min} is larger, the old value is replaced. A few modifications of the search procedure are:

- Rejection rules that reject codes with the same minimum distances are not used. Two codes with the same d_{min} might have different error coefficients, and thus differ in their effective coding gain.
- For complicated codes where the number of codes to be examined is too large, d_{min} is computed first; for codes with "good" d_{min}, N_0, N_1, and N_2 are then computed.
- A code with slightly less d_{min} might have significantly smaller error coefficients compared to a code with larger d_{min}. Therefore, codes with a minimum dis-

tance that is one less than the largest d_{min} found previously are still considered to have "good" d_{min}.

Some new codes are found with better effective coding gain than the codes Ungerboeck proposed, and those found by Honig (one-dimensional), Pottie and Taylor (two-dimensional). We also computed error coefficients N_0, N_1 and N_2 for codes found earlier. Honig's 16-state code was found to be catastrophic[1]. Pottie and Taylor's 4-state code has a N_0 of 20 instead of 4.

The following tables and plot for Ungerboeck-type trellis codes are from Forney's "Coset Code I", a comprehensive tutorial of all coded modulation schemes.

The tables are for one-dimensional codes with 4-way and two-dimensional codes with 8-way partitioned rectangular constellations. 2^{ν} is the number of encoder states, h^i's $i = 0, 1, 2$ are parity check polynomials(in octal form), d_{min} is the minimum squared Euclidean distance, γ is the asymptotic coding gain both in ratio and in dB, $\tilde{N}_0$, $\tilde{N}_1$, and $\tilde{N}_2$ are error coefficients normalized to two dimensions, and γ_{eff} is the effective coding gain. At the last column "U" indicates Ungerboeck's codes; "H" indicates codes found by Honig; "PT" stands for Pottie and Taylor, and "EL" indicates Eyüboğlu and Li[2]. Codes with starred N_1 or N_2 have these error coefficients much larger than N_0, and thus the effective coding gain is determined by N_1, $d_{min} + 1$, or N_2, $d_{min} + 2$, instead of N_0, d_{min}.

[1] Catastrophic codes have two code sequences with finite Euclidean distance that correspond to input sequences with infinite Hamming distance.

[2] This code search was done as part of this thesis research in Codex Corporation, Summer 1987, under supervision of Dr. V. Eyüboğlu and Dr. G. D. Forney.

Table V-3. Effective coding gains for Z/4Z codes

2^ν	h^1	h^0	d_{min}^2	γ	dB	$\bar{N}_0$	$\bar{N}_1$	$\bar{N}_2$	γ_{eff}(dB)	
4	2	5	9	2.25	3.52	8	16	32	3.32	U
8	04	13	10	2.5	3.98	8	16	32	3.78	U
16	04	23	11	2.75	4.39	16	16	32	3.99	U
16	10	23	11	2.75	4.39	8	16	48	4.19	EL
32	10	45	13	3.25	5.12	24	56	112	4.60	U
64	024	103	14	3.5	5.44	72	0	180	4.61	U
64	054	161	14	3.5	5.44	16	*64	132	4.94	H
128	126	235	16	4	6.02	132	0	512	5.01	U
128	160	267	15	3.75	5.74	16	68	*200	5.16	EL
128	124	207	14	3.5	5.44	8	16	28	5.24	EL
256	362	515	16	4	6.02	4	64	*160	5.47	U
256	370	515	15	3.75	5.74	8	12	*80	5.42	EL
512	0342	1017	16	4	6.02	4	0	*112	5.57	U

Table V-4. Effective coding gains for $Z^2/2RZ^2$ codes

2^ν	h^2	h^1	h^0	d_{min}^2	γ	dB	$\bar{N}_0$	$\bar{N}_1$	$\bar{N}_2$	γ_{eff}(dB)	
4	—	2	5	4	2	3.01	4	32	128	3.01	U
8	04	02	11	5	2.5	3.98	16	72	320	3.58	U
16	16	04	23	6	3	4.77	56	160	820	4.01	U
32	10	06	41	6	3	4.77	16	104	404	4.37	U
32	34	16	45	6	3	4.77	8	*128	404	4.44	EL
64	064	016	101	7	3.5	5.44	56	260	1008	4.68	U
64	060	004	143	7	3.5	5.44	48	292	1184	4.72	PT
64	036	052	115	7	3.5	5.44	40	252	992	4.78	EL
128	042	014	203	8	4	6.02	344	0	5900	4.74	U
128	056	150	223	8	4	6.02	172	624	2568	4.94	EL
128	024	100	245	7	3.5	5.44	8	*188	968	4.91	PT
128	164	142	263	7	3.5	5.44	8	*132	752	5.01	EL
256	304	056	401	8	4	6.02	44	*304	1316	5.28	U
256	370	272	417	8	4	6.02	36	*308	1224	5.28	EL
256	274	162	401	7	3.5	5.44	4	*64	248	5.22	EL
512	0510	0346	1001	8	4	6.02	4	128	*700	5.50	U

The plot of performance versus complexity illustrates clearly the growth of coding gain when using more complicated codes. The effective coding gain is in dB. The decoding complexity, defined by Forney[9], is the number of decoding operations(addition and comparison) needed for deciding one point nearest to the received point in each subset, followed by a conventional Viterbi algorithm for the convolutional code. The normalized complexity is the decoding complexity per two dimensions.

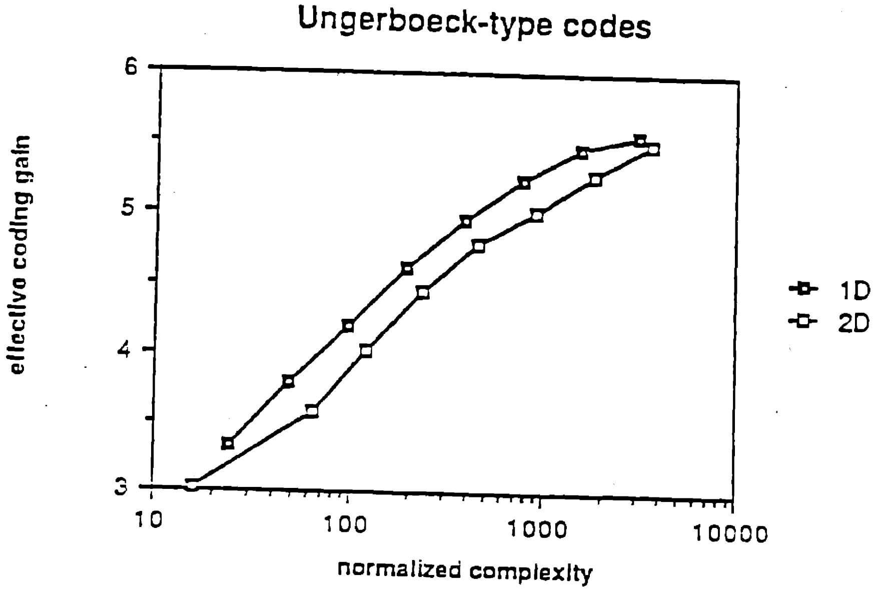


Figure 12-1. Performance vs. complexity for Ungerboeck-type one-dimensional and two-dimensional codes (as improved by Eyuboglu and Li).

Chapter 6

Conclusion

In this work the following things are done:

- The general structure of binary regular labelings for rectangular constellations is studied. It is found that the largest number of partitions where regular labeling exists is 2^{2m} for m-dimensional constellations. Structures of regular labelings for m-dimensional 2^{2m}-way partitions are found. Structures of regular labelings for fewer than 2^{2m}-way partitions in one and two dimensions are also determined.

- Performance measures for trellis codes are discussed and compared, including the approximation and upper bound for error event probability P_e, the asymptotic coding gain, and the effective coding gain.

- Recognizing that the first few error coefficients can be computed easily, a modified Viterbi algorithm for finding the first three error coefficients in addition to the minimum distance is proposed.

- Improved Ungerboeck-type codes with one- and two-dimensional rectangular signal sets are found by considering the first three error coefficients in addition to the minimum distance.

Ten years after Ungerboeck proposed his codes, with all the new schemes that open up new horizons for coded modulations, we still learn by looking at earlier schemes. Although the ideas of Ungerboeck-type codes are not new, the improved codes tell us that it is both necessary and easy to look at more error coefficients when finding good trellis codes; also the trade-off between minimum distance and error coefficient should be considered. Up to now no coded modulation scheme has used more than 2^{2m}-way partitioned m-dimensional rectangular constellations, where good codes are very complicated. From the structures of regular labeling we learned that regular trellis codes actually do not exist there.

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] G. Ungerboeck,"Trellis Coded Modulation with Redundant Signal Sets, Part I,II,"*IEEE Commun. Mag.*,Vol.25, No.2, pp.5-21, Feb. 1987

[3] G. D. Forney, Jr., R. G. Gallager, G. R. Lang, F. M. Longstaff and S. U. Qureshi,"Efficient Modulation for Bandlimited Channels,"*IEEE J. Select. Areas Commun.*,Vol. SAE-2, pp. 632-647 Sept. 1984

[4] R. G. Gallager, *Information Theory and Reliable Communication,* New York: Wiley, 1968.

[5] G. D. Forney, "The Viterbi Algorithm,"*Proc. of IEEE*, Vol. 61, No. 3, pp. 268-278, March 1973

[6] 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, March 1987.

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

[8] L. F. Wei, "Trellis-Coded Modulation with Multidimensional Constellations,"*IEEE Trans. Inform. Theory*, Vol. IT-33, pp.483-501, July, 1987.

[9] G. D. Forney, Jr. “Coset Codes I, II,” to appear, *IEEE Trans. Inform. Theory.*, 1988

[10] M. L. Honig,“Optimization of Trellis Codes with multilevel amplitude modulation with respect to an error probability criterion,” *IEEE Trans. Commun.*, Vol. COM-34, No.8, Aug. 1986.

[11] G. J. Pottie and D. P. Taylor,“An approach to Ungerboeck coding for rectangular signal sets,”*IEEE Trans. Info. Theory* Vol. IT-33, pp. 285-290, March 1987.

[12] A. J. Viterbi and J. K. Omura, *Principles of Digital Communication and Coding.* New York: McGraw-Hill, 1979.

[13] E. Zehavi and J. K. Wolf, “On the Performance Evaluation of Trellis Codes,” *IEEE Trans. Inform. Theory*, Vol. IT-33, pp.196-202, March 1987.

[14] S. Benedetto, M. A. Marsan, G. Albertengo, E. Giachin, “Combined Coding and Modulation: Theory and Applications,”*IEEE Trans. Info. Theory* Vol.IT-34, pp. 223-236, March 1988.