UNIVERSITY OF CALIFORNIA

Los Angeles

Trellis Codes Design for Periodic Erasures and Adaptive Coded Modulation Schemes for Time-Varying Channels

A dissertation submitted in partial satisfaction of the
requirements for the degree Doctor of Philosophy
in Electrical Engineering

by

Xueting Liu

2000

The dissertation of Xueting Liu is approved.

Mario Gerla

Gregory J. Pottie

Kung Yao

Richard D. Wesel, Committee Chair

University of California, Los Angeles

2000

To my parents

Contents

List of Figures

List of Tables

ACKNOWLEDGEMENTS

I would like to thank Professor Rick Wesel for being my advisor. He is such a great, wonderful advisor that I do not even know when to start to show my appreciation. I appreciate his kindness to accept me joining his fine research group. I appreciate his continuous guidance, encouragement and financial support during my working on this dissertation. He really cares about his students and provides the best environment a student could wish for. His active involvement in research, his always availability for questions and comments, his enthusiasm and his fascination with the labeling problems are always appreciated and are always in my memory. He helps and inspires me not only in academic areas, but also in nonacademic areas, like how to communicate with people and how to be a successful professional. Without his constant help and support, this dissertation could not have been done.

I thank Professor Stephen Jacobsen for promoting my admission at UCLA and for being my advisor during my first year at UCLA.

I was also fortunate to have the opportunity to work with Professor Abeer Alwan and Alexis Bernard. Their involvement in our research are greatly appreciated.

I also thank Professor Dennis Goeckel and his student Pinar Ormeci at University of Massachusetts, Amherst. I benefited a lot from our cooperation.

I am indebted to Professors Mario Gerla, Gregory Pottie and Kung Yao for taking the time to serve on my PH.D. committee.

My fellow group members Christine Fragouli, Christos Komninakis, Wei shi and later Mark Shane, Adina Matache, Cenk Kose have greatly contributed to my research with their useful comments and interaction, and have created a friendly and inspiring working environment. Thanks to Christos for being the "manager" of our group; thanks to Christine for her friendship and thanks to Wei for "gossiping" with me. Also I would like to welcome the newly-coming group member: Tom Sun (also Wei's husband).

My fellow students and good friends Brien Alkire, Shai Benjamin, Chi-Ching Chen, John Siwko, Yung-Szu Tu, Tai-Lai Tung, Tommy Yu and other OC Lab fellows made my stay at UCLA an enjoyable journey. I appreciate all your kind help and the happy time we had together.

I thank my friends Tao Zhang for always being on my side and Hong Chen for her hospitality.

Special thanks to my long-time friend Junling Li for everything. He is such a great friend that whenever I have a problem, I can always turn to him for suggestions.

Finally, I would like to thank my parents, my sisters Xuejun and Xueli, my brothers Weifeng and Lixin, my niece Minwen and my nephew Brandon for their immeasurable love and support.

VITA

1992	B. S., Automatic Control Engineering Central South University of Technology Changsha, P. R. China
1995	M. S., Systems Optimization Institute of Automation, Chinese Academy of Sciences Beijing, P. R. China
2000	Ph. D., Communications and Telecommunications Electrical Engineering Department University of California, Los Angeles Los Angeles, CA

PUBLICATIONS AND PRESENTATIONS

P. Ormeci, X. Liu, D. L. Goeckel and R. D. Wesel, "Adaptive Bit-Interleaved Coded Modulation", (accepted), *IEEE Trans. Communication.*

R. D. Wesel, X. Liu and W. Shi, "Trellis Codes for Periodic Erasures", *IEEE Transactions on Communications,* vol. 48, no. 6, pp. 938-947, June 2000.

R. D. Wesel and X. Liu, "Edge Profile Optimal Constellation Labeling", *Proc. IEEE ICC,* June 2000.

A. Bernard, X. Liu, R. D. Wesel and A. Alwan, "Embedded Joint Source-Channel Coding of Speech Using Symbol Puncturing of Trellis Codes", *Proc. IEEE ICASSP,* March, 1999.

A. Bernard, X. Liu, R. Wesel and A. Alwan, "Channel Adaptive Joint Source-Channel Coding of Speech", *Proc. of the 32nd Asilomar Conference on Signals, Systems, and Computers,* November, 1998.

X. Liu and R. D. Wesel, "Profile Optimal 8-QAM and 32-QAM Constellations", *Proc. of 36th Annual Allerton Conference on Communication, Control and Computing,* September 1998.

R. D. Wesel and X. Liu, "Analytic Techniques for Periodic Trellis Codes", *Proc. of 36th Annual Allerton Conference on Communication, Control and Computing,* September 1998.

R. D. Wesel, X. Liu, W. Shi and J. M. Cioffi, "Trellis Codes for Compound Periodic Gaussian Channels", *Proc. of IEEE-ISI,* August 1998.

R. D. Wesel, C. Komninakis and X. Liu, "Towards Optimality in Constellation Labeling", *Proc. IEEE-GLOBECOM,* November 1997.

R. D. Wesel, X. Liu, W. Shi, "Periodic Symbol Puncturing of Trellis Codes", *Proc. 31st Asilomar Conference on Signals, Systems and Computers,* November 1997.

ABSTRACT OF THE DISSERTATION

Trellis Codes Design for Periodic Erasures and Adaptive Coded Modulation Schemes for Time-Varying Channels

by

Xueting Liu

Doctor of Philosophy in Electrical Engineering
University of California, Los Angeles, 2000
Professor Richard D. Wesel, Chair

This thesis addresses trellis codes design for periodic erasures and adaptive coded modulations for time-varying channels. In periodic erasures where each channel is characterized by additive white Gaussian noise with a distinct periodic variation in signal-to-noise ratio, the design of trellis codes becomes a multi-criterion optimization problem. We propose two objective functions to search the optimal codes. Thirteen codes from searches that maximize the indicated objective functions are provided. These trellis codes designed specifically for periodic erasures are also the optimal mother codes of generating families of rate-compatible, unequal error protection trellis codes by periodic symbol-puncturing. We apply them to an embedded joint source-channel coding scheme of speech. The simulation results show that these codes are suitable for unequal error protection applications. The design of trellis codes involves both convolutional codes and constellations. We investigate cross constellations labelings in the context of the edge profile.

For time-varying channels, one possible method to achieve robust and spectrally efficient communication is to use adaptive coded modulations. In systems supporting highly-mobile users that can endure long-latency, adaptive bit-interleaved coded modulation (BICM) achieves better performance than previously proposed adaptive and nonadaptive schemes. A comparison of robustness to channel prediction errors between trellis codes and BICM codes is present. However, in systems with highly-mobile uses requiring low-latency, adaptive BICM is not a good choice due to the delay introduced by the bit interleaver. We propose two adaptive coded modulation (ACM) schemes for this scenario. One of the ACM schemes uses a forward trellis search algorithm (FTS) to adapt to the current channel fading. Numerical results demonstrate that the proposed FTS-ACM scheme achieves a bandwidth efficiency gain equal or better than that of adaptive BICM.

Chapter 1

Introduction

The properties of adaptivity and robustness are desired in communication systems. Adaptivity changes the transmission to match the estimated channel and traffic conditions while robustness allows the same transmission to work consistently well for a wide range of channel conditions.

The main objective of this thesis is to investigate techniques that improve the properties of adaptivity and robustness in communications systems. Several specific results related to such improvement are presented in this thesis.

This thesis is organized as follows. Chapter 2 is about the design of trellis codes for periodic erasures and their applications. In periodic erasures where each channel is characterized by additive white Gaussian noise with a distinct periodic variation in signal-to-noise ratio, the design of trellis codes is a multi-criterion optimization problem. We propose two objective functions to search the optimal codes. Thirteen codes from searches that maximize the indicated objective functions are provided. Those trellis codes provide reliable communication over all periodic erasure patterns of a given period for which the number of unerased coded bits per period is at least equal to the number of information bits per period.

In addition, those trellis codes designed specifically for periodic erasures are also the optimal mother codes of generating families of rate-compatible, unequal error protection trellis codes by using periodic symbol-puncturing. We apply them to an embedded joint source-channel coding scheme of speech. The simulation results show that those codes are suitable for unequal error protection applications and perform well over a wide range of channel conditions in such a joint source-channel coding schemes of speech.

The design of a trellis code involves the selection of not only a convolutional encoder but also a labeled constellation to maximize some subset of the metrics presented in [16] [29] [4]. A new constellation labeling paradigm which optimizes the edge profile was proposed in [20] and was applied to 8-QAM and 32-QAM constellations in [21]. This new constellation labeling paradigm offers better desired metrics than set-partitioning and Gray labeling in channels with puncturing or fading. Chapter 3 investigates cross constellations in the context of optimal edge profiles. Six distinct classes of quasi-Gray labeled 32-QAM are obtained. Chapter 3 also identifies 8-QAM constellations that have the best edge profile and the best edge profile for quasi-Gray labeled 32-QAM.

For time-varying channels, one possible method to achieve robust and spectrally efficient communication is to use adaptive coded modulation schemes. In systems supporting highly mobile users that can endure long-latency, adaptive bit-interleaved coded modulation (BICM) achieves a moderate bandwidth efficiency gain over previously proposed adaptive schemes and nonadaptive schemes of similar complexity. A comparison of robustness to channel prediction errors between trellis codes and bit-interleaved coded modulation is present in Chapter 4. This comparison shows that the structure of BICM is more robust to prediction errors than trellis codes modulation, thus motivates the use of adaptive BICM systems for channels supporting users with high mobility.

However, adaptive BICM requires a bit interleaver, which results in long latency. Therefore, in systems with highly mobile users requiring low-latency, adaptive BICM is not a good choice. We propose two adaptive coded modulation (ACM) schemes for this scenario in Chapter 5. One of the ACM schemes uses a forward trellis search algorithm (FTS) to adapt to the current channel fading. Numerical results demonstrate that the proposed FTS-ACM scheme achieves a bandwidth efficiency gain equal or better than that of adaptive BICM. FTS-ACM is particularly attractive for low latency transmission applications.

The main contributions of the work presented in this thesis, are:

- Proposed the design procedure of trellis codes for periodic erasures and present thirteen codes from searches that maximize the indicated objective functions (Chapter 2).

- Applied the above optimal codes to an embedded joint source-channel coding scheme of speech to achieve adaptive multi-rate speech transmission (Chapter 2).

- Investigated the cross constellations in the context of optimal edge profile. Showed that there are six distinct classes of quasi-Gray labeled 32-QAM and identified 8-QAM constellations that have the best edge profile and found the best edge profile for quasi-Gray labeled 32-QAM (Chapter 3).

- Comparison of robustness to channel prediction errors between trellis codes and bit-interleaved coded modulation using the transfer function bound (Chapter 4).

- Proposed and compared two adaptive coded modulation schemes for systems with highly mobile users requiring low-latency (Chapter 5).

- Compared the performance between our proposed adaptive coded modulation schemes and adaptive bit interleaved coded modulations (Chapter 5).

Chapter 2

Trellis Codes for Periodic Erasures and Applications

Partial band interference often distorts frequency hopped or multi-carrier transmissions. This interference might be caused by a malicious jammer in a military communications scenario or by adjacent channel interference in a multi-carrier digital audio broadcast scenario. In both cases the partial band interference is contiguous and may be transformed to a periodic erasure pattern by a block interleaver. This chapter considers the design of trellis codes specifically for such periodic erasure scenarios. The design procedure is actually more general than the periodic erasure case; it applies to any small set of periodic attenuations.

The channel of interest has input x_i and output $y_i = \tilde{a}_{i (\mathrm{mod}\ p)} x_i + z_i$ where z_i is additive white Gaussian noise (AWGN) with variance $N_0/2$ per dimension. The p-element vector $\tilde{a} = [\tilde{a}_1 \ \ldots \ \tilde{a}_p]$ contains the periodic scale factors with $\tilde{a}^2 = [|\tilde{a}_1|^2 \ \ldots \ |\tilde{a}_p|^2]$. The channel inputs x_i are 2D constellation points with transmitter energy $\mathcal{E}_x$ per dimension, $\bar{\mathcal{E}}_x = \mathcal{E}_x/2$.

The scope of this problem depends on how much the transmitter knows about

$\tilde{a}$. If the transmitter has complete knowledge of $\tilde{a}$ through timely feedback, the trellis code may be adapted to the specific $\tilde{a}$ as with discrete multi-tone for point-to-point asynchronous digital subscriber lines [1]. When the transmitter knows nothing about $\tilde{a}$ except its period, the appropriate code design metrics are periodic effective code length (PECL) and periodic product distance (PPD) [2], [3], [4], [5], [6], [7], [8], [9], [10], [11].

This chapter addresses an intermediate design problem where $\tilde{a}$ belongs to a known family of periodic attenuations. In this situation the code design problem is a multi-criterion optimization problem where the designer seeks to simultaneously optimize performance for each $\tilde{a}$ in the family of potential periodic attenuations. One important practical application for this code design problem is the design of a code to withstand periodic erasures.

Section 2.1 presents theoretical performance limits for providing reliable transmission for every $\tilde{a}$ in a specified family. Section 2.2 presents two objective functions for the multi-criterion optimization problem of identifying a code that performs well for every $\tilde{a}$ in the specified family. Section 2.3 discusses the practical performance limits implied by finite constellation sizes and finite memory encoders. When designing codes for periodic erasures, special consideration must be given to catastrophic behavior. Section 2.4 explores how trellis codes that do not display catastrophic behavior under normal AWGN may become catastrophic when subjected to periodic erasures. Section 2.5 presents thirteen trellis codes designed for families of periodic erasure patterns with $2 \leq p \leq 5$. Section 2.6 gives an example of those trellis codes applied in an adaptive multi-rate speech transmission. Section 2.7 summarizes the chapter.

2.1 Theoretical Performance Limits

In the information theory literature, a compound channel occurs when the actual channel is unknown to both transmitter and receiver but belongs to a family of possible channels known to both. In this chapter we refer to the compound channel described in the introduction with AWGN and a periodic attenuation vector $\tilde{a}$ as the compound periodic Gaussian channel.

The compound channel capacity $C(\mathcal{C})$ is the highest information rate that a single code can achieve reliably on every channel in the family $\mathcal{C}$ of possible channels. Root and Varaiya's compound channel coding theorems [12] provide the compound channel capacities for general classes of compound linear Gaussian channels. Their discrete-time results may be applied to the compound periodic Gaussian channel by choosing diagonal channel and noise covariance matrices. The proof in [12] specifies two mild conditions that for the compound periodic Gaussian channel become the following statement: There must exist nonnegative finite real numbers $t_1, t_2,$ and t_3 such that every possible channel satisfies $|\tilde{a}_i| \leq t_1$ and $t_2 \leq N_0/2 \leq t_3$. These conditions are required in [12] to bound the space of channels that must be considered by a single decoding algorithm. They are not necessary if the channel may be estimated accurately at the receiver and provided to the decoder as side information.

Each channel in the family is described by an attenuation vector $\tilde{a}$ and a corresponding $\mathcal{E}_x/N_0$. The family of attenuation vectors (erasure patterns) might be known at the time of code design, but typically $\mathcal{E}_x/N_0$ will vary with the position of the wireless transceivers. However, one may fix the desired per-symbol rate R and determine the $\mathcal{E}_x/N_0$ values that provide sufficient mutual information (MI) for each $\tilde{a}$. The $\mathcal{E}_x/N_0$ for which MI $> R$ for a given $\tilde{a}$ depends on the transmitter power distribution over the period of p symbols. For specified $\mathcal{E}_x$, N_0, and $\tilde{a}$, water-filling applied to the p symbols transmitted in a period maximizes the MI. However, a single power allocation won't generally maximize MI for every channel in the family of possible channels.

For a well-defined family of channels where $\mathcal{E}_x/N_0$ is specified for each $\tilde{a}$, the allocation that maximizes the minimum MI (i.e. achieves the compound channel capacity) can be found explicitly. More generally, statistical knowledge about $\tilde{a}$ may indicate a certain power allocation. When stationary correlated fading or interference is mitigated by a sufficiently deep block interleaver, the a_i terms are approximately independent and identically distributed [7]. In such cases the family of attenuation vectors $\tilde{a}$ is closed under permutations. The symmetry of this situation indicates that transmitting every symbol with the same energy maximizes the compound channel capacity. Applying the results of [12] with an equal allocation of power to each symbol in the period, one may reliably transmit R bits per two-dimensional symbol over all period-p periodic Gaussian channels such that

$$\frac{1}{p}\sum_{i=1}^{p}\log\left(1+\frac{|\tilde{a}_i|^2\mathcal{E}_x}{N_0}\right) > R\,, \tag{2.1}$$

where the l.h.s. of (2.1) is the channel MI.

Now consider how closely a rate-1/3, 8-PSK, 64-state trellis code designed later in this chapter (code #13) approaches the performance limit of (2.1) for period-five erasure patterns. Fig. 2.1 shows the standard plot of BER vs. $\mathcal{E}_x/N_0$ using Monte Carlo simulation.

This standard plot shows the $\mathcal{E}_x/N_0$ requirements of each $\tilde{a}$, but it does not characterize how well the trellis code is performing relative to (2.1). To do this, Fig. 2.2 plots BER vs. MI.

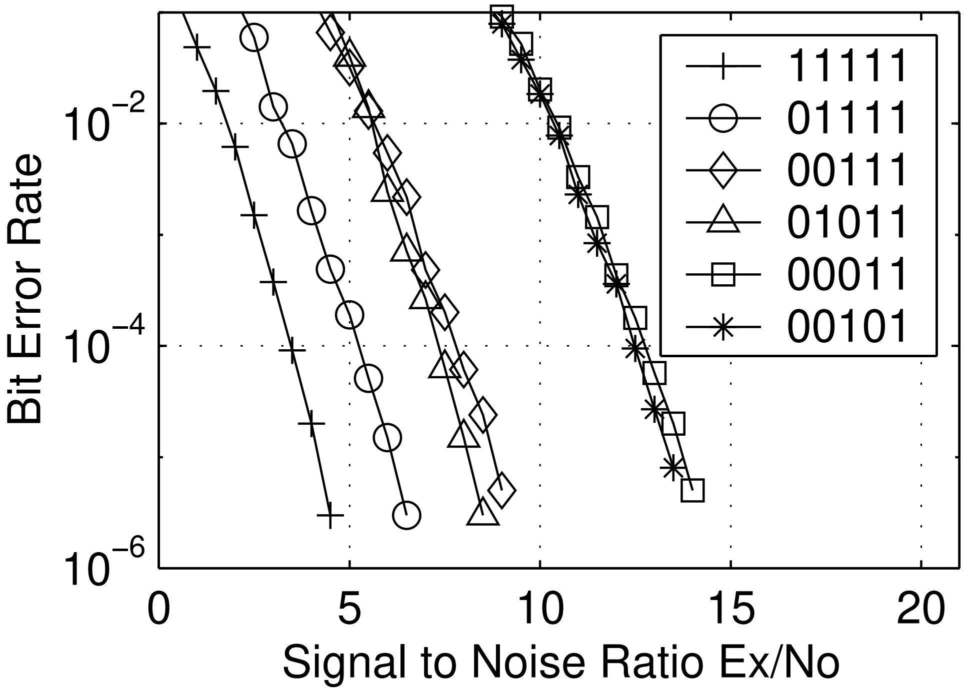


Figure 2.1: Bit error rate vs. signal-to-noise ratio $\mathcal{E}_x/N_0$ using rate-1/3 TCM (code #13 in Table 2.1) for six periodic erasure patterns.

Now the consistent performance of code #13 over the six erasure channels is clear. At BER $= 10^{-5}$ this code consistently requires an MI of between 1.74 and 1.88 bits; it consistently requires an excess MI of between 0.74 and 0.88 bits.

The MI requirement can be expressed in more familiar terms by computing the corresponding SNR in dB for the standard AWGN channel. The top of Fig. 2.2 labels the x-axis with "equivalent AWGN SNR", the $\mathcal{E}_x/N_0$ value that provides AWGN capacity equal to the MI value labeling the x-axis. Equivalent AWGN SNR is computed as $2^{\mathrm{MI}} - 1$. The MI of 1.74–1.88 bits corresponds to an equivalent AWGN SNR of 3.7–4.3 dB. For comparison, the maximum Euclidean distance 64-state trellis code for transmission of one bit per symbol (rate-1/2 maximum

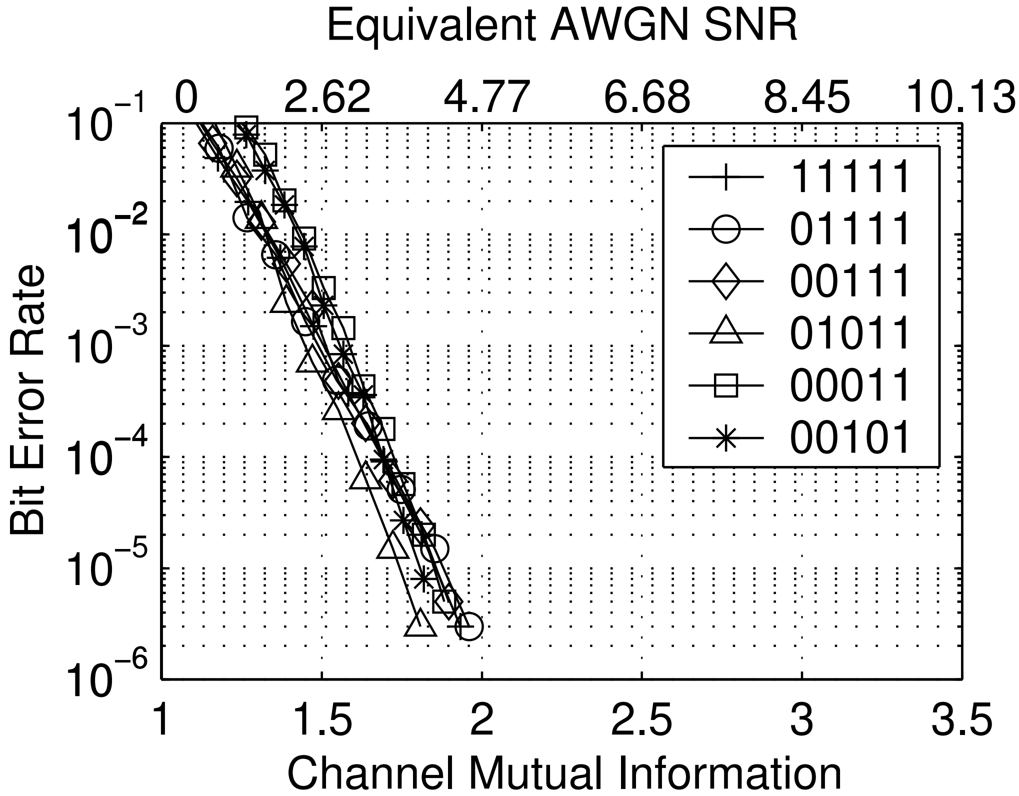


Figure 2.2: Bit error rate vs. channel mutual information, i.e. the l.h.s. of (2.1), for six periodic erasure patterns for rate-1/3 TCM code #13 in Table 2.1.

Hamming distance convolutional code used with Gray-labeled 4-PSK) requires an equivalent AWGN SNR of 4.3 dB for BER $= 10^{-5}$ on the AWGN channel.

2.2 A Multi-criterion Optimization Problem

The design of a trellis code to approach the information-theoretic performance limits described in the previous section is a multi-criterion optimization problem. With finite complexity, the best performance on one particular channel will often be obtained at the expense of sub-optimal performance on some other channel(s) in the family. We refer to a trellis code as undominated if no trellis code (of the same complexity) performs better on every channel in the family. Typically, there will be several undominated trellis codes. Such undominated solutions are called Pareto optimal in the multi-criterion optimization literature.

A natural extension of the standard AWGN notion of designing a trellis code to maximize Euclidean distance is to maximize the residual Euclidean distance (RED) remaining after periodic attenuation (or periodic erasure) by each $\tilde{a}$ in the family of possible channels. However, as discussed above one trellis code generally won't have the largest $\mathrm{RED}^2(\tilde{a})$ for every $\tilde{a} \in \mathcal{C}$. After formally defining RED, we present two objective functions for selecting among the Pareto optimal codes.

Let the normalized symbol-wise squared Euclidean distance between two constellation points be $d_j^2(x \to \hat{x}) = \|\hat{x}_j - x_j\|^2/\mathcal{E}_x$ where x_j and $\hat{x}_j$ are the correct and incorrect constellation points associated with the jth symbol of a trellis error event $x \to \hat{x}$. The periodic Gaussian channel scales distances with the same index modulo p by the same scale factor $\tilde{a}_i$. Define the periodic squared distance $\tilde{d}_i^2(x \to \hat{x})$ for $i \in \{1, 2, \ldots p\}$ as the sum of the distances scaled by the same $\tilde{a}_i$ for a given trellis error event $x \to \hat{x}$: $\tilde{d}_i^2 = \sum_{m=0}^{\infty} d_{i+mp}^2$. The p values of $\tilde{d}_i^2$ comprise the periodic squared distance vector $\tilde{d}^2 = \begin{bmatrix} \tilde{d}_1^2 & \tilde{d}_2^2 & \ldots & \tilde{d}_p^2 \end{bmatrix}$. To simplify notation, we suppress the error event argument $(x \to \hat{x})$ in d_i^2, $\tilde{d}_i^2$, and $\tilde{d}^2$. Minimization over the periodic distance vectors of valid error events provides the squared RED for a specified periodic attenuation vector $\tilde{a}$:

$$\mathrm{RED}^2(\tilde{a}) = \min_{\tilde{d}^2} \langle \tilde{a}^2, \tilde{d}^2 \rangle \tag{2.2}$$

where the expression $\langle \tilde{a}^2, \tilde{d}^2 \rangle$ represents an inner product.

Either forward trellis search employing the Viterbi algorithm or the bi-directional stack algorithm [13] may be extended to the periodic case to compute RED. The extended stack algorithm requires conversion to a strictly equivalent multiple trellis code [14], where each branch distance of the multiple trellis code minimal state diagram is the inner product of $\tilde{a}$ with the vector of p associated constellation-point distances.

Figs. 2.1 and 2.2 motivate two different objective functions for selecting among the Pareto optimal codes. From the perspective of Fig. 2.1., we seek a trellis code that minimizes the sum of the dB gaps from some fixed SNR point at a particular BER. Using only RED values limits consideration to the sum of the dB values required to maintain a specified maximum pairwise error probability. For a specified $\tilde{a}$, the pairwise error probability for the trellis error event $x \rightarrow \hat{x}$ is

$$P(x \rightarrow \hat{x}) = Q\left(\sqrt{\frac{\mathcal{E}_x \langle \tilde{a}^2, \tilde{d}^2 \rangle}{2N_0}}\right), \tag{2.3}$$

where $Q(t) = \frac{1}{\sqrt{2\pi}} \int_t^\infty e^{-u^2/2} du$ and $\tilde{d}^2$ is the periodic distance vector for $x \rightarrow \hat{x}$. Note that $\text{RED}^2(\tilde{a})$ identifies the smallest $\langle \tilde{a}^2, \tilde{d}^2 \rangle$, the one that maximizes (2.3).

Let the jth vector $\tilde{a}$ in $\mathcal{C}$ be $\tilde{a}^{(j)}$, abbreviate $\text{RED}^2(\tilde{a}^{(j)})$ to RED_j^2, and define SNR_j as the value of $\mathcal{E}_x/N_0$ for which the target pairwise error probability is achieved for $\tilde{a}^{(j)}$. We seek the trellis code that minimizes sum of $10\log_{10} \text{SNR}_j$. Achieving a target worst-case $P(x \rightarrow \hat{x})$ is equivalent to achieving $\text{RED}_j^2 \mathcal{E}_x/N_0 > \lambda$ for each vector $\tilde{a}^{(j)}$. Allowing the SNR_js to be slack variables produces the following dB-sum minimization problem:

$$\text{Minimize} \qquad \sum_j 10\log_{10} \text{SNR}_j \tag{2.4}$$

$$\text{Subject to} \qquad 10\log_{10} \text{SNR}_j + 10\log_{10} \text{RED}_j^2 \geq 10\log_{10} \lambda\,. \tag{2.5}$$

The optimal trellis code in this sense is the one that maximizes $J_{dB} = \sum_j 10\log_{10} \text{RED}_j^2$.

Fig. 2.2 motivates trellis codes that minimize the sum of MIs required to maintain a desired error probability. Equation (2.6) gives a high-$\mathcal{E}_x/N_0$ approximation for the l.h.s. of (2.1).

$$\mathrm{MI}_j \approx \frac{1}{p} \sum_{i:\tilde{a}_i^{(j)} \neq 0} \log_2 \left(\frac{|\tilde{a}_i^{(j)}|^2 \mathcal{E}_x}{N_0} \right) . \tag{2.6}$$

With the approximation in (2.6), the MI-sum minimization problem is to identify the trellis code with the RED^2 values that

$$\text{Minimize} \qquad \sum_j \mathrm{MI}_j \tag{2.7}$$

$$\text{Subject to} \qquad \mathrm{MI}_j + \frac{1}{p} \sum_{i:\tilde{a}_i^{(j)} \neq 0} \log_2 \left(\frac{\mathrm{RED}_j^2}{|\tilde{a}_i^{(j)}|^2} \right) \geq \frac{p - q_j}{p} \log_2 \lambda \, , \tag{2.8}$$

where q_j is the number of elements of $\tilde{a}^{(j)}$ equal to zero. The optimal trellis code in this sense is the one that maximizes $J_{\mathrm{MI}} = \sum_j \frac{p-q_j}{p} \log_2(\mathrm{RED}_j^2)$.

2.3 Practical Limitations

Before searching for a code to maximize J_{dB} or J_{MI}, attention must be given to how trellis code constraint length and constellation size limit their ability to achieve rate R for some $\tilde{a}$ vectors that satisfy (2.1) with a large margin. This section demonstrates how limited code constraint length and constellation size can preclude reliable communication even for $\tilde{a}$'s with large mutual information margin.

Reliable communication with a trellis code over a periodic Gaussian channel with $\tilde{a}$ is possible only if $\mathrm{RED}(\tilde{a}) > 0$. Define the periodic effective code length (PECL) to be the smallest number of nonzero positions in the periodic distance vector $\tilde{d}$ of a valid error event of the trellis code. Whenever PECL $< p$, one can easily choose $\tilde{a}$ to force $\mathrm{RED}(\tilde{a}) = 0$. Specifically, select a target error event with exactly PECL nonzero positions in $\tilde{d}^2$ and set $\tilde{a}_i = 0$ whenever $\tilde{d}_i^2 > 0$. Note that the remaining $p-$ PECL values $\tilde{a}_i$ may be made arbitrarily large, easily satisfying (2.1).

Thus a large value of PECL (preferably PECl= p) is desirable for robust performance in periodic Gaussian channels. We now give an equivalent definition of PECL in terms of erasure patterns. Define a $p-q$ erasure pattern to have $\tilde{a}_i \in \{0,1\}$ with q of the p $\tilde{a}_i$ values equal to zero. The periodic effective code length (PECL) is the smallest value of q such that $\text{RED}(\tilde{a}) = 0$ for some $p-q$ erasure pattern $\tilde{a}$.

Trellis code complexity in terms of the number of encoder memory elements limits the value of PECL and thus limits the channels that can be supported to a subset of the channels that satisfy (2.1). Considering all convolutional encoders with ν memory elements, canonical realizations of minimal feedforward encoders achieve all possible sets of output codewords [15]. Thus, the largest value of PECL possible for a given ν is achieved with such an encoder. In a canonical feedforward encoder, the shift register with the least memory elements has at most $\lfloor \nu/k \rfloor$ memory elements. For a Hamming-weight-1 input sequence with the nonzero bit entering a shift register with $\lfloor \nu/k \rfloor$ memory elements, the output codeword has at most $\lfloor \nu/k \rfloor + 1$ nonzero symbols. Thus, any rate-k/n convolutional code with ν memory elements has

$$\text{PECL} \leq \lfloor \nu/k \rfloor + 1 \,. \tag{2.9}$$

To see the limitation imposed by finite constellation size, consider how a finite constellation affects (2.1). Using a constellation $\mathcal{X}$ with 2^n points restricts (2.1) to

$$\frac{1}{p}\sum_{i=1}^{p} I_i^{(\mathcal{X})}(x;y) > R \tag{2.10}$$

where $I_i^{(\mathcal{X})}(x;y)$ is the mutual information between input x drawn uniformly from $\mathcal{X}$ and output $y = \tilde{a}_i x + z$. Ungerboeck computed these mutual informations for several constellations of interest in [16]. A simple upper bound captures the fundamental limitation of a 2^n-point constellation $\mathcal{X}$:

$$I_i^{(\mathcal{X})}(x;y) \leq \min\left\{n, \log\left(1 + \frac{|\tilde{a}_i|^2 \mathcal{E}_x}{N_0}\right)\right\}. \tag{2.11}$$

Thus, when $\tilde{a}_i$ is large the finite constellation size severely limits the amount mutual information provided by the ith symbol in each period. This limitation prevents a trellis code from achieving reliable communication for many channels where (2.1) is satisfied with substantial margin even when ν is large.

This limitation may also be seen through its effect on PECL. For $\tilde{a}_i \in \{0, 1\}$, Using the high SNR values $I_i^{(\mathcal{X})}(x;y) = \tilde{a}_i n$ in (2.10) to compare per-period rate to per-period mutual information demonstrates that rate-k/n trellis codes ($R = k$ and a 2^n-point constellation) cannot provide reliable communication for any $p - q$ erasure pattern where $kp > n(p - q)$, i.e.

$$\text{PECL} \leq \left\lfloor \frac{(n-k)p}{n} \right\rfloor + 1. \tag{2.12}$$

This bound on PECL was first observed in [3]. Later, Knopp et al. [11] arrived at a similar expression via the Singleton bound. As a rule, (2.9) and (2.12) are excellent guides for the choice of ν and n. However, as n is increased beyond the value of $k+1$ suggested by Ungerboeck [16], performance on the standard AWGN channel will actually degrade for fixed ν. Thus there is a conflict between the desire for a larger n to handle more erasures and a smaller n to perform well when there are no erasures.

2.4 Catastrophic Behavior

A final consideration in practical trellis code design for periodic erasures is catastrophic behavior. Catastrophic behavior occurs when an infinite number of bit errors result from a finite Euclidean distance error event. Any encoder that is minimal or systematic (or both) avoids catastrophic behavior [15] for the standard AWGN channel. However, periodic erasure patterns can cause catastrophic behavior even when the encoder is minimal and systematic. The periodic erasures (viewed as puncturing) effectively produce a new higher rate encoder that may be neither systematic nor minimal. It is this punctured encoder that displays the catastrophic behavior. Before maximizing either J_{dB} or J_{MI} of Section 2.2, trellis encoders that become catastrophic for any $\tilde{a} \in \mathcal{C}$ must be discarded [19].

2.4.1 Determining catastrophic behavior

The algorithms for determining catastrophic behavior induced by periodic erasures are closely related to the algorithms used to compute PECL. One approach for computing PECL and determining catastrophic behavior uses multiple trellis codes [14]. Expand any feedforward rate-k/n trellis code to a strictly equivalent rate-kp/np multiple trellis code and remove the q submatrices of size $kp \times n$ that correspond to zero values (erased symbols) in $\tilde{a}$. Define the resulting $kp \times n(p-q)$ matrix of polynomials to be the code C'. This code may be used to determine whether $\langle \tilde{a}^2, \tilde{d}^2 \rangle = 0$ for some $\tilde{d}^2$ (the fundamental step in computing PECL) and to determine whether the erasure pattern induces catastrophic behavior. If C' has zero free Hamming distance then for the original code $\langle \tilde{a}^2, \tilde{d}^2 \rangle = 0$ for some $\tilde{d}^2$. Otherwise, $\langle \tilde{a}^2, \tilde{d}^2 \rangle \neq 0$ for all $\tilde{d}^2$. The original code is noncatastrophic for erasure pattern $\tilde{a}$ iff C' is noncatastrophic, having a power of D as the greatest common divisor of all $kp \times kp$ submatrices.

An adaptation of Lapidoth's algorithm for computing BER in noiseless periodic erasure channels, Theorem 1 in [17], provides another efficient algorithm for determining whether $\langle \tilde{a}^2, \tilde{d}^2 \rangle = 0$. This algorithm may be extended to check for catastrophic behavior. Following [17], consider two matrices, A_0 and A_1, corresponding respectively to the state transition diagram when $\tilde{a}_i = 0$ (erased symbol) and $\tilde{a}_i = 1$ (no erasure). These matrices have only zeros and ones as entries. For both matrices, $A(s_1, s_2) = 1$ if and only if a transition from s_1 to s_2 is possible and cannot be distinguished from the zero-state self loop by the (possibly erased) encoder output. For an erased symbol, $A_0(s_1, s_2) = 1$ if there is a branch connecting state s_1 to state s_2 in a single step. Otherwise, $A_0(s_1, s_2) = 0$. For an unerased symbol, $A_1(s_1, s_2) = 1$ if there is a branch connecting state s_1 to state s_2 in a single step and the convolutional encoder output for this branch is all zeros. Otherwise, $A_1(s_1, s_2) = 0$.

To consider the state transitions possible in p steps corresponding to one period of the erasure pattern, let $A_{\tilde{a}}$ be the product $A_{\tilde{a}_p} A_{\tilde{a}_{p-1}} \dots A_{\tilde{a}_1}$. Define S_m to be the 2^ν-element row vector of integers where the $j+1$ entry of S_m is the number of distinct zero-Hamming-distance paths from the zero state to state j after mp state transitions. S_m can be computed through the recursion $S_{m+1} = S_m A_{\tilde{a}}$ with $S_0 = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \end{bmatrix}$. Note that $\langle \tilde{a}^2, \tilde{d}^2 \rangle \neq 0$ if and only if the first entry of S_m is 1 for all m, i.e. there is no nontrivial zero Hamming distance path from the zero state to itself.

Let $|S_m|$ be the number of nonzero elements in S_m. If $|S_{m+1}| = |S_m|$ no more iterations are necessary. Also, computation may cease with the declaration that $\langle \tilde{a}^2, \tilde{d}^2 \rangle = 0$ whenever the first entry of S_m is greater than 1. Following an argument in the appendix of [17], the linearity of convolutional codes ensures that $|S_m|$ is always a power of 2. Thus the condition $|S_{m+1}| = |S_m|$ will hold for $m > \nu + 1$. Cyclic shifts of $\tilde{a}$ don't affect the outcome of this computation since they don't affect whether $\langle \tilde{a}^2, \tilde{d}^2 \rangle = 0$ for some $\tilde{d}^2$.

To determine catastrophic behavior define S_0^j to be the 2^ν-element row vector that is all zeros except a one in the jth position, and let $S_{m+1}^j = S_m^j A_{\tilde{a}}$. A feedforward encoder is noncatastrophic under the erasure pattern $\tilde{a}$ if and only if the jth entry of S_m^j is zero for all $m > 0$ and all $j \in \{1, 2^\nu - 1\}$. Feedback encoders require an additional step. The zero output Hamming weight loops must be checked to ascertain whether one of them has an associated nonzero input Hamming weight. The existence of a nonzero input Hamming weight loop with zero output Hamming weight indicates catastrophic behavior.

2.4.2 Superiority of recursive encoders

For a feedforward encoder, the periodically punctured code is catastrophic if and only if there is a zero output Hamming weight path from any nonzero state to itself. Such a loop must have nonzero input Hamming weight in a feedforward implementation. This yields the following observation relating catastrophic behavior in feedforward and feedback encoders.

If a feedback convolutional encoder is catastrophic under a fixed periodic symbol erasure pattern, every range-equivalent feedforward convolutional encoder will also be catastrophic for that pattern. Range equivalence ensures that the feedforward encoder has a zero output Hamming weight path for this pattern. The feedforward structure ensures that this path has nonzero input Hamming weight. Thus the feedforward encoder will be catastrophic under this erasure pattern.

A straightforward corollary states that converting an encoder from feedforward to feedback won't introduce new catastrophic behavior. Such a conversion may actually prevent catastrophic behavior. As examples consider the period-8, rate-1/4, 16-QAM feedforward trellis codes listed in Table I of [18]. All these codes are catastrophic for at least one 8-6 erasure pattern. Encoder #7 in Table I of [18] is catastrophic for erasure patterns $\tilde{a}$=[10100000] and $\tilde{a}$=[10001000]. Converting to the range-equivalent recursive encoder with state-space description

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad \mathbf{C} = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix} \quad \mathbf{D} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

provides noncatastrophic behavior for every 8-6 erasure pattern. Here $\mathbf{s}(i+1) = \mathbf{s}(i)\mathbf{A} + \mathbf{u}(i)\mathbf{B}$ and $\mathbf{x}(i) = \mathbf{s}(i)\mathbf{C} + \mathbf{u}(i)\mathbf{D}$ with $\mathbf{u}$ as the k-bit input vector, $\mathbf{x}$ the n-bit output vector, and $\mathbf{s}$ the state vector.

However, conversion to a recursive encoder does not always remove the catastrophic behavior. Conversion of code #8 in Table I of [18] produces

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad \mathbf{C} = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \quad \mathbf{D} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

which has the same catastrophic behavior as the feedforward encoder, catastrophic for $\tilde{a} = [10001000]$.

2.5 Example Codes

This section presents thirteen codes for periodic erasure channels resulting from searches to maximize J_{dB}, J_{MI}, or (for comparison) Euclidean d_{free}. The searches used profile optimal constellation labelings [20]. Table 2.1 provides k, n, and ν for each rate-k/n, 2^{ν}-state convolutional code. This table also gives the polynomial, constellation, and the maximized objective function for each code.

Tables 2.2-2.4 organize the codes by erasure pattern period, presenting RED($\tilde{a}$) and $N_b(\tilde{a})$ for each erasure pattern. These tables also provide the decoding depth $L_D(\tilde{a})$, computed by extending [22], at which the Viterbi algorithm will provide RED($\tilde{a}$) with coefficient $N_b(\tilde{a})$. Figure 2.3 shows the excess mutual information required by each code on each erasure pattern as a function of complexity measured as ν/k.

Codes #1, #2, and #3 compare J_{MI}, J_{dB}, and Euclidean d_{free} as objective functions for identifying rate-1/3, 64-state trellis codes for $p = 2$. Codes #1 and #2, maximizing J_{MI} and J_{dB} respectively, provide similar performance. Code #1 sacrifices a small amount of performance relative to #2 when half the symbols are erased for a comparable gain under the no erasures case.

Code #3 is the maximum free Euclidean distance $\nu = 6$, rate-1/3, 8-PSK trellis code identified in [3]. Code #3's larger free Euclidean distance is sabotaged by its larger value of N_b, and code #3 actually requires slightly more mutual information for BER 10^{-5} with no erasures ($q = 0$) than comparable codes #1, #6, and #12. Code #3 happens to be noncatastrophic for $q = 1$ erasure patterns. However, its performance under this erasure pattern is significantly worse than codes #1 and #2. Notice that in this example the excellent erasure performance comes with no real cost in AWGN performance, given the $\nu = 6$, rate-1/3, 8-PSK structure.

#	p	Objective Function	k	n	ν	Constellation	Poly. ($43 \Longrightarrow 1 + D + D^5$)
1	2	J_{MI}	1	3	6	8-PSK	173 62 115
2	2	J_{dB}	1	3	6	8-PSK	142 75 127
3	2	Euclidean d_{free}	1	3	6	8-PSK	107 32 127
4	3	J_{dB}	1	3	3	8-PSK	13 15 7
5	3	J_{dB}	1	3	4	8-PSK	35 16 21
6	3	J_{dB}	1	3	6	8-PSK	113 172 67
7	3	J_{dB}	1	4	3	16-QAM	15 17 6 5
8	3	J_{dB}	1	4	4	16-QAM	27 35 31 17
9	3	J_{dB}	1	4	5	16-QAM	45 63 57 11
10	3	J_{MI}	1	4	5	16-QAM	57 71 65 53
11	3	Euclidean d_{free}	1	4	6	16-QAM	117 155 145 137
12	5	J_{MI} and J_{dB}	1	3	6	8-PSK	171 46 133
13	5	Progressive $\tilde{a}$'s	1	3	6	8-PSK	173 62 115

Table 2.1: These thirteen codes resulted from searches that maximized the indicated objective functions. The least significant bit of the constellation label is produced by the leftmost polynomial with significance increasing moving left to right. The 8-PSK labeling is 0, 2, 3, 1, 5, 7, 6, 4 in octal going around the circle. The 16-QAM labeling in raster order using hexadecimal is 3, 1, 5, 7; 2, 0, 4, 6; a, 8, c, e; b, 9, d, f.

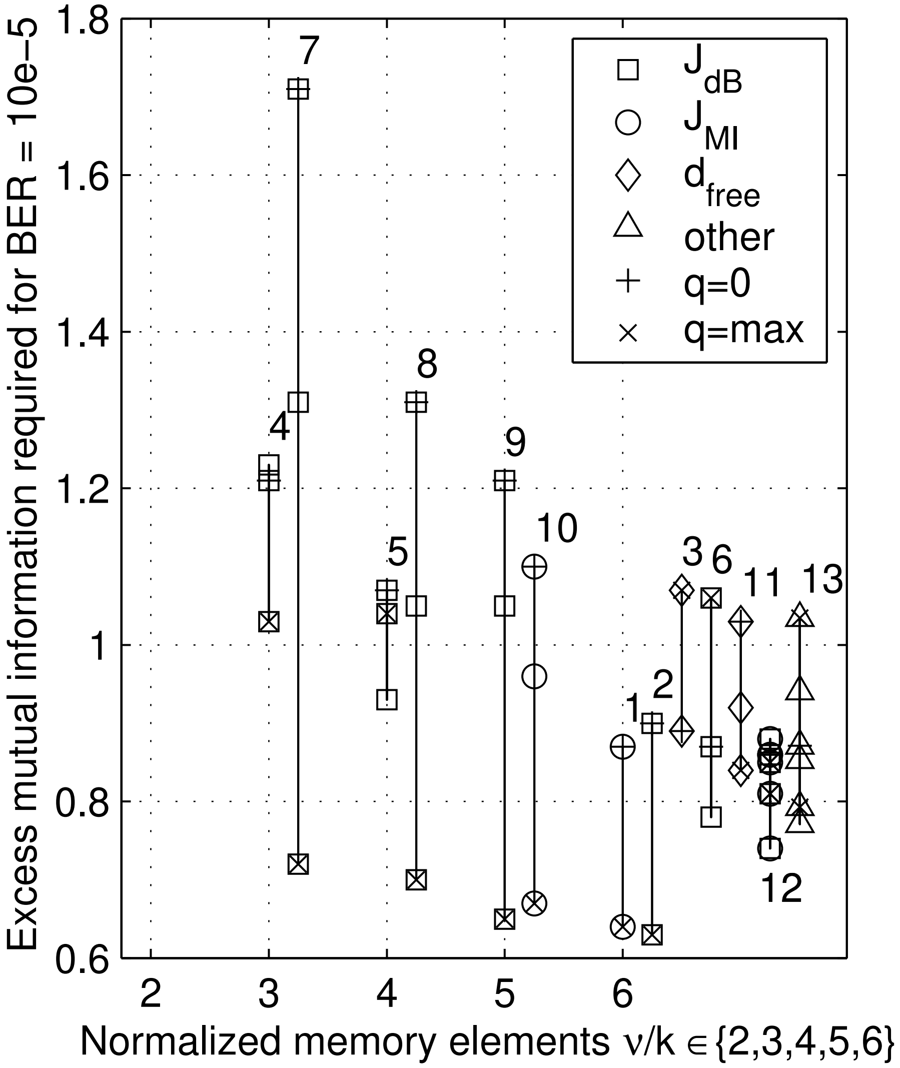


Figure 2.3: Excess mutual information vs. ν/k for each of the thirteen codes described in Tables I-IV. Each code is shown as a line segment connecting symbols that indicate the excess mutual information required by each erasure pattern to achieve a BER if 10^{-5}. The ν/k for each code is the nearest integer $\in \{2, 3, 4, 5, 6\}$ to the left of the code's line segment.

$\tilde{a}$					11			01		
#	J	k	n	ν	RED	N_b	L_D	RED	N_b	L_D
1	J_{MI}	1	3	6	4.21	0.7500	31	2.27	3.6250	38
2	J_{dB}	1	3	6	4.07	1.0000	25	2.40	10.7500	45
3	d_{free}	1	3	6	4.24	1.0000	29	1.53	1.7500	24

Table 2.2: Trellis codes for period-two erasure patterns.

However, Code #3 serves to introduce the conflict between the desire for a large constellation to optimize good erasure performance and the desire for a small constellation to optimize free Euclidean distance. The maximum Euclidean distance of 4.24 possible with a rate-1/3, 8-PSK, 64-state trellis code is less than the distance of $\sqrt{20} = 4.47$ that is possible with a rate-1/2, 4-PSK, 64-state trellis code (i.e. 133 171 and Gray-coded 4-PSK). This 4-PSK code does perform slightly better than #1 and #2 in AWGN, (requiring a MI of only 0.88 for BER 10^{-5}), but no rate-1/2 code can provide coding gain under $2-1$ erasure patterns since they have no redundancy after the erasures. We will explore this conflict in more detail below.

To study the performance limitation of constellations that allow no redundancy after certain erasure patterns, codes #4, #5, and #6 are rate-1/3, 8-PSK trellis codes with 8, 16, and 64 states respectively designed for erasure patterns with $p = 3$. J_{dB} was used as the design metric since it places more emphasis on the maximum erasure cases. No 32-state code was found that provided an improvement in J_{dB} over the 16-state code. With $q = 2$ the redundancy is reduced to zero. No amount of complexity can provide much improvement over the performance of uncoded 8-PSK for this erasure pattern. In fact, added complexity can even increase BER by associating more bit errors with the least Euclidean

$\tilde{a}$					111			011			001		
#	J	k	n	ν	RED	N_b	L_D	RED	N_b	L_D	RED	N_b	L_D
4	J_{dB}	1	3	3	3.25	1.5000	18	2.08	0.5833	17	0.77	2.8333	9
5	J_{dB}	1	3	4	3.55	1.0000	17	2.52	0.5000	21	0.77	2.6667	12
6	J_{dB}	1	3	6	4.00	0.5000	23	2.83	0.1667	36	0.77	3.8333	18
7	J_{dB}	1	4	3	2.61	0.1875	10	2.00	1.3333	12	1.10	3.8750	12
8	J_{dB}	1	4	4	3.10	1.0000	13	2.19	0.2917	15	1.10	2.5833	18
9	J_{dB}	1	4	5	2.97	0.1250	12	2.19	0.4167	17	1.26	2.2500	27
10	J_{MI}	1	4	5	3.46	0.5000	19	2.37	0.1667	20	1.10	1.1667	21
11	d_{free}	1	4	6	3.79	1.0000	26	2.45	0.6250	24	0.89	0.6667	18

Table 2.3: Trellis codes for period-three erasure patterns.

$\tilde{a}$		11111			01111			00111			01011			00011			00101		
#	k	RED	N_b	L_D	RED	N_b	L_D	RED	N_b	L_D	RED	N_b	L_D	RED	N_b	L_D	RED	N_b	L_D
12	1	4.11	0.1562	35	3.03	0.6000	33	2.08	0.0625	29	2.40	0.5000	33	1.33	3.0750	44	1.33	0.9500	40
13	1	4.21	0.7500	31	3.22	0.8000	33	2.08	0.8000	34	2.40	0.9000	35	1.08	1.8000	30	1.33	1.0000	55

Table 2.4: Trellis codes for period-five erasure patterns. All codes have $\nu = 6$ and $n = k + 2$.

distance error events. In Table 2.3 codes #4, #5, and #6 have RED increasing with complexity for $q = 0$ and $q = 1$, but RED stays constant for $q = 2$. Also for $q = 2$, N_b is similar for $\nu = 3$ and $\nu = 4$ but *increases* by 35% for $\nu = 6$.

In contrast to #4, #5, and #6, codes #7, #8, and #9 show how increasing the constellation size from 8-PSK to 16-QAM allows redundancy to remain for the $q = 2$ erasure pattern. However, the gains for $q = 2$ come at the cost of significant performance loss for the standard AWGN channel $q = 0$. Unlike $q = 2$ with 8-PSK, the $q = 0$ performance with 16-QAM does improve with increasing complexity.

Code #10 shows that using J_{MI} rather than J_{dB} can improve AWGN performance by placing more weight on the $q = 0$ erasure pattern. The relationship between #9 and #10 is similar to the relationship between #2 and #1. An exhaustive search over 64-state, rate-1/4 trellis codes to maximize either J_{MI} or J_{dB} is computationally too intensive at present. Instead we analyzed code #11, the rate-1/4, 64-state maximum free Euclidean distance code in [3]. This trellis code happens to have relatively good performance although it suffers a performance loss for $q = 2$ relative to the 32-state codes #9 and #10.

Together, the codes in Table 2.3 illustrate the difficulty of selecting the appropriate constellation size. For rate-1/3, 8-PSK a large excess mutual information was required for $q = 2$ because this erasure pattern produced uncoded 8-PSK performance regardless of complexity. On the other hand, the largest Euclidean distance rate-1/4, 16-QAM, 64-state trellis code requires a similarly large excess mutual information when $q = 0$.

The dilemma in constellation size selection is avoided in the design of code #12, a rate-1/3, 8-PSK trellis code for $p = 5$ that maximizes both J_{MI} and J_{dB}. Here, no erasure pattern reduces the redundancy to exactly zero. The code can not operate for the negative redundancy scenarios of $q = 4$ or $q = 5$, but has a consistent excess mutual information between 0.74 and 0.88 for the those erasure patterns that preserve positive redundancy. Note that despite its smaller Euclidean distance of 4.11, code #12 has superior BER performance to the distance 4.47 4-PSK code discussed earlier because of its much smaller N_b value (.1562 vs. 36). Code #13 is inferior to #12 both in terms of J_{dB} and J_{MI} computed over the six erasure patterns of interest. However, this Pareto optimal code is superior to #12 in both these metrics when they are computed over the progressive-puncturing family $\{11111, 01111, 01011, 01010\}$.

2.6 Applications

These trellis codes are suitable for periodic erasures. They can also be used as a family of rate-compatible, unequal error protection trellis codes over AWGN channel by periodic symbol-puncturing. In other words, we can periodically puncture symbols of such a trellis code to get a family of different rates and unequal error protection. We apply these trellis codes to adaptive multi-rate speech transmission.

Figure 2.4 gives an example to illustrate unequal level of error protection of a rate-1/3, 16-state, 8-PSK, period = 8, generator polynomial G = [32 11 27] (in octal form) trellis code by puncturing 4, 3, or 2 symbols per period (the puncturing patters are 01010101, 01010111 and 01110111 respectively, 0 means puncturing) over AWGN channel with SNR = 6dB. The traceback depth used throughout the frame is L = 48. Each level of protection is 96 bits long. The delay caused by traceback is reflected in the transition zones.

The following application is to apply the rate-compatible punctured trellis codes (RCPT) to an embedded source coder: a perceptual subband coder (SBC) [25], [26]. This embedded joint source-channel coding scheme of speech is shown in Figure 2.5.

The goal of joint source-channel coding scheme of speech is to design a system that provides best possible speech quality throughout a wide range of channel conditions. Figure 2.5 shows one solution which allows the transceivers to monitor the state of the communication channel and to dynamically allocate the bit-stream between source and channel coding according. For low SNR channels, the source

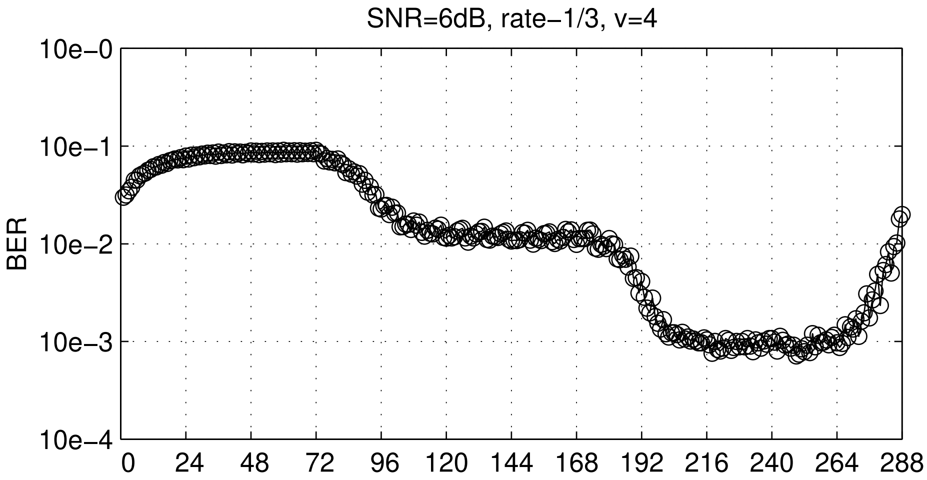


Figure 2.4: Bit error rates for each of the 315 bit positions in a frame with varying puncturing severity. There are three protection levels with puncturing patters 01010101, 01010111 and 01110111 respectively, 0 means puncturing. Each level of protection is 96 bits long. The traceback depth used throughout the frame is L = 48.

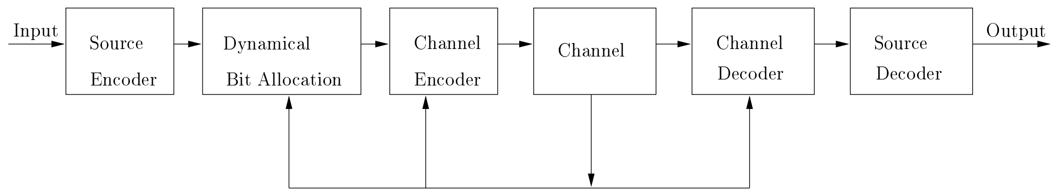


Figure 2.5: Joint source-channel coding scheme.

coder operates at a low bit rate, thus allowing powerful forward error control. For high SNR channels, the source coder uses its highest rate resulting in high speech quality.

To show the advantage of our RCPT code in this joint source-channel coding scheme of speech, we also apply rate-compatible punctured convolutional codes (RCPC) [28] and RCPC-bit interleaved coded modulation codes (RCPC-BICM) to this joint system. Table 2.5 gives the characteristics of RCPT, RCPC-BICM and RCPC over AWGN channel.

The simulation results of the embedded joint source-channel coding scheme are shown in Figure 2.6 [27]. Compared with using RCPC, the performance of using RCPT is better at high SNR and less complicated, since puncturing symbols is simpler than puncturing bits as well as having a smaller punctured traceback depth. Compared with using RCPC-BICM, using RCPT performs better at low SNR.

2.7 Summary

The compound channel coding theorems of Root and Varaiya show that a single code can support a given rate over all periodic Gaussian channels with sufficient mutual information. This chapter explores how this theoretical promise may be

RCPT Rate-1/3, 8-PSK G=[165 142 127], p=9				RCPC-BICM Rate-1/3, 8-PSK G=[155 127 117], p=3*9 bits				RCPC Rate-1/2, 4-PSK G=[133 171], p=2*8 bits			
Rate	**punct-Vector**	RED^2	L_D	**Rate**	**punct-Vector**	RED^2	L_D	**Rate**	**punct-Vector**	RED^2	L_D
1.00	111111111	16	24	1.00	111111111 111111111 111111111	8.79	21	1.00	11111111 11111111	20	27
1.12	111111110	11.2	25	1.12	111111111 111101100 111111111	7.03	21	1.14	11111111 11101110	14	27
1.29	111101110	9.17	26	1.29	111111111 100100100 111111111	5.86	21	1.33	11111111 10101010	12	34
1.50	110101110	6.93	31	1.50	111111111 000000000 111111111	5.86	28	1.60	11111111 10001000	8	48
1.80	110101100	4.34	31	1.80	110110110 000000000 111111111	4.10	34	1.78	11110111 10001000	6	92
2.25	100101100	1.76	27	2.25 -----	100010100 000000000 111111111 Catastrophic	2.42	41	-----	Catastrophic		
3.00	100101000	0.59	36	3.00	uncoded 8-PSK			2.00	uncoded 4-PSK		

Table 2.5: Characteristics of RCPT, RCPC-BICM and RCPC over AWGN Channel.

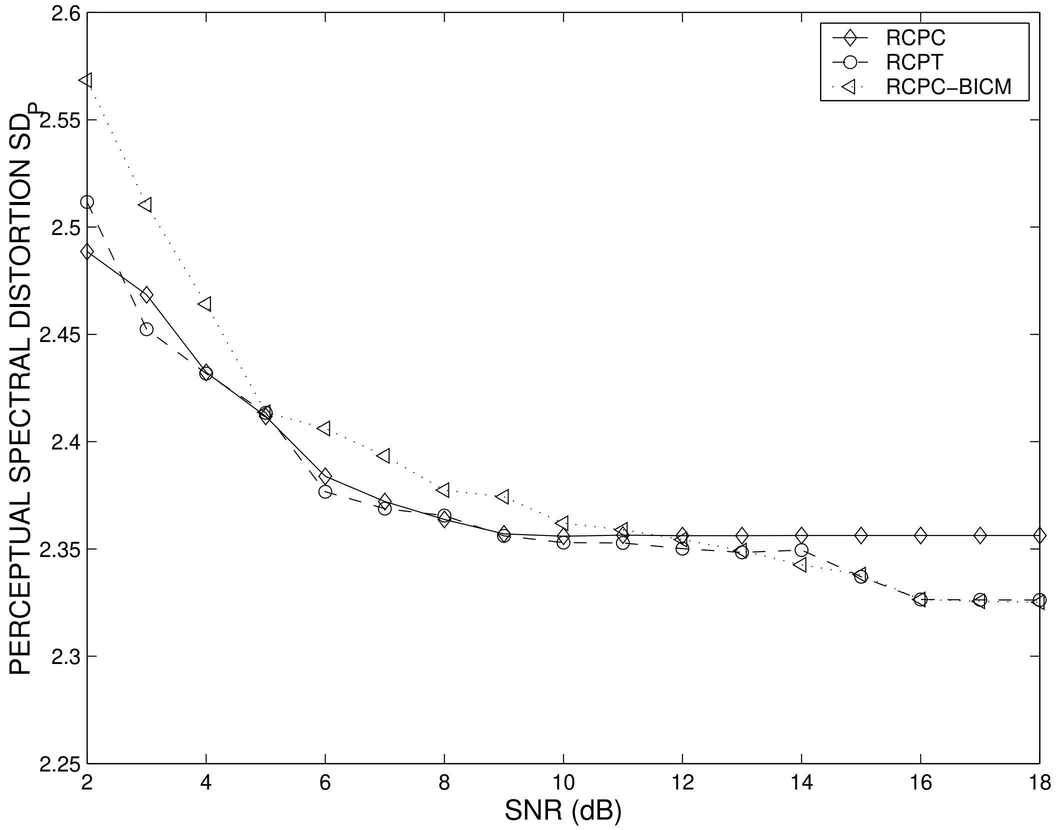


Figure 2.6: Comparison of joint source-channel encoding with RCPT, RCPC-BICM and RCPC over AWGN channel.

approached by trellis codes designed to support a given rate over a specified family of periodic erasure patterns. One benefit of the compound channel perspective is the realization that BER performance should be plotted against mutual information to compare performance of many channels simultaneously.

Designing a trellis code to perform well for all the channels in a family of possible channels is a multi-criterion optimization problem with potentially many Pareto optimal codes, each best for some, but not all, of the channels. Objective functions that minimize the sum of the required margin, either in SNR or in MI, allow one code to be selected from among the many Pareto optimal alternatives. Simple bounds on PECL demonstrate that limitations on the number of memory elements and the constellation size can prevent trellis codes from working on a channels for which reliable communication is theoretically possible.

When a periodic erasure pattern removes all redundancy, BER performance similar to uncoded transmission using the same constellation is the best that can be expected. Despite this obvious limitation, it is surprising how well a trellis code can perform under severe erasure as long as some redundancy remains. Perhaps it is also surprising that increasing the constellation size beyond $k+1$ tends to degrade performance on the AWGN channel with no erasures for a given number of states. There is a difficult trade-off between a constellation size large enough for the most severe erasure patterns of interest and a constellation size small enough to maintain good performance when no erasures are present.

This difficult trade-off might be partially mitigated by using non-uniform constellations. Furthermore, the problem might be entirely avoidable in situations where sufficient latency is available for the use of concatenated codes with iterative decoders. Such codes naturally have large block-lengths and are likely to perform well on channels with no erasures even when n is large compared to $k+1$.

In general, the extension of this work to concatenated codes promises to bring performance much closer to that promised by Root and Varaiya.

This chapter restricts attention to rate-$1/n$ trellis codes combined with 2^n-point constellations. There are natural extensions of this work to multidimensional trellis codes or higher dimension constellations as proposed by Caire, Leonardini, and Viterbo [24].

For the simulations in this chapter we have assumed that the receiver has perfect knowledge of $\tilde{a}$, $\mathcal{E}_x$, and N_0. However, the results of Root and Varaiya do not require this knowledge. Another interesting area of future work analyzes how much performance of such trellis codes is affected by imperfect channel estimates.

Those trellis codes designed specifically for periodic erasures can also be used as families of rate-compatible, unequal error protection trellis codes by periodic symbol-puncturing. We applied such codes to the embedded joint source-channel coding scheme of speech. In comparison with RCPC and RCPC-BICM, our rate-compatible punctured trellis codes perform best over the entire range of channel conditions.

Chapter 3

Cross Constellation Labeling

The design of a trellis code involves both the selection of a convolutional code and the selection of a labeled constellation (with a given size and shape) to maximize some subset of the metrics presented in [16], [29], [30], [2], [7], [31]. The specified constellation labeling controls the mapping of each output symbol from the binary convolutional encoder to a constellation point in Euclidean space. Typical searches for "optimal" trellis codes fix the constellation labeling and then exhaustively search over a space of binary encoders.

There are two standard constellation labeling paradigms: Gray-code labeling and set-partition labeling. Gray-code labeling was first introduced in the context of pulse code modulation by F. Gray in his 1953 patent [32]. Gray-code labeling minimizes the effect of noise on the bit error rate by having labels between any two nearest neighbors differ by exactly one bit. Set-partition labeling was proposed by Ungerboeck [16] (see also [33]) as part of a technique for designing maximum Euclidean distance trellis encoders that are easily designed and implemented because they utilize parallel transitions in the trellis diagram. Gray-code labelings have an obvious application for uncoded transmission, while set-partition labelings have demonstrated good performance in trellis coded modulation over additive white Gaussian noise channels.

Both Gray-code and set-partition labelings are well-suited to their original applications. However, for other applications, such as trellis coded modulation over channels with fading or puncturing or as constituent codes for parallel concatenation, there is no compelling reason to use either a Gray-code labeling or a set-partition labeling. A new approach to constellation labeling maximizes the constellation's edge profile [20], [34]. This new labeling paradigm produces better trellis codes than the two standard paradigms when the trellis does not have parallel transitions.

With the best edge profile structure as the primary consideration, Wesel [34] identified 4-PSK, 8-PSK, and 16-QAM labeling structures that have the best edge profile. These structures admit either a Gray-code labeling or a set-partition labeling [34]. This chapter examines the best edge profile labeling structures for 8-QAM and quasi-Gray-coded (i. e., a labeling minimizing the number of minimum distance edge labels for which the Gray condition is not satisfied) 32-QAM cross constellations. We also investigate the relationship between best edge profile labeling structures and set-partition labeling structures. For 8-QAM and 32-QAM, Gray-code labelings are impossible [34].

Section 3.1 presents the edge-profile-maximizing labeling structures for the 8-QAM constellation. Section 3.1 also demonstrates that any set-partition labeling of 8-QAM maximizes the edge profile. Section 3.2 explores the edge profiles of various quasi-Gray-coded 32-QAM labelings. Six distinct edge profile upper bounds of quasi-Gray-coded 32-QAM are found and all are shown to be achievable. One of the six distinct quasi-Gray-coded 32-QAM labelings dominates other five in terms of edge profile. No set-partition labeling of 32-QAM can achieve this best edge profile. Section 3.3 focuses on applications of the best edge profile labelings of cross constellations. A code design example is given to illustrate ramifications of the choice of a constellation labeling. Section 3.4 gives conclusions with some open questions. Appendix A contains the proofs of some lemmas used in this chapter.

3.1 Superior Edge Profile 8-QAM Constellation Labelings

Before examining the best edge profile 8-QAM constellation labelings, we review some definitions and several basic labeling lemmas [34].

3.1.1 Definition of the edge profile and labeling lemmas

A constellation is a fully connected graph whose points have distinct **point labels**. Any line which connects a pair of constellation points is an **edge**. The binary symbol error that corresponds to an edge (the XOR summation of its pair of point labels) is its **edge label**. For a constellation with 2^n points, each binary symbol error labels 2^{n-1} edges. The edges with a given binary symbol error $\mathbf{v}$ may not all have the same Euclidean distance. In the context of a code search where design metrics are related to Euclidean distance, each binary symbol is often simply assigned its worst-case Euclidean distance (squared) $d^2_{\text{MIN}}(\mathbf{v})$ ([16], [35], [36]). This approach can be pessimistic, but often it is not. The worst-case Euclidean distance sequences are guaranteed to exist for many common code rates and constellation sizes [16].

Now we list definitions and basic lemmas on which the following discussion are based.

Definition 1 *The* **edge length profile** $\{p_i\}$ *of a labeled constellation is the list of* $d^2_{\text{MIN}}(\mathbf{v})$ *values for all nonzero edge labels* $\mathbf{v}$ *ordered to increase monotonically [4].*

Definition 2 *Let* $p_i^{(1)}$ *and* $p_i^{(2)}$ *be the edge length profiles for constellations* $\mathcal{C}_1$ *and* $\mathcal{C}_2$ *respectively. Labeled constellation* $\mathcal{C}_1$ *is* **equivalent in profile** *to* $\mathcal{C}_2$ *if* $p_i^{(1)} = p_i^{(2)}$ *for* $i = 1, \ldots, (|\mathcal{C}_1| - 1)$*. Labeled constellation* $\mathcal{C}_1$ *is* **superior in profile** *to* $\mathcal{C}_2$ *if* $p_i^{(1)} \geq p_i^{(2)}$ *for* $i = 1, \ldots, (|\mathcal{C}_1| - 1)$ *with at least one strict inequality [4].*

Lemma 1 *All the edges emanating from a given point must have different edge labels [4].*

Lemma 2 *Two different paths emanating from the same initial constellation point have the same edge label sum if and only if they end at the same final constellation point (i.e. the two paths form a cycle). In particular, any path that is not a cycle must have an edge label sum that is nonzero [4].*

The following three lemmas assume that all the points in the constellation are connected by paths consisting only of the minimum-length edges.

Lemma 3 *The set of edge labels of the minimum length edges must contain a basis for the set of all edge labels [4].*

Lemma 4 *If there are no more than* n *linearly independent minimum distance edge labels, then these edge labels can connect at most* $2^n - 1$ *distinct nonzero edges starting from a point, i. e., at most* 2^n *distinct points [21].*

Lemma 5 *A constellation (or subconstellation) with more than* $2^{n-1} - 1$ *distinct nonzero edges or* 2^{n-1} *distinct points must have at least* n *linearly independent minimum distance edge labels.*

3.1.2 Superior edge profile of 8-QAM

For 8-QAM and 32-QAM cross constellations, since Gray-code labelings are impossible [34], there must be at least four distinct minimum distance edge labels in 8-QAM and at least six distinct minimum distance edge labels in 32-QAM. We refer to 8-QAM or 32-QAM constellation labelings with exactly four or six distinct minimum distance edge labels, respectively, as quasi-Gray-coded. For convenience, we denote the minimum distance as one for the rest of the chapter.

Start from a central constellation point and draw an edge to every other constellation point. These distinctly labeled edges (according to Lemma 1) construct an edge profile upper bound for the 8-QAM constellation which is demonstrated in Fig. 3.1. Table 3.1 shows the bounding edge length profile resulting from the edges in Fig. 3.1.

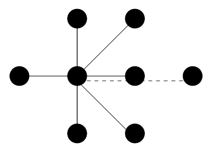


Figure 3.1: Edges for profile upper bound in 8-QAM.

i	1	2	3	4	5	6	7
$d^2_{\text{MIN}}(E_i)$	1	1	1	1	2	2	4

Table 3.1: Edge profile upper bound for 8-QAM.

There are exactly four $d^2_{\text{MIN}} = 1$ edge labels in a labeled constellation that achieves the edge length profile of Table 3.1. Lemma 3 implies that the linearly independent $d^2_{\text{MIN}} = 1$ edge labels must form a basis. The vector space of 3-bit vectors that label the edges of the 8-QAM constellation requires only three vectors to form a basis. The fourth edge label must be a linear combination of the basis edge labels.

Choose $\mathbf{v_1}, \mathbf{v_2}, \mathbf{v_3}$ and $\mathbf{v_4}$ to be a set of four distinct edge labels that span the space of 3-bit edge labels. Choose one of the two central points and label the four minimum d^2_{MIN} edges connected to this point. Step 1 (Figure 3.2(a)) shows a labeling for these four edges. Each of the four edge labels must be different by Lemma 1.

Now consider the edges labeled $\mathbf{a}$ and $\mathbf{b}$ in Step 2 (Figure 3.2(b)). The label $\mathbf{a}$ could be any of the other three minimum $d^2_{\text{MIN}} = 1$ edge labels except $\mathbf{v_3}$ by Lemma 1. Thus there are three possibilities at Step 3 (Fig. 3.2(c)-(e)). Lemma 2 gives

$$\mathbf{v_3} + \mathbf{v_4} = \mathbf{a} + \mathbf{b}. \tag{3.1}$$

Consider the 1st possibility at Step 3 (Fig. 3.2(c)), $\mathbf{a} = \mathbf{v_1}$, then $\mathbf{b} \neq \mathbf{v_1}$ and $\mathbf{b} \neq \mathbf{v_4}$ by Lemma 1. If $\mathbf{b} = \mathbf{v_3}$, then $\mathbf{v_4}$ must equal $\mathbf{v_1}$ to satisfy Equation (3.1). However, $\mathbf{v_4}$ can not equal $\mathbf{v_1}$ since they are two distinct edge labels. Thus $\mathbf{b} \neq \mathbf{v_3}$ and $\mathbf{b}$ must equal $\mathbf{v_2}$. Similar arguments can be applied to show that when $\mathbf{a} = \mathbf{v_2}$, we get $\mathbf{b} = \mathbf{v_1}$ (Fig. 3.2(d)) and when $\mathbf{a} = \mathbf{v_4}$, then $\mathbf{b} = \mathbf{v_3}$ (Fig. 3.2(e)).

In Step 4, consider the labels for edges $\mathbf{c}$, $\mathbf{d}$ and $\mathbf{e}$. There are two possibilities at Step 4 for each of the three possibilities at Step 3. Consider Fig. 3.2(c) as an example to illustrate how to label edges $\mathbf{c}$, $\mathbf{d}$ and $\mathbf{e}$ at Step 4. Since $\mathbf{c} \neq \mathbf{v_1}$ and $\mathbf{c} \neq \mathbf{v_3}$ by Lemma 1 from Step 3, then $\mathbf{c} = \mathbf{v_2}$ in Fig. 3.2(f) or $\mathbf{c} = \mathbf{v_4}$ in Fig. 3.2(g). In Fig. 3.2(f), since $\mathbf{c} = \mathbf{v_2}$, then by Lemma 1, $\mathbf{e} = \mathbf{v_4}$.

Lemma 2 gives

$$\mathbf{v_3} + \mathbf{v_2} = \mathbf{c} + \mathbf{d}. \tag{3.2}$$

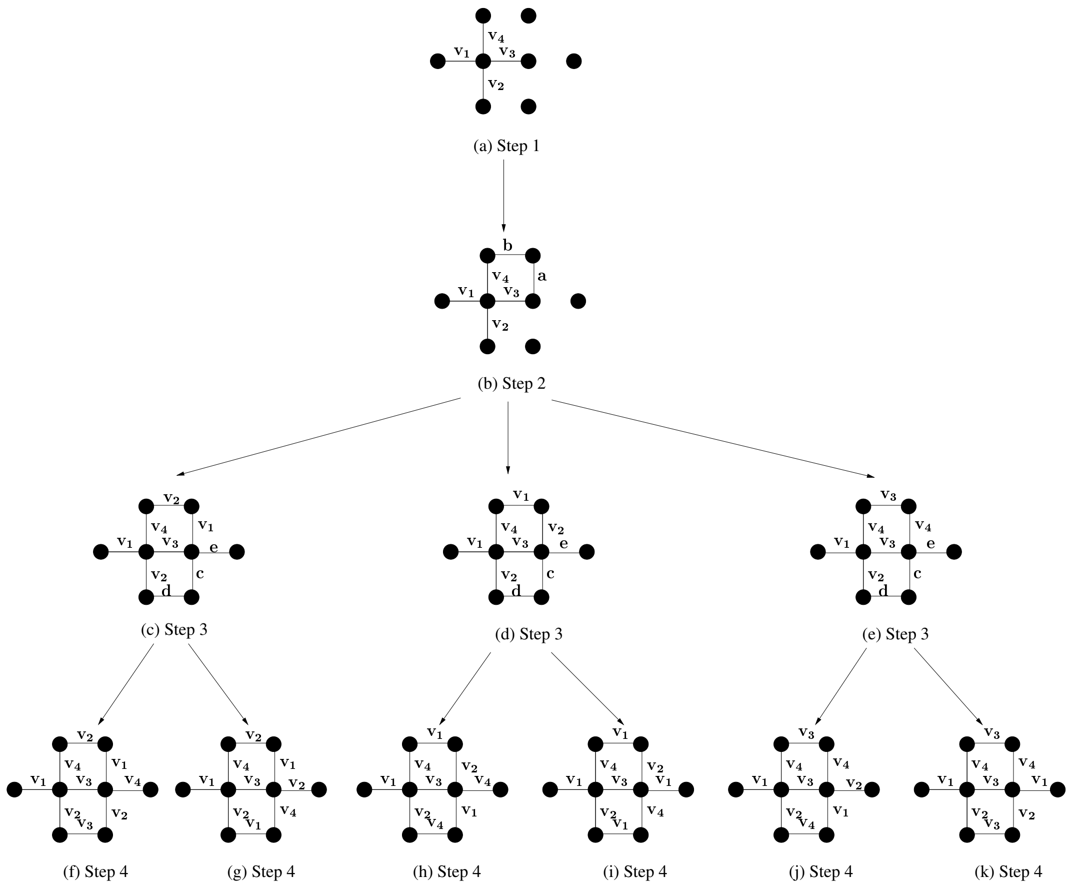


Figure 3.2: Steps for obtaining labeling structures for quasi-Gray-coded 8-QAM constellations.

In Fig. 3.2(f), since $\mathbf{c} = \mathbf{v_2}$, then Equation (3.2) forces $\mathbf{d} = \mathbf{v_3}$. Thus we complete the labelings for 8-QAM in Fig. 3.2(f). Similarly, we can obtain the labelings as in Fig. 3.2(g)-(k) for the 8-QAM constellation. All six labeling structures are under the restriction that there are exactly four distinct minimum distance edge labels. Every structure except that of Fig. 3.2(k) requires $\mathbf{v_4} = \mathbf{v_1} \oplus \mathbf{v_2} \oplus \mathbf{v_3}$ by Lemma 2, permitting any three of these edge labels to form the basis. So Fig. 3.2(f) and Fig. 3.2(j) are isomorphic labeling structures and so are Fig. 3.2(g) and Fig. 3.2(h).

Table 3.2 shows the edge profiles of the six labeling structures for 8-QAM. Only Fig. 3.2(i) and Fig. 3.2(k) achieve the edge profile upper bound given in

Fig. 3.2(f) and Fig. 3.2(j)	1	1	1	1	2	2	2
Fig. 3.2(g) and Fig. 3.2(h)	1	1	1	1	2	2	2
Fig. 3.2(i)	1	1	1	1	2	2	4
Fig. 3.2(k)	1	1	1	1	2	2	4

Table 3.2: Edge profiles of six labeling structures for 8-QAM.

Table 3.2.

The two labeling structures that achieve the best possible edge profile are shown in Figure 3.3. Select any point as the zero point and choose $\mathbf{v_1}$, $\mathbf{v_2}$ and $\mathbf{v_3}$ as linearly independent three-bit labels. In Fig. 3.3(a) $\mathbf{v_4} = \mathbf{v_1} \oplus \mathbf{v_2} \oplus \mathbf{v_3}$ by Lemma 2, and in Fig. 3.3(b) either $\mathbf{v_4} = \mathbf{v_1} \oplus \mathbf{v_2}$ or $\mathbf{v_4} = \mathbf{v_1} \oplus \mathbf{v_2} \oplus \mathbf{v_3}$ (since $\mathbf{v_4} \neq \mathbf{v_1} \oplus \mathbf{v_3}$ and $\mathbf{v_4} \neq \mathbf{v_2} \oplus \mathbf{v_3}$ by Lemma 2). In this manner, we can obtain superior edge profile 8-QAM constellation labelings.

(a) (b)

Figure 3.3: The two labeling structures achieving the superior edge profile for 8-QAM constellations.

3.1.3 Equivalence between superior edge profile labeling and set-partition labeling of 8-QAM

This subsection investigates the structural relationship between superior edge profile labeling and set-partition labeling. We begin with a lemma that characterizes the 8-QAM constellation labeling structure imposed by a set-partitioning.

Lemma 6 *Set-partitioning the 8-QAM constellation causes minimum distance horizontal (or vertical) edge labels to alternate between exactly two different three-bit labels, which differ only in the third (the most significant) bit. Together, the vertical and horizontal minimum distance edge labels consist of four distinct labels, which are selected from {001, 011, 101, 111}. Note that any one of the minimum distance edge labels is the sum of the other three minimum distance edge labels.*

Proof:

By definition, set-partitioning the 8-QAM constellation causes the minimum distance of two points sharing the same m least significant bits ($m \in \{1, 2\}$) to be as large as possible, i. e., $d^2_{\text{MIN}} = 2$ when $m = 1$, $d^2_{\text{MIN}} = 4$ when $m = 2$. This structure and the fact that nearest neighbors separated by minimum distance differ in the least-significant bit produce the constellation labeling shown in Figure 3.4. Every letter represents a bit and $\bar{\mathbf{a}}$, $\bar{\mathbf{b}}$, $\bar{\mathbf{c}}$, $\bar{\mathbf{d}}$, $\bar{\mathbf{e}}$,$\bar{\mathbf{f}}$ and $\bar{\mathbf{g}}$ are complementary to $\mathbf{a}$, $\mathbf{b}$, $\mathbf{c}$, $\mathbf{d}$, $\mathbf{e}$, $\mathbf{f}$ and $\mathbf{g}$ respectively. For example, the points with $\mathbf{c}$ as the least significant bit form a four-point coset. Also by Lemma 1, $\mathbf{v_x}$, $\mathbf{v_z}$, $\mathbf{v_j}$ and $\mathbf{v_k}$ are four distinct

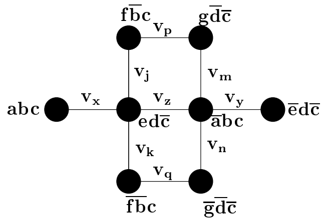


Figure 3.4: Set-partition labeling of 8-QAM.

minimum distance edge labels. The edge labels obey the following equations,

$$\mathbf{v_x} = \mathbf{abc} - \mathbf{ed\bar{c}}\,, \quad \mathbf{v_y} = \mathbf{\bar{a}bc} - \mathbf{\bar{e}d\bar{c}}\,, \quad \mathbf{v_z} = \mathbf{\bar{a}bc} - \mathbf{ed\bar{c}}\,. \tag{3.3}$$

$$\mathbf{v_p} = \mathbf{f\bar{b}c} - \mathbf{g\bar{d}\bar{c}}\,, \quad \mathbf{v_q} = \mathbf{\bar{f}\bar{b}c} - \mathbf{\bar{g}\bar{d}\bar{c}}\,. \tag{3.4}$$

$$\mathbf{v_j} = \mathbf{f\bar{b}c} - \mathbf{ed\bar{c}}\,, \quad \mathbf{v_k} = \mathbf{\bar{f}\bar{b}c} - \mathbf{ed\bar{c}}\,. \tag{3.5}$$

$$\mathbf{v_m} = \mathbf{\bar{a}bc} - \mathbf{g\bar{d}\bar{c}}\,, \quad \mathbf{v_n} = \mathbf{\bar{a}bc} - \mathbf{\bar{g}\bar{d}\bar{c}}\,. \tag{3.6}$$

Since operations are over GF(2), for horizontal edge labels, we obtain

$$\mathbf{v_x} = \mathbf{v_y}\,, \quad \mathbf{v_p} = \mathbf{v_q}\,, \tag{3.7}$$

$\mathbf{v_x}$, $\mathbf{v_y}$, $\mathbf{v_z}$, $\mathbf{v_p}$, $\mathbf{v_q}$ all have the same two least significant bits (LSBs). Furthermore the LSB is 1 for all these labels, and $\mathbf{v_x}$, $\mathbf{v_z}$ differ in the most significant bit (MSB). Since these labels have 3 bits, $\mathbf{v_p}$ and $\mathbf{v_q}$ are both equal to either $\mathbf{v_x}$ or $\mathbf{v_z}$. For the vertical edge labels, $\mathbf{v_j}$, $\mathbf{v_k}$, $\mathbf{v_m}$, $\mathbf{v_n}$ all have the same two least significant bits, their LSB is 1. Also, $\mathbf{v_j}$, $\mathbf{v_k}$ differ in the MSB, and $\mathbf{v_m}$, $\mathbf{v_n}$ differ in the MSB. Since these labels are 3 bits, there are only two distinct edge labels among $\mathbf{v_j}$, $\mathbf{v_k}$, $\mathbf{v_m}$ and $\mathbf{v_n}$. Thus there are a total of only four distinct minimum distance edge labels $\mathbf{v_x}$, $\mathbf{v_z}$, $\mathbf{v_j}$ and $\mathbf{v_k}$, and their LSBs are all 1. Therefore $\mathbf{v_x}, \mathbf{v_z}, \mathbf{v_j}, \mathbf{v_k} \in \{001,$ $011, 101, 111\}$, i. e., exactly one of the minimum distance edge labels is the sum of the other three minimum-distance edge labels. $\square$

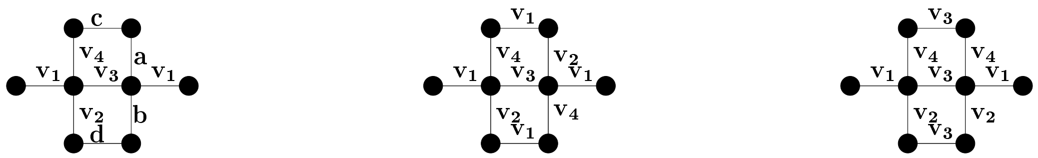
Based on Lemma 6 and the two superior edge profile labeling structures shown in Fig. 3.3, we can obtain Theorem 1 on the structural relationship between superior edge profile labeling and any set-partition labeling for the 8-QAM constellation.

Theorem 1 *For the 8-QAM constellation, any set-partition labeling is a superior edge profile labeling.*

Proof:

Fig. 3.5 shows steps to achieve 8-QAM set-partition labelings. By using the same argument as labeling quasi-Gray-coded 8-QAM at Step 1 and Lemma 6, the set-partition labeling structure of Step 1 is given in Fig. 3.5(a), where $\mathbf{v_1}, \mathbf{v_2}, \mathbf{v_3}$ is a basis and $\mathbf{v_4} = \mathbf{v_1} \oplus \mathbf{v_2} \oplus \mathbf{v_3}$, $\mathbf{v_1}, \mathbf{v_2}, \mathbf{v_3}, \mathbf{v_4} \in \{001, 011, 101, 111\}$, $\mathbf{v_1}$ and $\mathbf{v_3}$ differ only in the MSB and so do $\mathbf{v_2}$ and $\mathbf{v_4}$.

By Lemma 1, edge labels $\mathbf{a}$ and $\mathbf{b}$ are distinct and $\mathbf{a}, \mathbf{b} \in \{\mathbf{v_2}, \mathbf{v_4}\}$, so edge labels $\mathbf{a}$ and $\mathbf{b}$ can be labeled in two different ways. In Fig. 3.5(b), $\mathbf{a} = \mathbf{v_2}$ and $\mathbf{b} = \mathbf{v_4}$, using the same argument as labeling quasi-Gray-coded 8-QAM at Step 3 or Step 4, we can get $\mathbf{c} = \mathbf{v_1}$ and $\mathbf{d} = \mathbf{v_1}$. Similarly, with $\mathbf{a} = \mathbf{v_4}$ and $\mathbf{b} = \mathbf{v_2}$, we can get the alternate set-partition labeling structure shown in Fig. 3.5(c).



(a) Step 1.

(b) Step 2.

(c) Alternate Step 2.

Figure 3.5: Steps for achieving set-partiton labeling structures for 8-QAM constellations.

Selecting any point in Fig. 3.5(b) or Fig. 3.5(c) as a zero point and choosing $\mathbf{v_1}, \mathbf{v_2}, \mathbf{v_3}, \mathbf{v_4}$ from $\{001, 011, 101, 111\}$, we can obtain all the set-partition labelings for the 8-QAM constellation. The labeling structures of Fig. 3.5(b) and Fig. 3.5(c) are exactly the same as those of Fig. 3.3(a) and Fig. 3.3(b) respectively (the two superior edge profile labeling structures for 8-QAM). Thus for the 8-QAM constellation, any set-partition labeling is edge profile superior.

For clarity, we have restricted attention to traditional set-partition labelings where more significant bits are used deeper in the partition tree. However, the proof easily generalizes to any branch labeling. □

From the above proof, note that there are two distinct underlying structures for set-partitioned 8-QAM labelings (as Fig. 3.5(b) and Fig. 3.5(c)). Two labelings for 8-QAM which have the respective structures as Fig. 3.5(b) and Fig. 3.5(c) are listed below in octal form.

	7	6	
1	0	5	4
	3	2	

(a)

	7	2	
1	0	5	4
	3	6	

(b)

Figure 3.6: Two distinct labelings for set-partitioned 8-QAM

3.2 Superior Edge Profile Quasi-Gray-Code 32-QAM constellation labelings

For the 32-QAM constellation, at least six minimum distance edge labels are required [34]. We restrict attention to quasi-Gray-coded 32-QAM labelings, i. e., those with exactly six minimum distance edge labels in 32-QAM. This section constructs six distinct edge profile upper bounds that limit the edge profile of all possible quasi-Gray-coded 32-QAM labelings. Then six distinct quasi-Gray-coded 32-QAM labelings which achieve the respective six upper bounds are presented. One of these labeling structures dominates all other five in terms of edge profile. Finally, we show that no set-partition labeling can achieve the superior edge profile of the 32-QAM constellation. Appendix A proves several lemmas used in this section.

3.2.1 Six distinct edge profile upper bounds for quasi-Gray-coded 32-QAM

Using the same technique as for the 8-QAM constellation, i. e., enumerating the edges from a central point in the 32-QAM constellation, combined with the fact that there are at least six distinct minimum distance edge labels in the 32-QAM constellation [34] results in an upper bound of 32-QAM edge profiles shown in Table 3.3 [20]. As usual, nearest neighbors are separated by minimum distance.

Upper bound	1	1	1	1	1	1	2	2	4	4	4	4	5	5	5	5	5	5	5	5	8	8	8	9	9	10	10	10	10	13	13

Table 3.3: An upper bound for 32-QAM edge profile.

However, based on the following discussion, this upper bound is not achievable. We begin with a lemma that partially characterizes the labeling structure imposed by six distinct minimum distance edge labels.

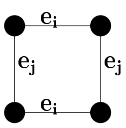
Lemma 7 *In any valid quasi-Gray-coded 32-QAM constellation labeling, one minimum distance edge label is equal to the XOR summation of three other distinct minimum distance edge labels.*

Proof:

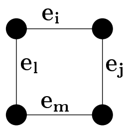
In any valid quasi-Gray-coded 32-QAM constellation labeling, there are six distinct minimum distance edge labels. Call these labels $\mathbf{e_1}$, $\mathbf{e_2}$, $\mathbf{e_3}$, $\mathbf{e_4}$, $\mathbf{e_5}$ and $\mathbf{e_6}$ where $\mathbf{e_1}$ through $\mathbf{e_5}$ are the linearly independent vectors. Then $\mathbf{e_6}$ is some linear combination of these linearly independent vectors,

$$\mathbf{e_6} = \mathbf{k_1 e_1} \oplus \mathbf{k_2 e_2} \oplus \mathbf{k_3 e_3} \oplus \mathbf{k_4 e_4} \oplus \mathbf{k_5 e_5}\,, \quad \mathbf{k_i} \in \{\mathbf{0}, \mathbf{1}\}. \tag{3.8}$$

According to labeling lemmas 1 and 2, the valid edge labels for a minimum distance square are shown in Fig. 3.7 where $\mathbf{i}, \mathbf{j}, \mathbf{l}, \mathbf{m}$ are distinct and from $\{1, 2, 3, 4, 5, 6\}$. We will show that it is impossible to label the 32-QAM constellation only as in Fig. 3.7(a). In other words, the structure shown in Fig. 3.7(b) must appear in any valid 32-QAM constellation labeling. For the 32-QAM constellation, by Lemma 5, each row (column) with six constellation points is labeled with at least three distinct minimum distance edge labels, say $\mathbf{e_i}$, $\mathbf{e_j}$, $\mathbf{e_k}$



(a) $\mathbf{e_i}$ and $\mathbf{e_j}$ are distinct.

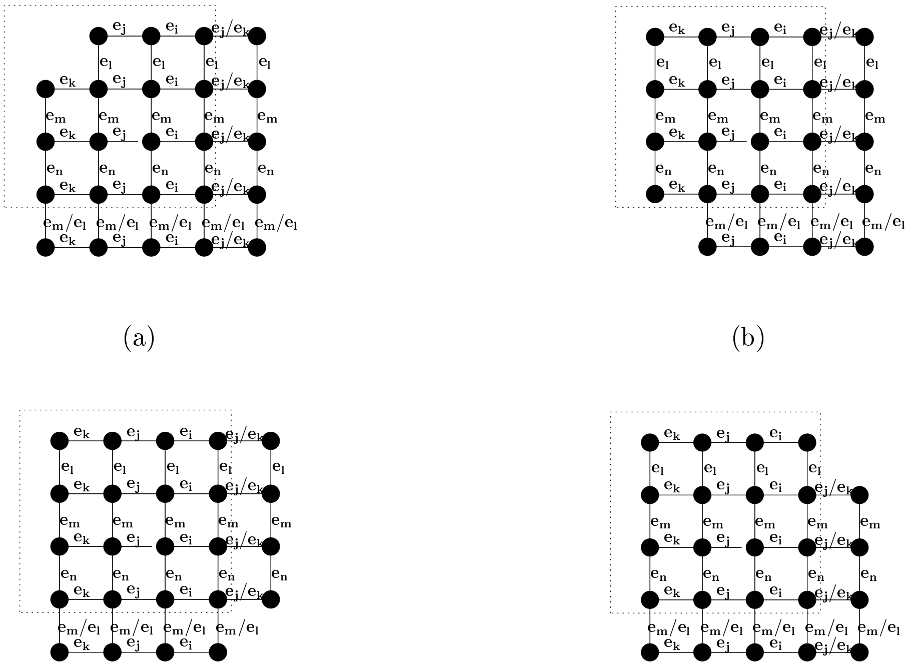


(b) One is $\mathbf{e_6}$.

Figure 3.7: Valid edge labels in quasi-Gray coded 32-QAM.

($\mathbf{e_l}$, $\mathbf{e_m}$, $\mathbf{e_n}$). Now assuming we label 32-QAM only as in Fig. 3.7(a) (denoted as Assumption 1), then the horizontal edges sharing the same two vertical columns all have the same edge label, and the vertical edges sharing the same two horizontal rows all have the same edge label, thus any horizontal edge label must be distinct from any vertical edge label by Lemma 1. Therefore, $\mathbf{e_i}$, $\mathbf{e_j}$, $\mathbf{e_k}$, $\mathbf{e_l}$, $\mathbf{e_m}$ and $\mathbf{e_n}$ must be distinct and $\mathbf{i}, \mathbf{j}, \mathbf{k}, \mathbf{l}, \mathbf{m}, \mathbf{n} \in \{1, 2, 3, 4, 5, 6\}$. Under Assumption 1, there are many kinds of labelings for 32-QAM. However, by using the symmetry of 32-QAM, these labelings can be classified into four distinct labeling structures according to the position of a certain subconstellation labeling. The four kinds of subconstellation labeling structures are shown in Fig. 3.8(a)-(d).

In any such labeling, one of the six minimum distance edge labels must be a linear combination of the other five minimum distance edge labels. However, for any linear combination, a path (not a cycle) consisting of this edge label and its linear combination of other edge labels can be found, which means this edge label is not equal to its corresponding linear combination by Lemma 2. This contradicts the fact that one of the six minimum distance edge labels must be some linear combination of the basis vectors (see Equation (3.8)). Thus all such kinds of labeling structures under Assumption 1 are invalid, therefore Fig. 3.7(b)



(a) (b) (c) (d)

Figure 3.8: Four kinds of subconstellation labeling structures under Assumption 1 (the classification subconstellation labeling structure is inside the dotted rectangular box), $\mathbf{e_j}/\mathbf{e_k}$ meaning either $\mathbf{e_j}$ or $\mathbf{e_k}$.

must appear in any valid quasi-Gray coded 32-QAM labeling, i.e.,

$$\mathbf{e_6} = \mathbf{e_x} \oplus \mathbf{e_y} \oplus \mathbf{e_z}\,, \quad \text{x, y, z are distinct and from } \{1, 2, 3, 4, 5\}. \tag{3.9}$$

□

Without loss of generality, let $\mathbf{e_1}$, $\mathbf{e_2}$, $\mathbf{e_3}$, $\mathbf{e_4}$, $\mathbf{e_5}$ and $\mathbf{e_6}$ be the six distinct minimum distance edge labels where $\mathbf{e_1}$ through $\mathbf{e_5}$ are the basis vectors and

$$\mathbf{e_6} = \mathbf{e_3} \oplus \mathbf{e_4} \oplus \mathbf{e_5}\,. \tag{3.10}$$

Under these assumptions, according to Lemma 1 and Lemma 2, $\mathbf{e_1}$ and $\mathbf{e_2}$ can only be used as in Figure 3.7(a). If one $\mathbf{e_1}$ labels a horizontal edge, then the horizontal edges sharing the same two columns as that $\mathbf{e_1}$ must also be labeled with $\mathbf{e_1}$. If one $\mathbf{e_1}$ is also used to label a vertical edge, then the vertical edges sharing the same two rows as that $\mathbf{e_1}$ must also be labeled with $\mathbf{e_1}$. Thus invalid subconstellation labelings as Fig. 3.9 would appear in 32-QAM labeling, which contradicts Lemma 1. Therefore we obtain another labeling lemma.

Figure 3.9: Invalid subconstellation labeling if $\mathbf{e_1}$ labels both horizontal and vertical edges.

Lemma 8 *In any valid quasi-Gray-coded 32-QAM, with minimum distance edge labels* $\mathbf{e_1}$, $\mathbf{e_2}$, $\mathbf{e_3}$, $\mathbf{e_4}$, $\mathbf{e_5}$, $\mathbf{e_6}$ *and* $\mathbf{e_6} = \mathbf{e_3} \oplus \mathbf{e_4} \oplus \mathbf{e_5}$, $\mathbf{e_1}$ *and* $\mathbf{e_2}$ *must be used as Fig. 3.7(a). Neither may label both horizontal and vertical edges.*

From Lemma 7, several additional lemmas that characterize the edge profile labeling structures of quasi-Gray-coded 32-QAM can be derived. The following three lemmas apply to any valid quasi-Gray-coded 32-QAM constellation labeling, with minimum distance edge labels $\mathbf{e_1}$, $\mathbf{e_2}$, $\mathbf{e_3}$, $\mathbf{e_4}$, $\mathbf{e_5}$, $\mathbf{e_6}$ and $\mathbf{e_6} = \mathbf{e_3} \oplus \mathbf{e_4} \oplus \mathbf{e_5}$.

Lemma 9 *The edge profile structure must conform to the following table, assuming the minimum distance is one (The proof is given in Appendix A).*

Lemma 10 $\mathbf{e_1}$ *and* $\mathbf{e_2}$ *can not both be used to label vertical edges or both be used to label horizontal edges. As a result, either* $\mathbf{e_1}$ *labels horizontal edges and* $\mathbf{e_2}$ *labels vertical edges or vice versa (The proof is given in Appendix A).*

The number of minimum-distance basis vectors in edge labels	Possible d^2_{MIN}	The number of edge labels
1	1	$\binom{5}{1} = 5$
2, 4	2, 4, 8, 10	$\binom{5}{2} + \binom{5}{4} = 15$
3, 5	1*, 5, 9, 13	$\binom{5}{3} + \binom{5}{5} = 11$

Table 3.4: The edge profile structure of a valid quasi-Gray-coded 32-QAM constellation labeling (*1 only possible here for the edge label $\mathbf{e_6} = \mathbf{e_3} \oplus \mathbf{e_4} \oplus \mathbf{e_5}$).

Without loss of generality, in the following discussion, let $\mathbf{e_1}$ label horizontal edges and $\mathbf{e_2}$ label vertical edges, then we have Lemma 11.

Lemma 11 *If* $\mathbf{e_1}$ *appears exactly once in a six-point horizontal row, then* $\mathbf{e_1}$ *must appear in the central position;* $\mathbf{e_1}$ *can not appear twice in a six-point horizontal row. The same labeling rules are followed by* $\mathbf{e_2}$ *appearing in a six-point vertical column (The proof is given in Appendix A).*

After obtaining these lemmas, the edge profile labeling structures for quasi-Gray-coded 32-QAM can be investigated.

In any valid quasi-Gray-coded 32-QAM constellation labeling, according to Lemmas 1, 2, 8, 10 and 11, $\mathbf{e_1}$ is used to label horizontal edges and can appear exactly once or three times in a six-point horizontal row; and $\mathbf{e_2}$ is used to label vertical edges and can appear exactly once or three times in a six-point vertical column. Therefore there are four cases to be investigated:

1. exactly one $\mathbf{e_1}$ in a six-point horizontal row and exactly one $\mathbf{e_2}$ in a six-point vertical column.

2. exactly one $\mathbf{e_1}$ in a six-point horizontal row and exactly three $\mathbf{e_2}$ in a six-point vertical column.

3. exactly three $\mathbf{e_1}$ in a six-point horizontal row and exactly one $\mathbf{e_2}$ in a six-point vertical column.

4. exactly three $\mathbf{e_1}$ in a six-point horizontal row and exactly three $\mathbf{e_2}$ in a six-point vertical column.

Using the symmetry of $\mathbf{e_1}$ and $\mathbf{e_2}$, we need to consider only the cases of 1, 2 and 4.

Case 1: $\mathbf{e_1}$ appears exactly once in a six-point horizontal row and $\mathbf{e_2}$ appears exactly once in a six-point vertical column.

The structure for Case 1 is shown in Figure 3.10(a).

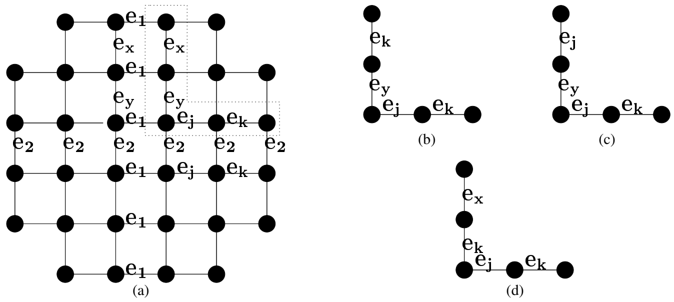
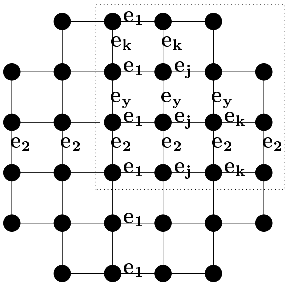


Figure 3.10: (a) 32-QAM labeling structure of Case 1. (b), (c), (d) Subconstellation labelings of Case 1.

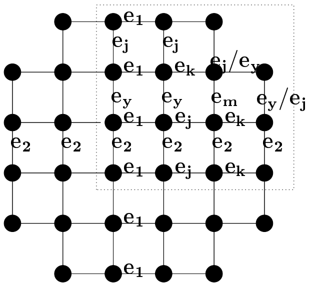
First consider the labelings of the subconstellation surrounded by dotted lines in Fig. 3.10(a) where $\mathbf{e_x} \neq \mathbf{e_y}$, $\mathbf{e_y} \neq \mathbf{e_j}$, $\mathbf{e_j} \neq \mathbf{e_k}$ by Lemma 1, $\mathbf{x}, \mathbf{y}, \mathbf{j}, \mathbf{k}$ can not be $\{1, 2\}$ under the assumption of Case 1, thus $\mathbf{x}, \mathbf{y}, \mathbf{j}, \mathbf{k} \in \{3, 4, 5, 6\}$. According to Lemma 5, at least three of the vectors $\mathbf{e_x}$, $\mathbf{e_y}$, $\mathbf{e_j}$ and $\mathbf{e_k}$ must be linearly independent. By Equation (3.10), $\mathbf{e_x}$, $\mathbf{e_y}$, $\mathbf{e_j}$ and $\mathbf{e_k}$ can not be four distinct vectors since this would force a violation of Lemma 2. Thus two must be equal among $\mathbf{e_x}$, $\mathbf{e_y}$, $\mathbf{e_j}$ and $\mathbf{e_k}$. By following the labeling lemmas, the only three possible types of labelings for the subconstellation are shown in Fig. 3.10(b)-(d).

Using the symmetry of $\mathbf{e_1}$ and $\mathbf{e_2}$, Fig. 3.10(c) and Fig. 3.10(d) are isomorphic. Therefore only the types in Fig. 3.10(b) and Fig. 3.10(c) need to be considered.

Fig. 3.11 shows the two resulting labeling structures for Case 1, where $\mathbf{e_j}$, $\mathbf{e_k}$, $\mathbf{e_y}$ and $\mathbf{e_m}$ are distinct, $\mathbf{j}, \mathbf{k}, \mathbf{y}, \mathbf{m} \in \{3, 4, 5, 6\}$ and $\mathbf{e_m} = \mathbf{e_j} \oplus \mathbf{e_k} \oplus \mathbf{e_y}$.



(a)



(b)

Figure 3.11: Two possible kinds of labeling structures for Case 1, $\mathbf{e_y}/\mathbf{e_j}$ meaning either $\mathbf{e_y}$ or $\mathbf{e_j}$.

From Fig. 3.11(a) and Fig. 3.11(b), we can obtain Table 3.5 which represents the upper bounds of squared Euclidean distances of edges in Case 1.

Case 2: $\mathbf{e_1}$ appears exactly once in a six-point horizontal row and $\mathbf{e_2}$ appears exactly three times in a six-point vertical column.

For this case, the labeling structure of 32-QAM is shown in Figure 3.12 where $\mathbf{e_x} \neq \mathbf{e_y}$ by Lemma 2, $\mathbf{e_x} \neq \mathbf{e_j}$, $\mathbf{e_y} \neq \mathbf{e_j}$ by Lemma 1. First consider the labelings of the subconstellation inside the dotted rectangular box. Six possible labelings for this subconstellation are shown in Figures 3.12(b)-(g) where $\mathbf{e_j}$, $\mathbf{e_x}$, $\mathbf{e_y}$ and $\mathbf{e_m}$ are distinct, $\mathbf{j}, \mathbf{x}, \mathbf{y}, \mathbf{m} \in \{3,\ 4,\ 5,\ 6\}$ and $\mathbf{e_m} = \mathbf{e_j} \oplus \mathbf{e_x} \oplus \mathbf{e_y}$.

Figures 3.13(a)-(f) show the corresponding six constellation labeling structures.

E in Case 1 (a)	$d^2_{\text{MIN}}(E) \leq$	E in Case 1 (b)	$d^2_{\text{MIN}}(E) \leq$
$\mathbf{e_1} \oplus \mathbf{e_2}$	2	$\mathbf{e_1} \oplus \mathbf{e_2}$	2
$\mathbf{e_1} \oplus \mathbf{e_k}$	2	$\mathbf{e_1} \oplus \mathbf{e_j}$	2
$\mathbf{e_1} \oplus \mathbf{e_y}$	2	$\mathbf{e_1} \oplus \mathbf{e_y}$	2
$\mathbf{e_2} \oplus \mathbf{e_j}$	2	$\mathbf{e_2} \oplus \mathbf{e_j}$	2
$\mathbf{e_2} \oplus \mathbf{e_k}$	2	$\mathbf{e_2} \oplus \mathbf{e_k}$	2
$\mathbf{e_k} \oplus \mathbf{e_j}$	2	$\mathbf{e_k} \oplus \mathbf{e_j}$	2
$\mathbf{e_k} \oplus \mathbf{e_y}$	2	$\mathbf{e_k} \oplus \mathbf{e_y}$	2
$\mathbf{e_y} \oplus \mathbf{e_j}$	2	$\mathbf{e_y} \oplus \mathbf{e_j}$	2
$\mathbf{e_1} \oplus \mathbf{e_j}$	4	$\mathbf{e_1} \oplus \mathbf{e_k}$	4
$\mathbf{e_2} \oplus \mathbf{e_y}$	4	$\mathbf{e_2} \oplus \mathbf{e_y}$	4
$\mathbf{e_y} \oplus \mathbf{e_2} \oplus \mathbf{e_1} \oplus \mathbf{e_j}$	8	$\mathbf{e_2} \oplus \mathbf{e_m}$	4
$\mathbf{e_1} \oplus \mathbf{e_k} \oplus \mathbf{e_y} \oplus \mathbf{e_j}$	8	$\mathbf{e_y} \oplus \mathbf{e_2} \oplus \mathbf{e_1} \oplus \mathbf{e_j}$	8
$\mathbf{e_2} \oplus \mathbf{e_k} \oplus \mathbf{e_y} \oplus \mathbf{e_j}$	8	$\mathbf{e_2} \oplus \mathbf{e_1} \oplus \mathbf{e_y} \oplus \mathbf{e_k}$	8
		$\mathbf{e_2} \oplus \mathbf{e_j} \oplus \mathbf{e_y} \oplus \mathbf{e_k}$	8
$\mathbf{e_1} \oplus \mathbf{e_y} \oplus \mathbf{e_k}$	5	$\mathbf{e_1} \oplus \mathbf{e_y} \oplus \mathbf{e_k}$	5
$\mathbf{e_1} \oplus \mathbf{e_y} \oplus \mathbf{e_j}$	5	$\mathbf{e_1} \oplus \mathbf{e_y} \oplus \mathbf{e_j}$	5
$\mathbf{e_1} \oplus \mathbf{e_j} \oplus \mathbf{e_k}$	5	$\mathbf{e_1} \oplus \mathbf{e_j} \oplus \mathbf{e_k}$	5
$\mathbf{e_2} \oplus \mathbf{e_j} \oplus \mathbf{e_k}$	5	$\mathbf{e_2} \oplus \mathbf{e_j} \oplus \mathbf{e_k}$	5
$\mathbf{e_2} \oplus \mathbf{e_y} \oplus \mathbf{e_k}$	5	$\mathbf{e_2} \oplus \mathbf{e_y} \oplus \mathbf{e_k}$	5
$\mathbf{e_2} \oplus \mathbf{e_j} \oplus \mathbf{e_y}$	5	$\mathbf{e_2} \oplus \mathbf{e_j} \oplus \mathbf{e_y}$	5
$\mathbf{e_1} \oplus \mathbf{e_2} \oplus \mathbf{e_y}$	5	$\mathbf{e_1} \oplus \mathbf{e_2} \oplus \mathbf{e_y}$	5
$\mathbf{e_1} \oplus \mathbf{e_2} \oplus \mathbf{e_j}$	5	$\mathbf{e_1} \oplus \mathbf{e_2} \oplus \mathbf{e_j}$	5

Table 3.5: Upper bounds of edge profiles of Case 1 (a) (Fig. 3.11(a)) and Case 1 (b) (Fig. 3.11(b)).

Using the labeling lemmas, from Figures 3.13(a)-(f), we obtain Table 3.6, which presents the upper bounds of the constellation edge profiles where Fig.3.13(a) and Fig.3.13(c) are isomorphic and so are Fig.3.13(b) and Fig.3.13(d).

Case 4: $\mathbf{e_1}$ appears exactly three times in a six-point horizontal row and $\mathbf{e_2}$ appears exactly three times in a six-point vertical column.

Fig. 3.14(a) shows 32-QAM labeling structure for Case 4. All unlabeled edges must be labeled by using only $\mathbf{e_3}$ through $\mathbf{e_6}$. In Fig. 3.14(a), since edge label sums are GF(2) operations, arbitrarily starting from a point whose label is assumed as

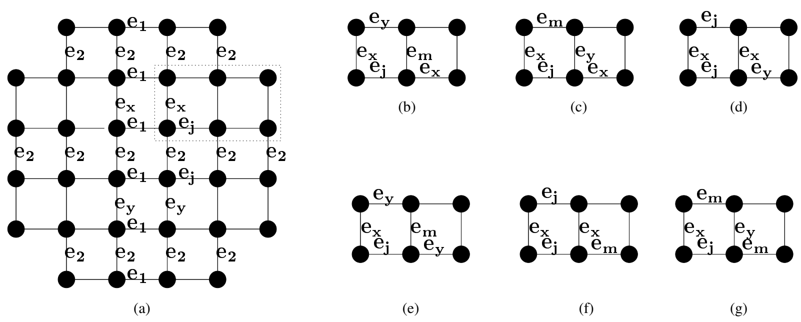


Figure 3.12: (a) 32-QAM labeling structure of Case 2. (b)-(g) six possible labelings for the subconstellation of Case 2.

$\mathbf{g_1}$, 7 different edges can be found whose edge labels are linear combinations of only $\mathbf{e_3}$ through $\mathbf{e_6}$, the corresponding 7 different point labels are $\mathbf{g_2}$ through $\mathbf{g_8}$. Also in Fig. 3.14(a), the square point has the label $\mathbf{g_x} = \mathbf{g_8} \oplus \mathbf{e_1} \oplus \mathbf{e_x}$, where $\mathbf{x} \in \{3, 4, 5, 6\}$. Then we group the 32-QAM constellation as in Fig. 3.14(a) and obtain Fig. 3.14(b). $\mathbf{g_i}$ in Fig. 3.14(b) represents 3 or 4 points in Fig. 3.14(a) whose point labels come from $\{\mathbf{g_i}, \mathbf{g_i} \oplus \mathbf{e_1}, \mathbf{g_i} \oplus \mathbf{e_2}, \mathbf{g_i} \oplus \mathbf{e_1} \oplus \mathbf{e_2}\}$. Note that there are only three linearly independent vectors in $\mathbf{e_3}$ through $\mathbf{e_6}$. These vectors can label at most 7 nonzero edges starting from a point by Lemma 4. Therefore $\mathbf{g_x} = \mathbf{g_8} \oplus \mathbf{e_1} \oplus \mathbf{e_x} = \mathbf{g_j} \oplus \mathbf{e_1}$, where $\mathbf{g_j} \in \{\mathbf{g_2}, \mathbf{g_3}, \mathbf{g_4}, \mathbf{g_5}, \mathbf{g_6}, \mathbf{g_7}, \mathbf{g_8}\}$. The three points in group $\mathbf{g_x}$ in Fig. 3.14(b) have point labels $\mathbf{g_x}, \mathbf{g_x} \oplus \mathbf{e_2}, \mathbf{g_x} \oplus \mathbf{e_2} \oplus \mathbf{e_1}$, i.e., $\mathbf{g_j} \oplus \mathbf{e_1}, \mathbf{g_j} \oplus \mathbf{e_1} \oplus \mathbf{e_2}, \mathbf{g_j} \oplus \mathbf{e_2}$. Then in Fig. 3.14(a), at least two points in group $\mathbf{g_x}$ have the same point labels as the other two points in other groups, which contradicts the fact that different constellation points must have different point labels according to Lemma 1. Therefore the 32-QAM labeling structure of Case 4 is invalid. Thus Case 4 can not appear in any valid quasi-Gray-coded 32-QAM labeling.

Using Table 3.4 in Lemma 9, Table 3.5 and Table 3.6, we obtain upper bounds

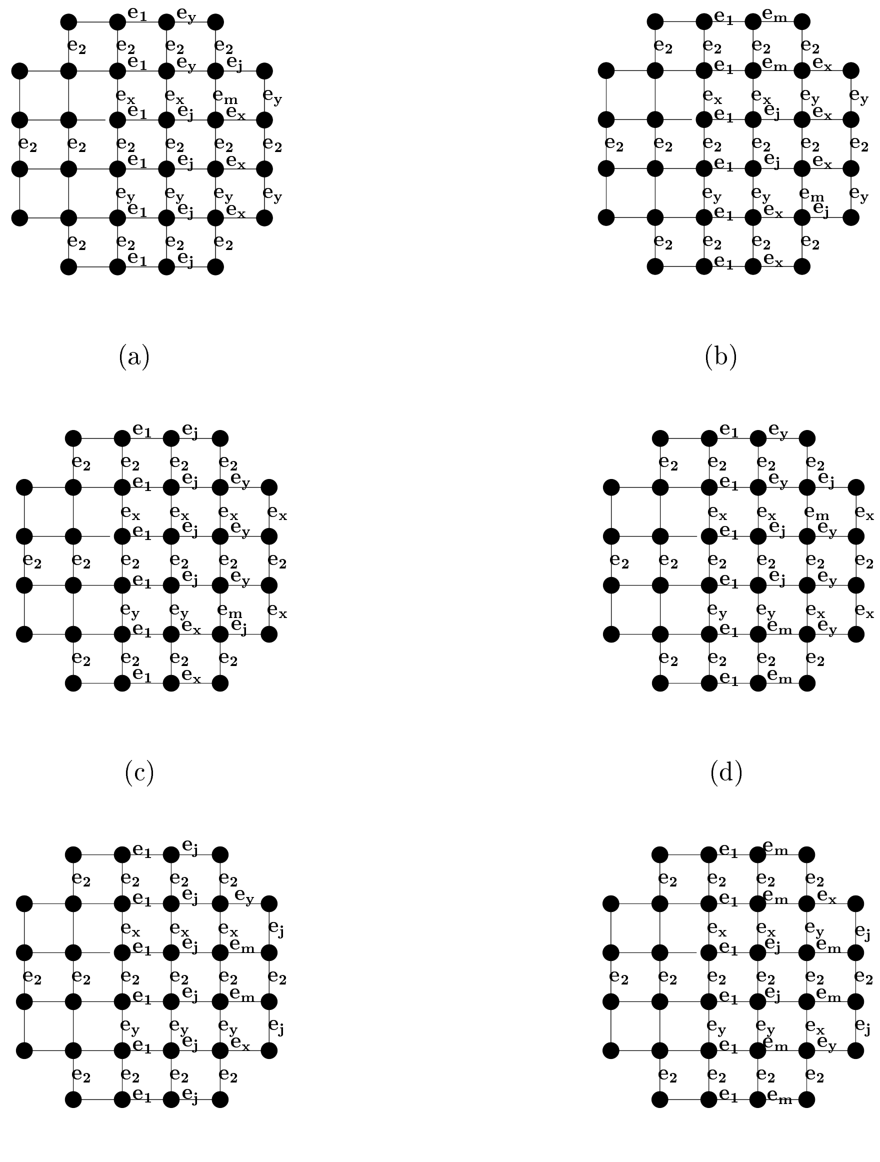


Figure 3.13: Six possible labelings for Case 2.

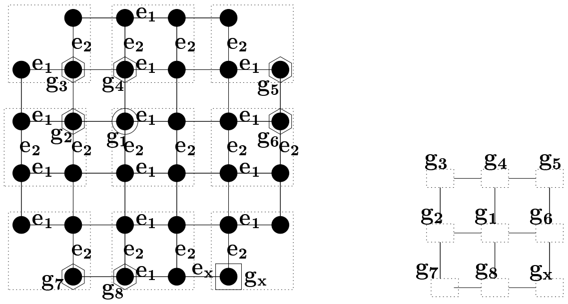
E *in Case* 2 $(a)(c)$	$d^2_{\text{MIN}}(E) \leq$	E *in Case* 2 $(b)(d)$	$d^2_{\text{MIN}}(E) \leq$	E *in Case* 2 (e)	$d^2_{\text{MIN}}(E) \leq$	E *in Case* 2 (f)	$d^2_{\text{MIN}}(E) \leq$
$\mathbf{e_1 \oplus e_2}$	2	$\mathbf{e_1 \oplus e_2}$	2	$\mathbf{e_1 \oplus e_2}$	2	$\mathbf{e_1 \oplus e_2}$	2
$\mathbf{e_1 \oplus e_x}$	2	$\mathbf{e_1 \oplus e_x}$	2	$\mathbf{e_1 \oplus e_x}$	2	$\mathbf{e_1 \oplus e_x}$	2
$\mathbf{e_1 \oplus e_y}$	2	$\mathbf{e_1 \oplus e_y}$	2	$\mathbf{e_1 \oplus e_y}$	2	$\mathbf{e_1 \oplus e_y}$	2
$\mathbf{e_2 \oplus e_j}$	2	$\mathbf{e_2 \oplus e_j}$	2	$\mathbf{e_2 \oplus e_j}$	2	$\mathbf{e_2 \oplus e_j}$	2
$\mathbf{e_2 \oplus e_x}$	2	$\mathbf{e_2 \oplus e_x}$	2	$\mathbf{e_2 \oplus e_x}$	2	$\mathbf{e_2 \oplus e_m}$	2
$\mathbf{e_2 \oplus e_y}$	2	$\mathbf{e_2 \oplus e_m}$	2	$\mathbf{e_2 \oplus e_y}$	2	$\mathbf{e_2 \oplus e_x}$	2
$\mathbf{e_y \oplus e_j}$	2	$\mathbf{e_y \oplus e_j}$	2	$\mathbf{e_2 \oplus e_m}$	2	$\mathbf{e_2 \oplus e_y}$	2
$\mathbf{e_x \oplus e_y}$	2	$\mathbf{e_x \oplus e_y}$	2	$\mathbf{e_y \oplus e_j}$	2	$\mathbf{e_y \oplus e_j}$	2
$\mathbf{e_x \oplus e_j}$	2	$\mathbf{e_x \oplus e_j}$	2	$\mathbf{e_x \oplus e_y}$	2	$\mathbf{e_x \oplus e_j}$	2
				$\mathbf{e_x \oplus e_j}$	2	$\mathbf{e_x \oplus e_y}$	2
$\mathbf{e_1 \oplus e_j}$	4	$\mathbf{e_1 \oplus e_j}$	4	$\mathbf{e_1 \oplus e_j}$	4	$\mathbf{e_1 \oplus e_j}$	4
$\mathbf{e_2 \oplus e_m}$	4	$\mathbf{e_1 \oplus e_m}$	4			$\mathbf{e_1 \oplus e_m}$	4
		$\mathbf{e_2 \oplus e_y}$	4				
$\mathbf{e_x \oplus e_2 \oplus e_1 \oplus e_j}$	8	$\mathbf{e_x \oplus e_2 \oplus e_1 \oplus e_j}$	8	$\mathbf{e_x \oplus e_2 \oplus e_1 \oplus e_j}$	8	$\mathbf{e_x \oplus e_2 \oplus e_1 \oplus e_j}$	8
$\mathbf{e_j \oplus e_1 \oplus e_2 \oplus e_y}$	8	$\mathbf{e_j \oplus e_1 \oplus e_2 \oplus e_y}$	8	$\mathbf{e_j \oplus e_1 \oplus e_2 \oplus e_y}$	8	$\mathbf{e_j \oplus e_1 \oplus e_2 \oplus e_y}$	8
$\mathbf{e_y \oplus e_1 \oplus e_2 \oplus e_x}$	8	$\mathbf{e_1 \oplus e_j \oplus e_2 \oplus e_m}$	8				
$\mathbf{e_1 \oplus e_x \oplus e_j}$	5	$\mathbf{e_1 \oplus e_x \oplus e_j}$	5	$\mathbf{e_1 \oplus e_x \oplus e_j}$	5	$\mathbf{e_1 \oplus e_x \oplus e_j}$	5
$\mathbf{e_1 \oplus e_y \oplus e_j}$	5	$\mathbf{e_1 \oplus e_y \oplus e_j}$	5	$\mathbf{e_1 \oplus e_y \oplus e_j}$	5	$\mathbf{e_1 \oplus e_y \oplus e_j}$	5
$\mathbf{e_1 \oplus e_x \oplus e_y}$	5	$\mathbf{e_1 \oplus e_x \oplus e_y}$	5	$\mathbf{e_2 \oplus e_j \oplus e_y}$	5	$\mathbf{e_2 \oplus e_j \oplus e_y}$	5
$\mathbf{e_2 \oplus e_j \oplus e_y}$	5	$\mathbf{e_2 \oplus e_j \oplus e_y}$	5	$\mathbf{e_2 \oplus e_j \oplus e_x}$	5	$\mathbf{e_2 \oplus e_j \oplus e_x}$	5
$\mathbf{e_2 \oplus e_j \oplus e_x}$	5	$\mathbf{e_2 \oplus e_j \oplus e_x}$	5	$\mathbf{e_2 \oplus e_x \oplus e_y}$	5	$\mathbf{e_2 \oplus e_j \oplus e_m}$	5
$\mathbf{e_2 \oplus e_x \oplus e_y}$	5	$\mathbf{e_2 \oplus e_x \oplus e_y}$	5	$\mathbf{e_1 \oplus e_2 \oplus e_y}$	5	$\mathbf{e_1 \oplus e_2 \oplus e_y}$	5
$\mathbf{e_1 \oplus e_2 \oplus e_y}$	5	$\mathbf{e_1 \oplus e_2 \oplus e_y}$	5	$\mathbf{e_1 \oplus e_2 \oplus e_x}$	5	$\mathbf{e_1 \oplus e_2 \oplus e_x}$	5
$\mathbf{e_1 \oplus e_2 \oplus e_x}$	5	$\mathbf{e_1 \oplus e_2 \oplus e_x}$	5	$\mathbf{e_1 \oplus e_2 \oplus e_j}$	5	$\mathbf{e_1 \oplus e_2 \oplus e_j}$	5
$\mathbf{e_1 \oplus e_2 \oplus e_j}$	5	$\mathbf{e_1 \oplus e_2 \oplus e_j}$	5			$\mathbf{e_1 \oplus e_2 \oplus e_m}$	5
		$\mathbf{e_1 \oplus e_2 \oplus e_m}$	5				
				$\mathbf{e_1 \oplus e_j \oplus e_m}$	9	$\mathbf{e_1 \oplus e_j \oplus e_m}$	9

Table 3.6: Upper bounds of edge profiles of Case 2.

of the edge profiles of Case 1(a)(b) and Case 2(a)-(f). There are a total of six distinct upper bounds shown in Table 3.7. Note that all these upper bounds of edge profiles are dominated by the upper bound of Case 1(a).

Based on the above discussion, the following theorem can be obtained.

Theorem 2 *Quasi-Gray-coded 32-QAM constellation labelings that have the same edge profile as the first row of Table 3.7 (Case 1(a)'s edge profile upper bound) are edge-profile superior to all other quasi-Gray-coded 32-QAM constellation labelings.*



(a) (b)

Figure 3.14: (a) The labeling structure for Case 4. (b) The grouped constellation labeling structure for Case 4.

Case 1 (a)	1	1	1	1	1	1	2	2	2	2	2	2	2	2	4	4	5	5	5	5	5	5	5	5	8	8	8	10	10	13	13
Case 1 (b)	1	1	1	1	1	1	2	2	2	2	2	2	2	2	4	4	4	5	5	5	5	5	5	5	5	8	8	8	10	13	13
Case 2 (a), (c)	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	4	4	5	5	5	5	5	5	5	5	5	8	8	8	10	13
Case 2 (b), (d)	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	4	4	4	5	5	5	5	5	5	5	5	5	5	8	8	8
Case 2 (e)	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2	4	5	5	5	5	5	5	5	5	8	8	9	10	10	13
Case 2 (f)	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2	4	4	5	5	5	5	5	5	5	5	5	8	8	9	10

Table 3.7: Six distinct upper bounds for quasi-Gray-coded 32-QAM edge profile.

3.2.2 Six distinct classes of quasi-Gray-coded 32-QAM

All these six distinct edge profile upper bounds for quasi-Gray-coded 32-QAM labelings are achievable. One labeling which achieves the best edge profile is shown in octal form in Fig. 3.15. All other labelings which achieve the best edge profile share the same labeling structure as Case 1(a).

Labelings that achieve the other five non-superior upper bounds are listed below (in octal form) in Fig. 3.16(a)-(e) respectively and share the same labeling structures as case 1(b), case2(a)(c), case2(b)(d), case2(e) and case2(f) respectively.

	23	21	25	27	
20	03	01	05	07	24
22	02	00	04	06	26
32	12	10	14	16	36
30	13	11	15	17	34
	33	31	35	37	

Figure 3.15: One best edge profile labeling for quasi-Gray-coded 32-QAM

	34	30	31	35	
04	24	20	21	25	05
14	10	00	01	11	15
16	12	02	03	13	17
06	26	22	23	27	07
	36	32	33	37	

(a)

	16	06	07	17	
34	14	04	05	15	35
24	20	00	01	21	25
26	22	02	03	23	27
36	32	12	13	33	37
	30	10	11	31	

(b)

	32	06	07	33	
34	30	04	05	31	35
24	20	00	01	21	25
26	22	02	03	23	27
36	16	12	13	17	37
	14	10	11	15	

(c)

	26	06	07	27	
34	24	04	05	25	35
14	20	00	01	21	15
16	22	02	03	23	17
36	32	12	13	33	37
	30	10	11	31	

(d)

	26	12	13	27	
34	24	10	11	25	35
30	04	00	01	05	31
32	06	02	03	07	33
36	16	22	23	17	37
	14	20	21	15	

(e)

Figure 3.16: Five non-superior edge profile labelings for quasi-Gray-coded 32-QAM.

3.2.3 Inferior set-partition labelings for 32-QAM

In this subsection, the set-partition labeling paradigm for 32-QAM is discussed and compared with the superior edge profile labelings of quasi-Gray-coded 32-QAM.

Rather than exhaustively search all possible set-partitionings, we show that the structure imposed by set-partitioning restricts the maximum edge profile in at least two ways. We begin with a lemma which partially characterizes the structure of 32-QAM labelings imposed by set-partitioning. This lemma and its proof are similar to Lemma 6 for the 8-QAM constellation.

In the following discussion, we partition 32-QAM at least to the D-level (8 cosets). The three-bit edge labels implied by the coset labeling are called the coset edge labels. The complete 5-bit edge label is not completely determined by the coset edge label. In the following discussion, we restrict attention to set partitioning where groups of points that share the same m least significant bits are maximally separated. It is easy to generalize to any other set partitioning.

Lemma 12 *Set-partitioning the 32-QAM constellation to eight cosets causes minimum distance horizontal (or vertical) coset edge labels to alternate between exactly two different three-bit labels, which differ only in the third (the most significant bit) bit. Together, the vertical and horizontal minimum distance coset edge labels consist of four distinct labels [20].*

Proof:

This lemma follows from an examination of the partitioning tree for 32-QAM such as the one shown in [35, page 13]. In this tree the A-level coset is the entire constellation. B-level cosets have sixteen points whose labels all share the same least significant bit. C-level cosets have eight points whose labels all share the same two least significant bits. D-level cosets have four points whose labels all share the same three least significant bits.

Consider the nearest neighbors of a point in the C-level coset C0. These nearest neighbors will all be in B1, the B-level coset that does not contain C0. Either the horizontal nearest neighbors are in C1 and the vertical neighbors are in C3, or vice versa. Because all neighbors of a particular orientation (horizontal or vertical) are in the same C-level coset, they have their two least significant bits in common. Thus the minimum distance edges of a given orientation connecting a point in C0 to these neighbors have coset edge labels that differ only in the third bit. The horizontal and vertical minimum distance edges emanating from C2 have the same coset edge labels as the respective edges emanating from points in C0 by symmetry. Therefore all the minimum distance edges have labels that obey the lemma. $\square$

We now use this lemma to prove two statements about the edge-profile limitations of a set-partitioned 32-QAM constellation.

Lemma 13 *Set-partitioned 32-QAM has no edge labels with minimum normalized squared distance greater than 10 [20].*

Proof:

For a 32-QAM constellation normalized so that the closest points are separated by minimum distance, consider the central point shown inside a square in Fig. 3.17. Every non-zero five-bit pattern must label exactly one of the 31 edges emanating from this point. The largest squared distance for an edge emanating from this point is 13 (occurring twice), and the next largest squared distance is 10.

To show that set partitioning forces all edge labels to have minimum distance of at most 10, we need only show that the five-bit labels for the two edges with

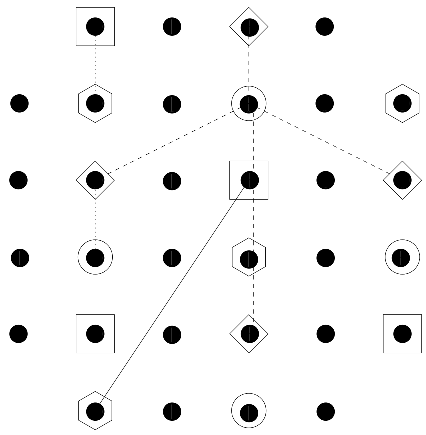


Figure 3.17: Proof of Lemma 13.

squared distance 13 emanating from the central square point must also be labels for edges with squared distances of 10 or less.

The alternating behavior proved in Lemma 12 implies that with set partitioning, the three least significant bits (LSBs) of the five-bit edge labels for the two dotted edges in Figure 3.17 must be identical. Every point in a given coset has the same three least significant bits. Thus all labels for edges connecting a diamond point to a circle point have the same three LSBs, and all labels for edges connecting a hexagon point to a square point have these same three LSBs.

With only two unspecified bits, there are only four labels with the three LSBs of interest. Each of these four labels must label exactly one of the four dashed edges shown in Figure 3.17 emanating from the circle point. The largest of these edges has squared distance 9. One of these edge labels must be the same as the label of the solid-line edge with squared distance 13 connecting the central square point to a hexagon point. Thus, the edge label of this edge has a minimum squared distance of 9 or less.

The same argument applies to the other edge with squared distance 13 emanating from the central square point. Therefore we can conclude that edge labels of the two edges with squared distance 13 emanating from the central square point have minimum squared distance 9 or less. Furthermore, since the next largest squared distance is 10, set partitioning forces the minimum distance associated with every edge label to be less than or equal to 10. $\square$

Lemma 14 *Set-partitioned 32-QAM has at least eight distinct edge labels for minimum distance edges.*

Proof:

From Lemma 12 there are two horizontal and two vertical three-bit minimum distance distinct coset edge labels $\mathbf{h_1}$, $\mathbf{h_2}$, $\mathbf{v_1}$, and $\mathbf{v_2}$ with $\mathbf{h_1} \oplus \mathbf{h_2} = \mathbf{v_1} \oplus \mathbf{v_2} = \mathbf{100}$. Coset edge labels only fix the three least significant bits of the edge label. Obviously, if there are two $\mathbf{h_1}$ coset edge labels that differ in their two MSBs, and two $\mathbf{h_2}$ coset edge labels that differ in their two MSBs, and two $\mathbf{v_1}$ coset edge labels that differ in their two MSBs, and two $\mathbf{v_2}$ coset edge labels that differ in their two MSBs, then the lemma is proven, i. e., set-partitioned 32-QAM has at least eight distinct edge labels for minimum distance edges.

For the rest of the argument, assume the above assumption is not true. By symmetry of horizontal rows and vertical columns, it is sufficient to consider the following cases:

A. All $\mathbf{h_1}$ coset edges have the same 2 MSBs.

B. All $\mathbf{h_2}$ coset edges have the same 2 MSBs.

In the following discussion, assume letters $\mathbf{e}, \mathbf{f}, \mathbf{g}, \mathbf{p}, \mathbf{q}, \mathbf{i}, \mathbf{j}, \mathbf{k}, \mathbf{l}, \mathbf{m}, \mathbf{u}, \mathbf{w}, \mathbf{x}, \mathbf{y}, \mathbf{z}$ denote the 2 MSBs in octal form and are chosen from $\{0, 1, 2, 3\}$.

CASE A: consider the case where all $\mathbf{h_1}$ coset edges have the same 2 MSBs. and $\mathbf{a_1}$, $\mathbf{a_2}$, $\mathbf{a_3}$, $\mathbf{a_4}$, $\mathbf{b_1}$, $\mathbf{b_2}$ and $\mathbf{b_3}$ are the distinct points shown in Fig. 3.18.

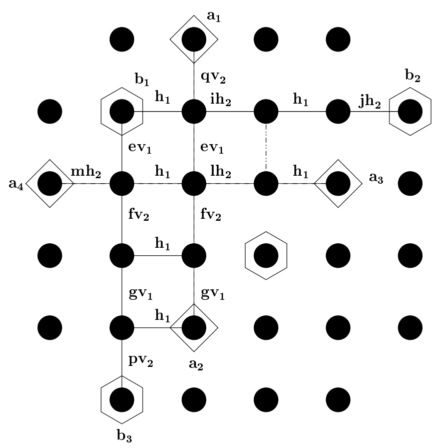


Figure 3.18: All $\mathbf{h_1}$ coset edges have the same 2 MSBs.

To make the proof more clear, we draw a diagram in Fig. 3.19 to show the procedure of the proof of CASE A.

Since $\mathbf{b_1}$, $\mathbf{b_2}$ and $\mathbf{b_3}$ are distinct points (see Fig. 3.18), we get

$$\mathbf{b_1} \oplus \mathbf{b_3} \neq \mathbf{b_1} \oplus \mathbf{b_2} \neq \mathbf{0}\,,$$

i. e.,

$$\mathbf{ev_1} \oplus \mathbf{fv_2} \oplus \mathbf{gv_1} \oplus \mathbf{pv_2} \neq \mathbf{ih_2} \oplus \mathbf{jh_2} \neq \mathbf{0}\,, \tag{3.11}$$

The coset structure also forces (see Fig. 3.18)

$$\mathbf{a_1} \oplus \mathbf{a_2} = \mathbf{a_3} \oplus \mathbf{a_4} \neq \mathbf{0}\,.$$

i. e.,

$$\mathbf{qv_2} \oplus \mathbf{ev_1} \oplus \mathbf{fv_2} \oplus \mathbf{gv_1} = \mathbf{lh_2} \oplus \mathbf{mh_2} \neq \mathbf{0}\,, \tag{3.12}$$

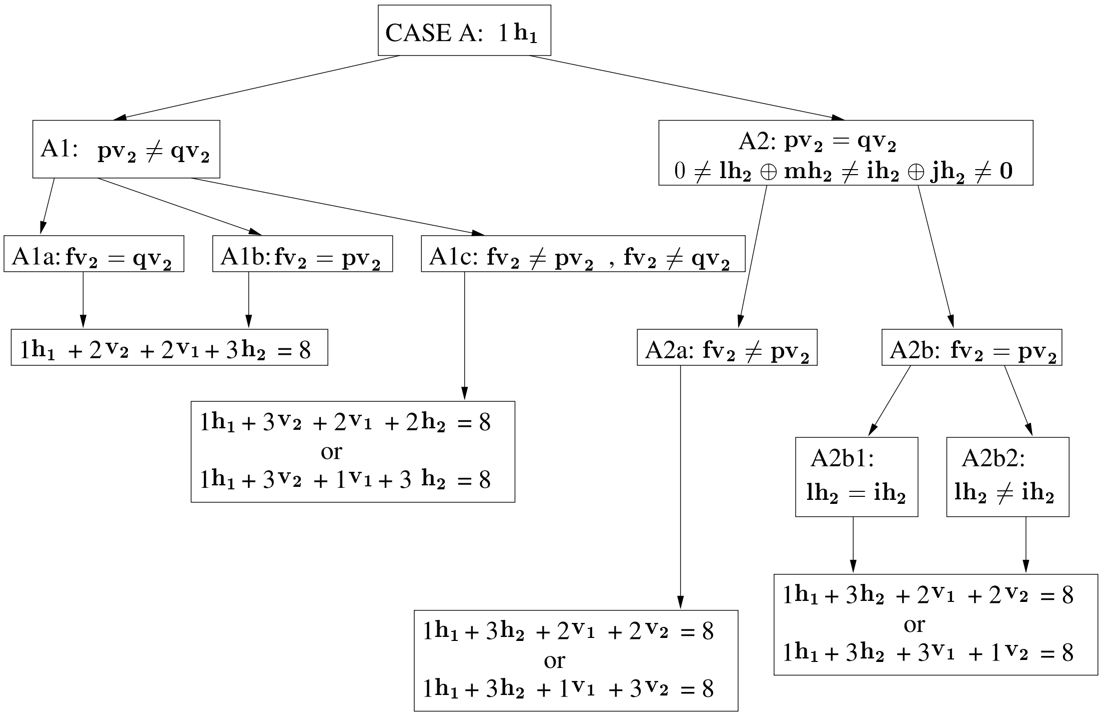


Figure 3.19: The procedure of the proof of CASE A of Lemma 14.

Comparing Equation (3.11) and Equation (3.12), if $\mathbf{ev_1} \oplus \mathbf{fv_2} \oplus \mathbf{gv_1} \oplus \mathbf{pv_2} \neq \mathbf{qv_2} \oplus \mathbf{ev_1} \oplus \mathbf{fv_2} \oplus \mathbf{gv_1}$, then

$$\mathbf{pv_2} \neq \mathbf{qv_2}\,, \tag{3.13}$$

if $\mathbf{ev_1} \oplus \mathbf{fv_2} \oplus \mathbf{gv_1} \oplus \mathbf{pv_2} = \mathbf{qv_2} \oplus \mathbf{ev_1} \oplus \mathbf{fv_2} \oplus \mathbf{gv_1}$, then $\mathbf{pv_2} = \mathbf{qv_2}$ and

$$\mathbf{lh_2} \oplus \mathbf{mh_2} \neq \mathbf{ih_2} \oplus \mathbf{jh_2} \neq \mathbf{0}\,. \tag{3.14}$$

Next we investigate these two cases in detail.

Case A1: All $\mathbf{h_1}$ coset edges have the same 2 MSBs and $\mathbf{ev_1} \oplus \mathbf{fv_2} \oplus \mathbf{gv_1} \oplus \mathbf{pv_2} \neq \mathbf{qv_2} \oplus \mathbf{ev_1} \oplus \mathbf{fv_2} \oplus \mathbf{gv_1}$, i. e., $\mathbf{pv_2} \neq \mathbf{qv_2}$. This case can be decomposed into three cases for investigation:

(A1a). All $\mathbf{h_1}$ coset edges have the same 2 MSBs, $\mathbf{pv_2} \neq \mathbf{qv_2}$ and $\mathbf{fv_2} = \mathbf{qv_2}$:

Since $\mathbf{pv_2} \neq \mathbf{qv_2}$, then two distinct $\mathbf{v_2}$ edge labels exist, denoted as $2\mathbf{v_2}$s.

Since $\mathbf{fv_2} = \mathbf{qv_2}$ and $\mathbf{a_1}$, $\mathbf{a_2}$ are distinct points (see Fig. 3.18), then $\mathbf{ev_1} \neq \mathbf{gv_1}$, i. e., $2\mathbf{v_1}$s exist.

The set-partition labeling structure of (a) is shown in Fig. 3.20.

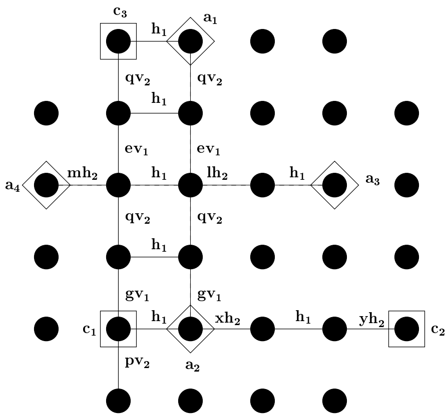


Figure 3.20: $\mathbf{pv_2} \neq \mathbf{qv_2}$, $\mathbf{fv_2} = \mathbf{qv_2}$ and $\mathbf{ev_1} \neq \mathbf{gv_1}$.

Since $\mathbf{c_1} \oplus \mathbf{c_3} \neq \mathbf{c_1} \oplus \mathbf{c_2} \neq \mathbf{0}$, then

$$\mathbf{ev_1} \oplus \mathbf{gv_1} \neq \mathbf{xh_2} \oplus \mathbf{yh_2}\,, \tag{3.15}$$

Since $\mathbf{a_1} \oplus \mathbf{a_2} = \mathbf{a_3} \oplus \mathbf{a_4} \neq \mathbf{0}$, then

$$\mathbf{ev_1} \oplus \mathbf{gv_1} = \mathbf{lh_2} \oplus \mathbf{mh_2} \neq \mathbf{0}\,. \tag{3.16}$$

Comparing equations (3.15) and (3.16), we get $0 \neq \mathbf{xh_2} \oplus \mathbf{yh_2} \neq \mathbf{lh_2} \oplus \mathbf{mh_2} \neq \mathbf{0}$. Therefore $3\mathbf{h_2}$s exist.

Thus, in (a), we get $2\mathbf{v_2}\mathbf{s} + \mathbf{2v_1s} + \mathbf{3h_2s} + \mathbf{1h_1}$. Therefore, in (a), there are at least eight distinct edge labels for minimum distance edges.

(A1b). All $\mathbf{h_1}$ coset edges have the same 2 MSBs, $\mathbf{pv_2} \neq \mathbf{qv_2}$ and $\mathbf{fv_2} = \mathbf{pv_2}$:

(A1b) is similar to (A1a). The same argument can be applied and the same result can be achieved.

(A1c). All $\mathbf{h_1}$ coset edges have the same 2 MSBs and $\mathbf{pv_2} \neq \mathbf{qv_2}$, $\mathbf{fv_2} \neq \mathbf{pv_2}$, $\mathbf{fv_2} \neq \mathbf{qv_2}$, Thus 3$\mathbf{v_2}$s exist:

Since all $\mathbf{h_1}$ coset edges have the same 2 MSBs, 1$\mathbf{h_1}$ and at least 2$\mathbf{h_2}$s exist by Lemma 5(the horizontal row has a total of six different points).

Therefore, if 2$\mathbf{v_1}$s exist, then we get 3$\mathbf{v_2}$s + 1$\mathbf{h_1}$ + 2$\mathbf{h_2}$s + 2$\mathbf{v_1}$s, the argument is complete.

If all $\mathbf{v_1}$ coset edges have the same 2 MSBs, 1$\mathbf{v_1}$ exists. The set-partition labeling structure is shown in Fig. 3.21.

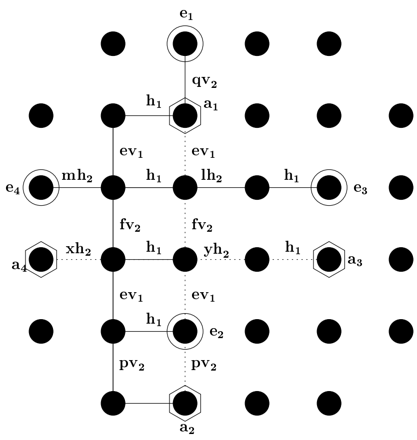


Figure 3.21: $\mathbf{pv_2} \neq \mathbf{qv_2}$, $\mathbf{fv_2} \neq \mathbf{pv_2}$, $\mathbf{fv_2} \neq \mathbf{qv_2}$ and $\mathbf{ev_1} = \mathbf{gv_1}$.

Since $\mathbf{e_1} \oplus \mathbf{e_2} = \mathbf{e_3} \oplus \mathbf{e_4} \neq \mathbf{0}$, we get

$$\mathbf{qv_2} \oplus \mathbf{fv_2} = \mathbf{lh_2} \oplus \mathbf{mh_2} \neq \mathbf{0}\,. \tag{3.17}$$

Since $\mathbf{a_1} \oplus \mathbf{a_2} = \mathbf{a_3} \oplus \mathbf{a_4} \neq \mathbf{0}$, we get

$$\mathbf{fv_2} \oplus \mathbf{pv_2} = \mathbf{xh_2} \oplus \mathbf{yh_2} \neq \mathbf{0}\,. \tag{3.18}$$

Comparing Equation (3.17) and Equation (3.18), since $\mathbf{qv_2} \neq \mathbf{fv_2}$, we get $0 \neq \mathbf{xh_2} \oplus \mathbf{yh_2} \neq \mathbf{lh_2} \oplus \mathbf{mh_2} \neq \mathbf{0}$, i. e., 3$\mathbf{h_2}$s exist. Therefore we get 3$\mathbf{v_2}$s + 1$\mathbf{h_1}$ + 3$\mathbf{h_2}$s + 1$\mathbf{v_1}$, the argument is complete.

Case A2: All $\mathbf{h_1}$ coset edges have the same 2 MSBs and $\mathbf{ev_1} \oplus \mathbf{fv_2} \oplus \mathbf{gv_1} \oplus \mathbf{pv_2}$ $= \mathbf{qv_2} \oplus \mathbf{ev_1} \oplus \mathbf{fv_2} \oplus \mathbf{gv_1}$, then $\mathbf{pv_2} = \mathbf{qv_2}$ and $0 \neq \mathbf{lh_2} \oplus \mathbf{mh_2} \neq \mathbf{ih_2} \oplus \mathbf{jh_2} \neq \mathbf{0}$. This case can be decomposed into two cases for investigation.

(A2a). All $\mathbf{h_1}$ coset edges have the same 2 MSBs, $\mathbf{pv_2} = \mathbf{qv_2}$, $0 \neq \mathbf{lh_2} \oplus \mathbf{mh_2} \neq \mathbf{ih_2} \oplus \mathbf{jh_2} \neq \mathbf{0}$ and $\mathbf{fv_2} \neq \mathbf{pv_2}$:

Since $\mathbf{lh_2} \oplus \mathbf{mh_2} \neq \mathbf{ih_2} \oplus \mathbf{jh_2} \neq \mathbf{0}$, $3\mathbf{h_2}$s exist.

Since $\mathbf{fv_2} \neq \mathbf{pv_2}$, at least $2\mathbf{v_2}$s exist.

If $2\mathbf{v_1}$s exist, then we get $3\mathbf{h_2}$s + $1\mathbf{h_1}$ + $2\mathbf{v_2}$s + $2\mathbf{v_1}$s, the argument is complete.

If all $\mathbf{v_1}$ coset edges have the same 2 MSBs, $1\mathbf{v_1}$ exists. The set-partition labeling structure is shown in Fig. 3.22.

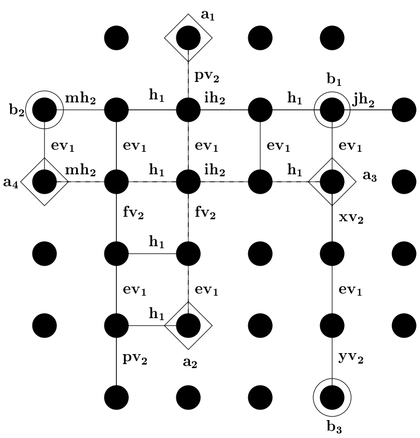


Figure 3.22: $\mathbf{fv_2} \neq \mathbf{pv_2}$, $\mathbf{qv_2} = \mathbf{pv_2}$ and $\mathbf{ev_1} = \mathbf{gv_1}$.

Since $\mathbf{b_1} \oplus \mathbf{b_3} \neq \mathbf{b_1} \oplus \mathbf{b_2}$, we get

$$\mathbf{ih_2} \oplus \mathbf{mh_2} \neq \mathbf{xv_2} \oplus \mathbf{yv_2}\,, \tag{3.19}$$

Since $\mathbf{a_1} \oplus \mathbf{a_2} = \mathbf{a_3} \oplus \mathbf{a_4} \neq \mathbf{0}$, we get

$$\mathbf{ih_2} \oplus \mathbf{mh_2} = \mathbf{pv_2} \oplus \mathbf{fv_2} \neq \mathbf{0}\,. \tag{3.20}$$

Comparing equations (3.19) and (3.20), we get $0 \neq \mathbf{xv_2} \oplus \mathbf{yv_2} \neq \mathbf{pv_2} \oplus \mathbf{fv_2} \neq \mathbf{0}$, i. e., $3\mathbf{v_2}$s exist. Therefore we get $3\mathbf{v_2}$s + $1\mathbf{h_1}$ + $3\mathbf{h_2}$s + $1\mathbf{v_1}$s, the argument is complete.

(A2b). All $\mathbf{h_1}$ coset edges have the same 2 MSBs, $\mathbf{pv_2} = \mathbf{qv_2}$, $0 \neq \mathbf{lh_2} \oplus \mathbf{mh_2} \neq \mathbf{ih_2} \oplus \mathbf{jh_2} \neq \mathbf{0}$ and $\mathbf{fv_2} = \mathbf{pv_2}$:

Since $\mathbf{fv_2} = \mathbf{pv_2} = \mathbf{qv_2}$ and $\mathbf{a_1}$, $\mathbf{a_2}$ are distinct points, at least $2\mathbf{v_1}$s exist (see Fig. 3.23, $\mathbf{ev_1} \neq \mathbf{gv_1}$).

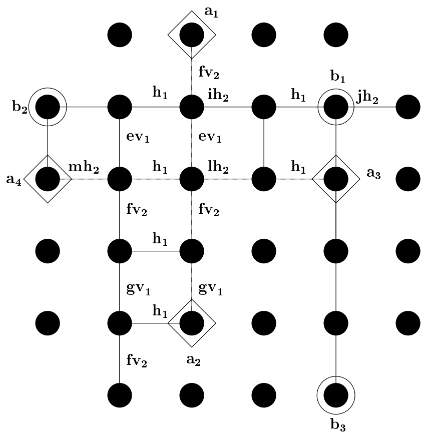


Figure 3.23: $\mathbf{fv_2} = \mathbf{pv_2} = \mathbf{qv_2}$ and $\mathbf{ev_1} \neq \mathbf{gv_1}$.

Since $0 \neq \mathbf{lh_2} \oplus \mathbf{mh_2} \neq \mathbf{ih_2} \oplus \mathbf{jh_2} \neq \mathbf{0}$, $3\mathbf{h_2}$s exist.

If $2\mathbf{v_2}$s exist, then we get $2\mathbf{v_2}$s + $1\mathbf{h_1}$ + $3\mathbf{h_2}$s + $2\mathbf{v_1}$s, the argument is complete.

If all $\mathbf{v_2}$ coset edges have the same 2 MSBs, $1\mathbf{v_2}$ exists. For this case, only two cases need to be investigated:

(A2b1). All $\mathbf{h_1}$ coset edges have the same 2 MSBs, all $\mathbf{v_2}$ coset edges have the same 2 MSBs, at least $3\mathbf{h_2}$s exist and $\mathbf{lh_2} = \mathbf{ih_2}$ in Fig. 3.23. In that case, the set-partition labeling structure is shown in Fig. 3.24.

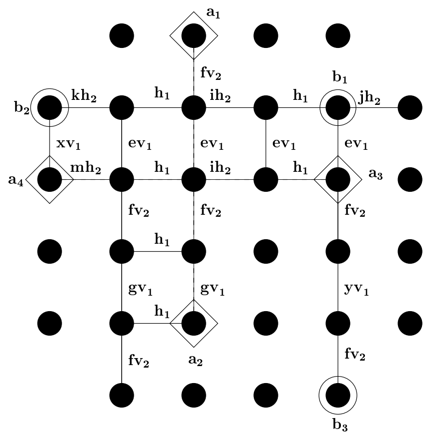


Figure 3.24: $\mathbf{fv_2} = \mathbf{pv_2} = \mathbf{qv_2}$, $\mathbf{ev_1} \neq \mathbf{gv_1}$ and $\mathbf{lh_2} = \mathbf{ih_2}$ (see Fig. 3.23).

Since $\mathbf{a_1} \oplus \mathbf{a_2} = \mathbf{a_3} \oplus \mathbf{a_4} \neq \mathbf{0}$, we get (see Fig. 3.24)

$$\mathbf{ev_1} \oplus \mathbf{gv_1} = \mathbf{ih_2} \oplus \mathbf{mh_2} \neq \mathbf{0}\,. \tag{3.21}$$

Since $\mathbf{b_1} \oplus \mathbf{b_2} \neq \mathbf{b_1} \oplus \mathbf{b_3}$, we get (see Fig. 3.24)

$$\mathbf{ev_1} \oplus \mathbf{yv_1} \neq \mathbf{ih_2} \oplus \mathbf{kh_2}\,. \tag{3.22}$$

By Lemma 2, we get (see Fig. 3.24)

$$\mathbf{xv_1} \oplus \mathbf{mh_2} = \mathbf{kh_2} \oplus \mathbf{ev_1} \neq \mathbf{0}\,,\ \mathbf{i.e.}\,\mathbf{xv_1} \oplus \mathbf{ev_1} = \mathbf{mh_2} \oplus \mathbf{kh_2} \neq \mathbf{0}\,. \tag{3.23}$$

From equations (3.21), (3.22) and (3.23), we get $0 \neq \mathbf{xv_1} \oplus \mathbf{yv_1} \neq \mathbf{ev_1} \oplus \mathbf{gv_1} \neq \mathbf{0}$, i.e., 3$\mathbf{v_1}$s exist. Therefore, we get 3$\mathbf{v_1}$s + 3$\mathbf{h_2}$s + 1$\mathbf{h_1}$ + 1$\mathbf{v_2}$, the argument is complete.

(A2b2). All $\mathbf{h_1}$ coset edges have the same 2 MSBs, all $\mathbf{v_2}$ coset edges have the same 2 MSBs, at least 3$\mathbf{h_2}$s exist and $\mathbf{lh_2} \neq \mathbf{ih_2}$ in Fig. 3.23. In this case,the set-partition labeling structure is shown in Fig. 3.25.

Since $\mathbf{a_1} \oplus \mathbf{a_2} = \mathbf{a_3} \oplus \mathbf{a_4} \neq \mathbf{0}$, we get

$$\mathbf{ev_1} \oplus \mathbf{gv_1} = \mathbf{mh_2} \oplus \mathbf{lh_2} \neq \mathbf{0}\,. \tag{3.24}$$

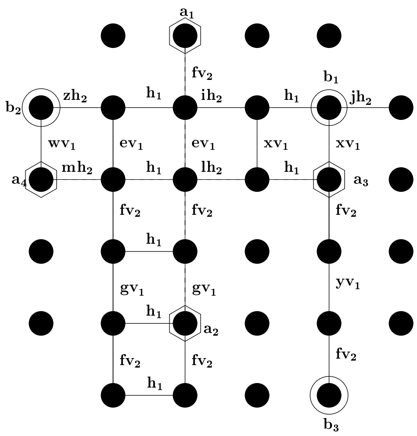


Figure 3.25: $\mathbf{fv_2} = \mathbf{pv_2} = \mathbf{qv_2}$, $\mathbf{ev_1} \neq \mathbf{gv_1}$, $\mathbf{lh_2} \neq \mathbf{ih_2}$.

By Lemma 2, we get

$$\mathbf{wv_1} \oplus \mathbf{mh_2} = \mathbf{zh_2} \oplus \mathbf{ev_1} \neq \mathbf{0}\,. \tag{3.25}$$

From Equation (3.24) and Equation (3.25), we get

$$\mathbf{wv_1} \oplus \mathbf{gv_1} = \mathbf{zh_2} \oplus \mathbf{lh_2} \neq \mathbf{0}\,. \tag{3.26}$$

Also by Lemma 2, we get

$$\mathbf{ev_1} \oplus \mathbf{lh_2} = \mathbf{ih_2} \oplus \mathbf{xv_1} \neq \mathbf{0}\,. \tag{3.27}$$

since $\mathbf{b_1} \oplus \mathbf{b_2} \neq \mathbf{b_1} \oplus \mathbf{b_3}$, we get

$$\mathbf{zh_2} \oplus \mathbf{ih_2} \neq \mathbf{xv_1} \oplus \mathbf{yv_1}\,. \tag{3.28}$$

From equations (3.27) and (3.28), we get

$$\mathbf{ev_1} \oplus \mathbf{yv_1} \neq \mathbf{zh_2} \oplus \mathbf{lh_2}\,. \tag{3.29}$$

Comparing equations (3.26) and (3.29), we get $0 \neq \mathbf{wv_1} \oplus \mathbf{gv_1} \neq \mathbf{ev_1} \oplus \mathbf{yv_1} \neq \mathbf{0}$, i.e, $3\mathbf{v_1}$s exist. Therefore, we get $3\mathbf{v_1}$s + $3\mathbf{h_2}$s + $1\mathbf{h_1}$ + $1\mathbf{v_2}$, the argument is complete.

CASE B: consider the case where all $\mathbf{h_2}$ coset edges have the same 2 MSBs.

The proof of CASE B is given in Appendix A. For CASE B, we can show that set-partitioned 32-QAM has at least eight distinct edge labels for minimum distance edges.

Since all cases have been covered, this lemma is proven.

$\square$

Based on the above discussion, under set-partitioning, the last two elements in the edge profile can be at most 10 (rather than 13 which is achieved by the superior edge profile of quasi-Gray-coded 32-QAM labelings in the first row of Table 3.7), and the first eight elements of the edge profile must be 1 (only six elements are 1 in the superior edge profile of quasi-Gray-coded 32-QAM labelings in the first row of Table 3.7). The superior edge profile of quasi-Gray-coded 32-QAM labelings are also given in the first row of Table 3.8. The second row in Table 3.8 gives the best edge profile we found using set-partitioning. Note that it is inferior to the superior edge profile of quasi-Gray-coded 32-QAM labelings.

superior edge profile	1	1	1	1	1	1	2	2	2	2	2	2	2	2	4	4	5	5	5	5	5	5	5	5	8	8	8	10	10	13	13
A set-partitioning	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	4	4	4	4	5	5	5	5	5	5	5	8	8	8	9	10

Table 3.8: 32-QAM edge profiles.

We conclude that no set-partition labeling can achieve the best possible edge profile labelings of quasi-Gray-coded 32-QAM constellations.

3.3 Application

[20]

Gray-code labeling was proposed in the context of uncoded transmission. For trellis coded modulation where the trellis has parallel transitions, set partitioning is the clear choice. Parallel transitions are often the optimal choice for practical complexity trellis encoders on the AWGN channel even when the search space is widened to include a trellis without parallel transitions.

However, when the best trellis code does not have parallel transitions, the labeling choice is less clear. Such cases include trellis coded modulation for fading channels (where the parallel transitions provide no protection from fading) and constituent encoders for parallel concatenation (where parallel transitions can allow an error event with unit input Hamming distance). In these situations set-partition labeling is not motivated, and edge profile superiority can provide an advantage. We use a code design example to illustrate this point [20].

Consider low rate trellis codes that are robust to severe periodic fading. As shown in [7] [5], robust trellis codes for a specific periodic interleaver can be obtained by maximizing the periodic effective code length (PECL) and the code periodic product distances (CPPDs) with orders ranging from the PECL to the interleaver period. The choice of constellation labeling does not affect the periodic effective code length, but it does affect the periodic product distances.

Consider the example of a rate-1/5 32-QAM trellis code designed to maximize PECL and the CPPDs for a period 6 interleaver using 4 memory elements. The largest possible PECL is 5. We seek a code with PECL 5 for which the CPPDs of order 5 and 6 are as large as possible. Table 3.9 shows the largest values of the sum of the logs of CPPD_5 and CPPD_6 obtained with each of the two labeling paradigms. As expected, the superior edge profile yields better CPPD values.

In Table 3.9 and Table 3.10 below, the generator polynomials of every rate-1/5 feedforward encoder $\mathrm{G}_0, \mathrm{G}_1, \mathrm{G}_2, \mathrm{G}_3, \mathrm{G}_4$ are given in octal representation with the LSB representing the D^0 coefficient of every polynomial. In Table 3.10 below,

	$G_0G_1G_2G_3G_4$	$\log_2 \text{CPPD}_5$	$\log_2 \text{CPPD}_6$	Sum
Set partitioning	06 07 02 30 27	14.8138	11.3219	26.1357
Superior edge profile labeling	11 35 37 27 33	16.0447	13.1674	29.2122
Gain		1.2317	1.8455	3.0765

Table 3.9: Best log sum of order 5 and order 6 code periodic product distance (CPPD) with four memory elements for rate-1/5 32-QAM trellis codes using set partitioning and the superior edge profile labeling of quasi-Gray-coded 32-QAM, G_0, G_1, G_2, G_3, G_4 are the generator polynomials of each rate-1/5 feedforward encoder.

the parity polynomials of each canonical systematic feedback rate-4/5 encoder h_0, h_1, h_2, h_3, h_4 are also given in octal representation with the LSB representing the D^0 coefficient of every polynomial. The set-partitioning 32-QAM labeling used in this section is given below in Fig. 3.26 in octal form. The edge profile superior quasi-Gray-coded 32-QAM labeling used in this section is given in Fig. 3.15 which achieves the best edge profile among all quasi-Gray-coded 32-QAM labelings. Also

	23	26	27	22	
21	10	15	14	11	20
36	17	12	13	16	37
25	04	01	00	05	24
32	03	06	07	02	33
	30	35	34	31	

Figure 3.26: The used set-partitioning 32-QAM labeling.

for the mapping of convolutional encoder outputs to constellation points, the LSB of the 32-QAM constellation label is produced by the leftmost polynomial with significance increasing moving left to right. Note that the rate-4/5 largest free Euclidean distance encoder with 32-QAM set-partitioning labeling has the two

MSBs uncoded.

Table 3.10 shows the largest free Euclidean distance obtained by trellis codes with four memory elements at rates 1/5 and 4/5 produced with each of the two labeling paradigms.

Rate	1/5 ($G_0G_1G_2G_3G_4$)	4/5 (h_0 h_1 h_2 h_3 h_4)
Set partitioning	36 (06 05 02 22 11)	6 (23 04 16 - -)
Superior edge profile labeling	44 (13 17 37 25 23)	4 (21 11 27 23 13)
Gain	0.87 dB	-1.76 dB

Table 3.10: Best free Euclidean distances with four memory elements for rate-1/5 and rate-4/5 32-QAM trellis codes using set partitioning and the superior edge profile labeling of quasi-Gray-coded 32-QAM.

The superior profile provides 0.87 dB more free Euclidean distance than set partitioning for the rate 1/5 trellis code design. But the superior profile provides 1.76 dB *less* Euclidean distance than set partitioning for the rate 4/5 trellis code design. These code design results confirm the earlier statements, i. e., that edge profile superiority provides an advantage when the best trellis code does not have parallel transitions, as with the best Euclidean distance rate-1/5 trellis code with four memory elements in Table 3.10, and set-partition labeling is the better choice when the best trellis code does have parallel transitions, as with the best Euclidean distance rate-4/5 trellis code with four memory elements in Table 3.10. We conjecture that even high rate trellis codes benefit from a superior edge profile when the number of trellis states is large enough that the best trellis code does not employ parallel transitions. Since the performance of bit-interleaved coded modulation depends on both the minimum binary Hamming distance and the minimum Euclidean distance [37], the superior edge profile labeling should be the best choice among all other labelings in 32-QAM bit-interleaved coded modulation.

3.4 Summary

Searches for optimal trellis codes typically fix the constellation labeling (typically a Gray-code or set-partition labeling) and then exhaustively search over possible binary encoders. Specific motivations of Gray-code labeling for uncoded transmission and set-partition labeling in the context of parallel trellis transitions are well known. However, for applications such as trellis coded modulation over channels with fading or puncturing and parallel concatenated codes, there is no compelling reason to use either Gray-code labeling or set-partition labeling. A new labeling paradigm which maximizes the constellation's edge profile produces better trellis codes than the two standard paradigms in these applications.

With a superior underlying structure as the primary consideration, we identify 8-QAM labeling structures that have the best possible edge profile. The 8-QAM constellation can not be Gray-code labeled, but appropriately chosen superior edge profile 8-QAM labelings are strictly equivalent to any set-partitioned 8-QAM labeling. For the 32-QAM cross constellation, we identify six distinct quasi-Gray-coded labeling structures and find that one of them dominates all other five structures in terms of edge profile superiority. No set-partition labeling can achieve the superior edge profile of quasi-Gray-coded 32-QAM labelings.

Set-partition labeling is the best choice when the best trellis has parallel transitions. However, when the best trellis code does not have parallel transitions, edge profile superiority provides an advantage. A code design example illustrates this behavior.

Still, the structure of the best trellis code usually can not be found before the code search. More investigation is needed to determine the choice of labeling paradigms quantitatively, i. e., based on the code rate, the number of encoder's memory elements, etc., we could choose the labeling paradigm for the exhaustive code search to maximize the desired metrics.

Chapter 4

Robustness Analysis to Channel Prediction Errors

In communication systems, adaptivity changes the transmission to match the estimated channel or traffic conditions. From this chapter on, we explore bandwidth-efficient adaptive coded modulation schemes for time-varying channels.

As noted in [38], [39], [40], the distribution of the current channel fading conditioned on the past fading has a Rician distribution. For systems supporting slowly mobile users, adaptive trellis coded modulation (TCM) [38], [39] achieves large bandwidth efficiency gains over their nonadaptive counterparts. However, the systems supporting highly mobile users greatly impede the ability to use uncoded bits for rate adaptation. In this case, adaptive TCM schemes, which employ uncoded bits, fail to be an efficient method of coded modulation.

A solution for adaptive signaling in systems supporting users with high mobility is to employ adaptive bit interleaved coded modulation (BICM) [41]. The nonadaptive BICM has recently been considered for nonadaptive systems operating over the Rayleigh fading channel [51], [37]. In the nonadaptive case, BICM increases the time diversity of systems operating over fading channels by providing an independent fading component for each channel bit out of the convolutional encoder, as opposed to each channel symbol as in standard symbol-interleaved TCM systems [44], [30]. The BICM structure has even greater potential in adaptive coding, as it does not require the use of uncoded bits to adapt constellation size.

An exact mathematical comparison of the adaptive BICM and adaptive TCM structures, although possible, is complicated by the fact that the effective channel changes on a symbol-by-symbol basis, which in turn alters the modulation scheme employed. Thus, the resulting equations are cumbersome and yield little insight into the system operation. However, it was noted in simulations of the TCM scheme that for highly mobile systems, most errors are caused by a single bad prediction, which is analogous to a single bad fade causing errors for a conventional nonadaptive TCM scheme operating over a Rician fading channel. Thus, this chapter analyzes the robustness of BICM and TCM schemes to a single prediction error. This analysis suggests that the code structure of BICM is more suitable for adaptive systems that must support highly mobile users.

4.1 Channel Model

With adaptive coding, if the channel is predicted perfectly at the transmitter, the minimum distance between constellation points remains constant despite the time variation in the fading. As motivated above, suppose that the channel is estimated perfectly except during the transmission of the l^{th} symbol, where a prediction error occurs. This implies that the minimum distance between constellation points remains constant except at the l^{th} transmitted symbol period.

To model this behavior simply for application of a transfer function union bound, we consider a simpler scenario that captures the effect of a failed prediction without the complication of variation in the constellation size. We consider non-adaptive TCM and BICM on an AWGN channel (i.e. $|X(t)| = 1$) except that the lth symbol suffers a fade $Y = |X(lT_s)|$. This is analogous to an adaptive system with perfect channel estimation except for an estimation error on the l^{th} symbol. The information bits encoded to produce the l^{th} transmitted symbol suffer the most from this prediction error. Thus, our modified transfer function union bound estimates the bit error rate only for those bits.

4.2 Transfer Function Bound on the BER for TCM and BICM under a Single Prediction Error in an AWGN Channel

For a trellis code, the standard transfer function is

$$T(W, I) = \sum_{w} \sum_{i} N(w, i) W^{w} I^{i} \tag{4.1}$$

where w is the Euclidean distance of an error event, i is the number of information bit errors associated with that error event, and $N(w, i)$ is the expected number of error events beginning at a fixed symbol that cause i information bit errors and have Euclidean distance w. The expectation is over the possible transmitted information sequences. The transfer function $T(W, I)$ can be computed by summing the path metrics in an appropriate state diagram of all paths with a nonzero exponent of I that begin and end in the zero error-state.

The technique of computing the transfer function to provide union bounds on bit error rate (BER) for trellis codes has been widely investigated [45], [46], [47], [48]. The transfer function can be found using various state transition diagrams. For a trellis code with ν_e memory elements, Biglieri [45] described a general algorithm using a $2^{2\nu_e}$-state transition diagram. Rouanne & Costello [46] and Zehavi & Wolf [47] demonstrated that a 2^{ν_e}-state transition diagram is sufficient for quasi-regular codes. Wesel [48] proposed to compute the transfer function of any trellis code using a $2^{\nu_e+\nu_q}$-state transition diagram where ν_q is an integer between zero and ν_e. Kucukyavuz & Fitz proposed an alternative state reduction technique in [50]. For simplicity and generality, we use Biglieri's product-state approach [45] to investigate the robustness of BICM and TCM schemes under a single channel fade (prediction error). However, the application of the state reduction techniques discussed above is straightforward.

Using the bound $Q(t) \leq \frac{e^{-\frac{t^2}{2}}}{2}$, the transfer function of a rate-k/n trellis code can provide the well-known union bound on the bit error rate (BER) for an AWGN channel as:

$$P_e \leq \frac{1}{2k} \frac{\partial T(W,I)}{\partial I}\bigg|_{I=1, W=e^{-1/4N_0}}, \tag{4.2}$$

where $T(W,I)$ is the standard transfer function (4.1) and $\frac{N_0}{2}$ is the variance per dimension of the AWGN. Tighter bounds on $Q(\cdot)$ can be applied to (4.2) if the free distance of the code including fading effects is known [47].

Biglieri [45] used the $2^{2\nu_e}$-state transition diagram to compute $T(W,I)$ for larger constellations by employing the state pair (encoder state, incorrect encoder state) (say $(\sigma, \tilde{\sigma})$). Call the $N_g = 2^{\nu_e}$ product states where $\sigma = \tilde{\sigma}$ the "good product states" and the remaining $N_b = 2^{2\nu_e} - 2^{\nu_e}$ product states the "bad product states".

Define $\mathbf{A}$ to be the $N_b \times N_b$ matrix of labels of the form $W^w I^i$ for transitions from one bad product state to another. Here w is the Euclidean distance between the correct and incorrect constellation points produced in the product state transition, and i is the number of information bit errors associated with that transition. Define $\mathbf{b}$ to be the $N_b \times N_g$ "tall" matrix of labels of the form $W^w I^i$ for transitions from a good product state to a bad product state and $\mathbf{c}$ to be the $N_g \times N_b$ "wide" matrix of labels of the form $W^w I^i$ for transitions from a bad product state to a good product state, and $\mathbf{d}$ be the $N_g \times N_g$ matrix of labels of the form $W^w I^i$ for transitions from one good product state to another. Then the equation for the transfer function T is:

$$T = \frac{1}{2^{\nu_e}} \left(\mathbf{1^T d 1} + \sum_{\mathbf{j=0}}^{\infty} \mathbf{1^T c A^j b 1} \right) = \frac{\mathbf{1}}{\mathbf{2^{\nu_e}}} \left(\mathbf{1^T d 1} + \mathbf{1^T c (I - A)^{-1} b 1} \right), \tag{4.3}$$

where $\mathbf{1}$ is the N_g-element column-vector of ones and $\mathbf{I}$ is the $N_b \times N_b$ identity matrix.

In this case, the derivative $\partial T(W, I)/\partial I$ in (4.2) is

$$\frac{\partial T}{\partial I} = \frac{1}{2^{\nu_e}} \left(\mathbf{1^T d' 1} + \mathbf{1^T c (I - A)^{-1} b' 1} + \mathbf{1^T c' (I - A)^{-1} b 1} + \mathbf{1^T c (I - A)^{-1} A' (I - A)^{-1} b 1} \right), \tag{4.4}$$

where $\mathbf{d}'$, $\mathbf{b}'$, $\mathbf{c}'$ and $\mathbf{A}'$ indicate the element-by-element partial derivative with respect to I. Each term in the derivative of (4.4) counts information bit errors on only one trellis branch. This derivative effectively fixes attention on one trellis branch, counting the information bit errors occurring on that branch caused by all error events. For example, $\mathbf{1^T c (I - A)^{-1} b' 1}$ counts information bit errors for error events that begin with the trellis branch of interest.

To incorporate the effect of a single fade Y (i.e. the result of an incorrect prediction) in an AWGN channel for the trellis branch of interest at the transmitted l^{th} symbol period time, replace $\mathbf{d}'$, $\mathbf{b}'$, $\mathbf{c}'$ and $\mathbf{A}'$ in (4.4) with $\mathbf{d'_Y}$, $\mathbf{b'_Y}$,

$\mathbf{c}'_{\mathbf{Y}}$ and $\mathbf{A}'_{\mathbf{Y}}$ respectively, where in every element involving W in $\mathbf{d}'_{\mathbf{Y}}$, $\mathbf{b}'_{\mathbf{Y}}$, $\mathbf{c}'_{\mathbf{Y}}$ and $\mathbf{A}'_{\mathbf{Y}}$, the exponent of W is multiplied by the variable scale factor Y^2 instead of 1 ($|X(t)| = 1$ in an AWGN channel). For example, W^2 is replaced by W^{2Y^2}. Thus for a trellis code under a single prediction error, the derivative in (4.2) becomes

$$\frac{\partial T}{\partial I} = \frac{1}{2^{\nu_e}}\left(\mathbf{1}^{\mathbf{T}}\mathbf{d}'_{\mathbf{Y}}\mathbf{1} + \mathbf{1}^{\mathbf{T}}\mathbf{c}(\mathbf{I}-\mathbf{A})^{-1}\mathbf{b}'_{\mathbf{Y}}\mathbf{1} + \mathbf{1}^{\mathbf{T}}\mathbf{c}'_{\mathbf{Y}}(\mathbf{I}-\mathbf{A})^{-1}\mathbf{b}\mathbf{1} + \mathbf{1}^{\mathbf{T}}\mathbf{c}(\mathbf{I}-\mathbf{A})^{-1}\mathbf{A}'_{\mathbf{Y}}(\mathbf{I}-\mathbf{A})^{-1}\mathbf{b}\mathbf{1}\right), \tag{4.5}$$

Therefore, by using (4.2) and (4.5), the transfer function bound on BER for TCM codes under a single prediction error can be obtained.

Similarly, for a rate-k/n BICM code, the transfer function T is [51]

$$T(W_1, \ldots, W_n, I) = \sum_{w_1 \ldots w_n} \sum_i N(w_1, \ldots, w_n, i) I^i \prod_{j=1}^{n} W_j^{w_j}, \tag{4.6}$$

where $N(w_1, w_2, \ldots, w_n, i)$ is the number of error events with Hamming weight w_j associated with the jth most significant codeword bit before the interleaver, and i is the number of information bit errors associated with those error events.

Based on Caire et al's error-probability analysis for a single interleaver processing all bits [37], the union bound on BER for a rate-k/n BICM code over an AWGN channel is computed as

$$P_e \leq \frac{1}{2k} \frac{\partial T(W_1, \ldots, W_n, I)}{\partial I}\bigg|_{I=1, W_j = N_{min}(1) e^{-d_{min}^2/4N_0}}, \tag{4.7}$$

where $N_{min}(1)$ is a constant value and depends only on the constellation set and on the labeling [37], and d_{min} is the minimum Euclidean distance of the constellation set.

Similar to the discussion for the transfer function bound on BER for a trellis code under a single fade Y, the representation of the transfer function bound on BER for a rate-k/n BICM code under a single fade can be obtained from (4.7) and (4.5). Note that for a BICM code, the incorrect prediction Y affects only one coded bit in each of n error events since every coded bit is interleaved by the ideal interleaver. Here the average case is considered; in other words, the fade Y is equally likely to affect each coded bit position in the trellis branch of interest in the associated error events. Specifically in $\mathbf{A}'$, $\mathbf{b}'$, $\mathbf{c}'$ and $\mathbf{d}'$, replace $\prod_{j=1}^{n} W_j^{w_j}$ with $\frac{1}{n}\sum_{p=1}^{n}(\prod_{j=1,j\neq p}^{n} W_j^{w_j}) W_p^{w_p Y^2}$ to produce the $\mathbf{A}'_{\mathbf{Y}}$, $\mathbf{b}'_{\mathbf{Y}}$, $\mathbf{c}'_{\mathbf{Y}}$ and $\mathbf{d}'_{\mathbf{Y}}$ for application of(4.5) to the BICM case.

Figure 4.1 shows the result of applying these two modified transfer function bounds to TCM and BICM respectively, each with a single fade Y at the l^{th} transmitted symbol period. The figure compares a rate-1/2, 4-state, 4-PSK TCM with a rate-1/2, 4-state, 4-PSK BICM. These two codes were selected because they provide the same performance on the AWGN channel. Both codes have a BER of 4×10^{-5} at 6 dB, which is the SNR examined in Fig. 4.1.

To interpret Figure 4.1 in the context of adaptive coding, consider the estimate for the lth symbol to have been normalized to 1. Y^2 (the x-axis) is the square of the actual fading value (given in dB). When the dB-value of Y^2 is zero, the estimate is perfectly correct. When the dB-value of Y^2 is negative, the estimate is optimistic. This is the situation where adaptive-TCM is known to fail, and Figure 4.1 illustrates how BICM provides significantly better performance (more robustness) in this region. The reason for the better performance of BICM in this region is that TCM sees the full effect of the small value of Y^2 on a single trellis branch while BICM spreads the effect of this prediction error over n well-separated trellis branches.

When the dB-value of Y^2 is positive, the estimate is pessimistic. In this region, TCM has a slight advantage over BICM. The explanation of the TCM advantage in this region is actually the same as the reason for its poor performance for the

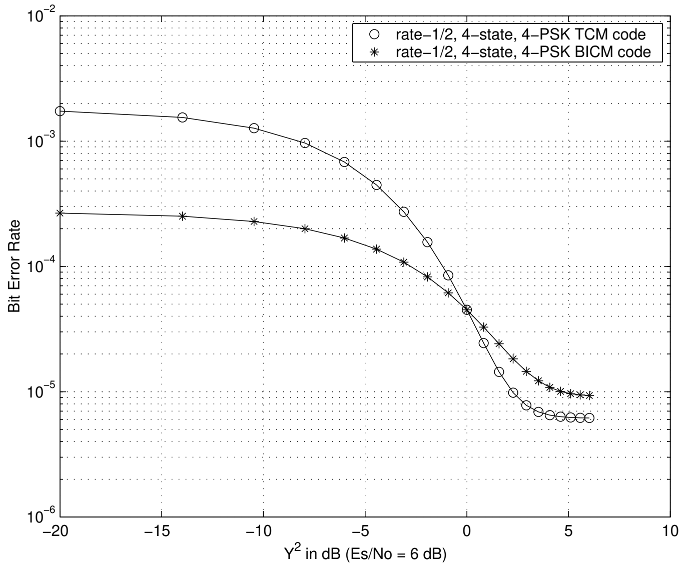


Figure 4.1: Transfer function union bounds of bit error rate at the faded symbol for a rate-1/2, 4-state, 4-PSK BICM code and a rate-1/2, 4-state, 4-PSK TCM code under a single fade Y(or prediction error). SNR for non-faded symbols is 6 dB.

optimistic prediction error. In this situation the pessimistic prediction causes one constellation to have a larger-than-expected minimum distance. With TCM the full effect of this larger distance is seen on a single trellis branch. For BICM, this effect is spread over n well-separated branches, causing a smaller decrease in BER at each branch. Hence BICM weakens the effect of prediction errors whether that effect is detrimental or beneficial to BER.

Chapter 5

Adaptive Coded Modulation Schemes For Time-Varying Channels

One possible method to achieve robust and spectrally efficient communication over multipath channels is to adapt the transmission scheme to the current channel characteristics using channel estimates available at the transmitter. Unlike nonadaptive schemes, which are designed for worst-case channel conditions to achieve acceptable performance, adaptive signaling methods take advantage of favorable channel conditions by allocating power and rate efficiently.

In recent years, numerous approaches have been proposed to optimize the transmitted signal using knowledge of the current channel fading value of a frequency-nonselective Rayleigh fading channel [52], [53], [54], [55], [39] . If the current fading value can be predicted perfectly at the transmitter, the channel is effectively an additive white Gaussian noise (AWGN) channel with a time-varying, but known signal-to-noise ratio (SNR). Thus, these schemes have employed set-partitioned trellis coded modulation (TCM) and have adapted the number of uncoded bits on a given symbol based on the channel estimates.

However, as noted in [39], when only past fading is known, the distribution of the current channel fading conditioned on the past is Rician. For systems supporting highly-mobile users, the effective channel conditioned on past fading can become nearly Rayleigh. This impedes the use of uncoded bits for rate adaptation. Therefore adaptive TCM schemes that employ uncoded bits fail to be efficient. Error events from poor predictions dominate the bit error probability in such an adaptive system in the same way that error events from bad fades dominate the bit error probability of nonadaptive wireless communication systems.

A solution for adaptive signaling in systems supporting users with high mobility is to employ adaptive bit interleaved coded modulation (BICM), which does not use uncoded bits to adapt constellations [41]. In the nonadaptive case, BICM increases the time diversity of systems operating over fading channels by providing an independent fading component for each channel bit out of the convolutional encoder [51], [37], as opposed to each channel symbol as in standard symbol-interleaved TCM systems [30], [29]. The inherent diversity of adaptive BICM protects every information bit from channel prediction errors. However, adaptive BICM uses a bit interleaver, which results in long latency in addition to the implementation complexity of the bit interleaver. This is incompatible with applications requiring very low latency transmissions, such as in speech transmissions.

Note that it is in low-latency applications that adaptive signaling is likely to be most applicable. Applications that can tolerate large latencies can employ interleaving to arrive at an equivalent channel that experiences approximately independent and identically distributed (IID) fading from symbol to symbol. For the IID Rayleigh fading channel, the capacity of an adaptive signaling system is only slightly better than that of a non-adaptive system [42]; thus, as iteratively-decoded codes approach capacity, there will be limited gain through transmitter adaptation to the multipath fading values for large-latency applications. However, for applications with a low-latency requirement, a more appropriate information-theoretic measure is the delay-limited capacity, which has been shown to be significantly larger for systems employing transmitter adaptation than for their non-adaptive counterparts [43]. This tenet was further supported for practical systems in [57].

In this chapter, two adaptive coded modulation (ACM) schemes are proposed that avoid both uncoded bits and the use of interleaving. One is modified from [39], [41] and is called stochastic ACM. The other scheme directly estimates the bit error probability P_b by employing a forward trellis search (FTS) algorithm and ensures that the estimated bit error probability $\hat{P}_b$ is less than the desired bit error probability P_d. This scheme is called FTS-ACM. FTS-ACM outperforms the stochastic ACM and is particularly attractive for users that require very low latency transmission.

Section 5.1 reviews adaptive signaling over time-varying channels. Section 5.2 presents the new ACM schemes. Section 5.3 gives numerical results for the two proposed ACM schemes over fading channels, and compares FTS-ACM with adaptive BICM. Two methods, termed FTS-ACM1 and FTS-ACM2, are presented for adapting the transmitted power. Under the same system performance target, these comparisons show that FTS-ACM achieves a higher bandwidth efficiency gain than the stochastic ACM and that FTS-ACM achieves a bandwidth efficiency that is equal or better than adaptive BICM. FTS-ACM comes with the benefit of low-latency but at the cost of increased complexity at the transmitter for the FTS operation. Section 5.4 presents the conclusions.

5.1 System Models and Characterization

5.1.1 System Model

The baseband system model, which is identical to that of [39], is illustrated in Figure 5.1.

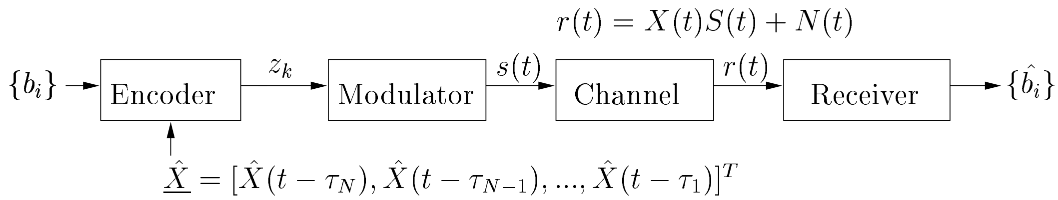


Figure 5.1: System Model

An independent and identically distributed sequence of information bits $\{b_i\}$, each equally likely to be 0 or 1, is assumed. The transmitted signal is given by $s(t) = \sum_{k=-\infty}^{\infty} z_k p(t-kT_s)$, where z_k is the k^{th} complex coded data symbol, $p(t)$ is a pulse shape that results in no intersymbol interference in the samples (spaced at T_s) of the output of the matched filter at the receiver, and $\frac{1}{T_s}$ is the symbol rate. The additive noise $n(t)$ is white Gaussian noise with two-sided power spectral density $\frac{N_0}{2}$.

A frequency-nonselective channel is considered in this work; the fading is modeled as a complex multiplier $X(t)$. The Gaussian wide-sense stationary uncorrelated scattering fading model [56] is assumed, where the independent in-phase and quadrature components of $X(t)$ are zero-mean Gaussian processes with autocorrelation function $R_X(\tau)$. Hence, the channel is modeled as a Rayleigh fading channel. It is also assumed that $X(t)$ varies slowly enough so that it can be considered as constant over a single symbol period.

As shown in Figure 5.1, the vector of past channel fading estimates, $\hat{\underline{X}} = \hat{\underline{X}}_R + j\hat{\underline{X}}_I$, is assumed to be available at the transmitter. It will be assumed that the past fading estimates are made perfectly; that is, $\hat{X}(kT_s - \tau_i) = X(kT_s - \tau_i)$, $\tau_i = iT_s$, $i = 1, \cdots, N$, and thus $\hat{\underline{X}} = \underline{X}$, although the consideration of noisy estimates is conceptually identical [57]. We assume no knowledge of the current fading beyond what can be inferred from the known past fading. Throughout this work, coherent reception with perfect channel state information (CSI) at the receiver is assumed.

5.1.2 Characterization of the Channel Variation

Adaptive signaling attempts to optimize the signal set using perfect or imperfect knowledge of the current fading values at the transmitter. Suppose the choice of the signal set for the k^{th} symbol is being considered at the transmitter. Let $Y_k = |X(kT_s)|$ be the amplitude of the fading at the k^{th} signal period. As shown in [39], Y_k is Rician when conditioned on $\underline{X}$ with conditional probability density function given by

$$p_{Y_k|\underline{X}}(y_k|\underline{x}) = \frac{y_k}{\sigma_{Y_k}^2} e^{-\frac{y_k^2+s^2}{2\sigma_{Y_k}^2}} I_0\left(\frac{y_k s}{\sigma_{Y_k}^2}\right) \qquad y_k \geq 0, \tag{5.1}$$

where $I_0(\cdot)$ is the 0^{th}-order modified Bessel function. Throughout this work, we compute the probability in (5.1) using only the most recent previous fading value. We normalize to achieve $E[|X(kT_s)|^2] = 1$ such that $E[(X_R(kT_s))^2] = E[(X_I(kT_s))^2] = \frac{1}{2}$. By defining the correlation coefficient $\rho = R_X(\tau_1) = R_X(T_s)$, one obtains $s^2 = |x|^2\rho^2$ and $\sigma_{Y_k}^2 = \frac{1-\rho^2}{2}$, where $x = X(kT_s - \tau_1) = X(kT_s - T_s)$.

Define the specular-to-diffuse component (Rician) factor of (1) by $K = \frac{s^2}{\sigma_{Y_k}^2} = \frac{2|x|^2\rho^2}{(1-\rho^2)}$. As the mobility of system users increases (i.e ρ decreases), K drops rapidly for any given estimate. Thus, in a system supporting highly mobile users, a large value of K is highly unlikely and thus adaptive TCM schemes, which employ uncoded bits for rate adaptation, fail to be an efficient method of coded modulation.

5.2 Adaptive Coded Modulation Schemes over Time-Varying Channels

5.2.1 General Structure for Adaptive Coded Modulations

The general structure of the new adaptive coded modulations is shown in Figure 5.2. The output coded sequence of the convolutional encoder is passed to the

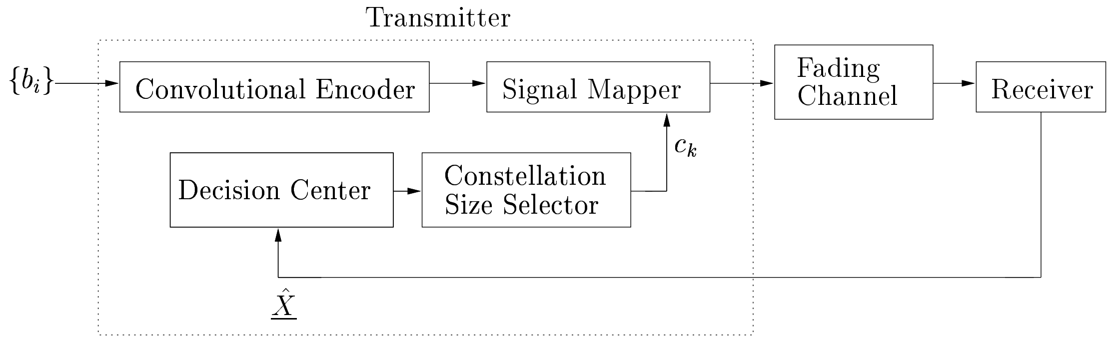


Figure 5.2: General structure for Adaptive Coded Modulations

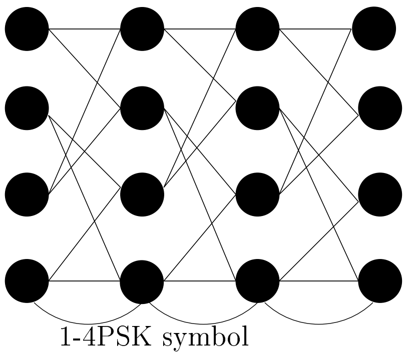
signal mapper. The signal mapper groups the output coded bits and then maps one group into one symbol of constellation c_k whose size at the k^{th} symbol interval is decided by the constellation size selector. The constellation c_k is selected from the allowable signal sets $\zeta_1, \zeta_2, \cdots, \zeta_L$ based on the known past channel fading. The signal constellation ζ_{l+1} has a size larger than that of ζ_l. Let M_k denote the number of symbol points in the chosen constellation c_k. Then, $\log_2 M_k$ bits are grouped together and mapped into one symbol of c_k.

This new adaptive transmitter generates a different information bit rate for each symbol by simply concatenating several adjacent trellis branches into one branch in the trellis. This process is very similar to the generation of multiple trellis coded modulation (MTCM)[14]. The only difference is that MTCM transmits more than one channel symbol from a fixed constellation per trellis branch while this new adaptive transmitter transmits a single channel symbol from a larger constellation. Figure 5.3 gives an example to illustrate the variation of data rate and constellation size using a rate-1/2, 4-state, 4-PSK convolutional encoder.

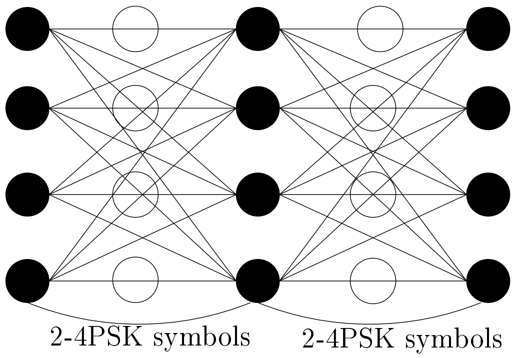
Figure 5.3(a) is the rate-1/2, 4-state regular trellis, every branch transmits 1 bit per 4-PSK symbol. Figure 5.3(b) is the trellis of the rate-2/4, 4-state multiple trellis code, every branch transmits 2 bits per 2 4-PSK symbols. Figure 5.3(c) is an example section of our ACM trellis: the first branch transmits 1 bit per 4-PSK, and the second branch transmits 2 bits per 16-QAM symbol by simply concatenating 2 adjacent trellis branches together. Similarly, by concatenating more adjacent trellis branches together, we can transmit 3 bits per 64-QAM symbol (rate-3/6) and 4 bits per 256-QAM symbol (rate-4/8).

Finally, the mapped signal is transmitted over the time-varying channel. Note that no interleaving is used and no uncoded information bits are sent through the channel. The metric generation of this adaptive scheme for a given signal set is the same as a regular trellis coded modulation. At the receiver, the Viterbi algorithm [58] is used to decode the received signals.

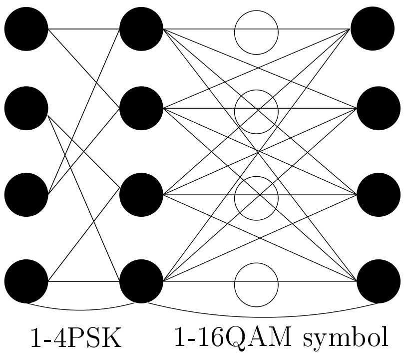
In this general structure, there are two design issues that must be considered: how to choose the convolutional encoder, and how to choose the signal set c_k based on past fading. The first issue is a multi-criterion optimization problem and not



(a)



(b)



(c)

Figure 5.3: Variation of data rate and constellation in the transmitter.

easily addressed. For a fixed code rate and constraint length, we simply chose the convolutional code with maximal free Hamming distance.

For the second issue, two types of decision algorithms are proposed. One is modified from [39], [41] and essentially uses the conditional distribution of the current channel fade to decide the constellation size. The other decision algorithm utilizes knowledge of past fading as well as the conditional distribution of the current channel fade to evaluate overall distances of error events and then to choose the constellation size accordingly. This adaptive decision algorithm employs a forward trellis search algorithm (similar to the Viterbi algorithm [58]) to compute accumulated distances at every trellis state.

5.2.2 Design Rules

For the general structure, a nominal rate-1/2, 4-PSK (i. e., ζ_0 = 4-PSK) trellis coded modulation scheme is assumed that provides P_d at some signal-to-noise ratio $(\mathrm{SNR})_0$ on a Rayleigh fading channel. The same trellis structure of the nominal scheme is used for the adaptive design. When the decision center selects a larger signal constellation ζ_l, a trellis code with higher bit rate per signal is generated by simply concatenating adjacent trellis branches into one branch of a higher rate trellis code.

Stochastic ACM

Let u_l be the minimum normalized squared Euclidean distance between any two points in the signal constellation c_l. As in [39], [41], the goal of our stochastic ACM is to stochastically maintain u_l. With $h = |X(kT_s - \tau_1)| = |X(kT_s - T_s)|$, we choose c_l to be the largest that satisfies

$$\int_0^\infty e^{-y_k^2 u_l \frac{E_s}{4N_0}} p_{Y_k|h}(y_k|h) dy_k \le D_0(0, u_0 E_s)\,. \tag{5.2}$$

This approximately maintains the bit error probability of the nominal scheme. E_s is the average transmitted energy. Here, $D_0(x, v)$ is the Bhattacharyya bound [59] on the error probability of choosing x when v is correct for the nominal code. For example, $D_0(x, v) = e^{-\frac{d_E^2(x,v)}{4E_s}(\mathrm{SNR})_0}$ if the nominal code is designed for the AWGN channel and $D_0(x, v) = \frac{1}{1+\frac{d_E^2(x,v)}{4E_s}(\mathrm{SNR})_0}$ if the nominal code is designed for a Rayleigh channel. For a Rician fading channel, $D_0(x, v)$ is a mixture of the above two special cases. In this chapter, we adopt the design for the worst case, i. e., the case for a Rayleigh channel.

The evaluation of the left side of (5.2) in terms of h, $\frac{E_s}{N_0}$, and ρ is given in [39]. Making c_l guarantee formula (5.2) yields h_l, the threshold such that for $h \ge h_l$, ζ_l can be employed. In the case that $h < h_0$, this symbol period is outage, i. e., no information bits can be transmitted through the channel during this symbol period.

We employ energy adaptation as in [39], [41] to overcome the problem caused by the fact that for all h such that $h_l < h < h_{l+1}$, the estimate is more favorable than that required to use ζ_l but not favorable enough to use ζ_{l+1}. First, a signal set is chosen according to the above given design method with no energy adaptation. Then (5.2) is used to decide the minimum energy required to maintain D_0 given the channel estimate h. This minimum energy is then employed for the current symbol, and the remaining energy is saved for the next symbol. This algorithm is used for the simulations of the stochastic ACM.

FTS-ACM

Stochastic ACM utilizes the distance accumulated on only the current signal period to select the constellation. To obtain better performance, we propose a new design rule which employs knowledge of past fading as well as the conditional distribution of the current channel fade to evaluate overall distances of error events and then chooses the constellation accordingly. This new adaptive scheme estimates the bit error probability P_b and ensures that this estimate is below the desired bit error probability P_d.

Exact calculation for P_b is formidable. Thus, we resort to a standard union bound. The standard union bound involves error events ending in the current signal period and error events starting from the current signal period as well as error events transiting through the current signal period. Those error events that do not end in the current signal period are affected by the future channel fading. Since we do not know the future channel fading, we consider only the error events ending in the current signal period (i. e., ending at the k^{th} symbol period). Those error events involving future channel fading are considered when selecting future constellations.

To make the calculation even simpler, at the $(k-1)^{st}$ signal period, we only keep the worst error event, i. e., the error event with the minimum accumulated distance, at every trellis state; then we sum only the error events which end at the k^{th} signal period. Therefore we obtain

$$\hat{P}_b|Y_k \approx \sum_{j\in A} \frac{b_j}{k'_j} Q\left(\sqrt{\frac{d^2_{k-1,j} + |y_k|^2 \delta^2_j(c_k)}{2N_o}}\right), \tag{5.3}$$

where $\hat{P}_b|Y_k$ is the estimated bit error probability conditioned on the k^{th} fading Y_k.

The set A contains those states that have a branch connecting to the zero state in the trellis for c_k. In Figure 5.3(c), A={0,2} for c_k = 4-PSK and A={0,1,2,3} for c_k = 16-QAM. $d^2_{k-1,j}$ is the minimum squared accumulated Euclidean distance of an error event (ending in the k^{th} signal period) up to the $(k-1)^{st}$ signal period at state j, k'_j is the average transmitted information bit rate of this error event, b_j is the total number of information bit errors associated with this error event, y_k is the fading estimate at the k^{th} signal period, $\delta^2_j(c_k)$ is the part of the squared Euclidean distance of this error event incurred by transiting from state j back to the zero state using constellation c_k at the k^{th} signal period.

In (5.3), based on the assumption that the past fading estimates are made perfectly, $d^2_{k-1,j}$ can be precisely computed by using a forward trellis search algorithm where error events can start from any signal period. For the k^{th} signal period, FTS-ACM selects the largest candidate constellation for which the computed $E_{Y_k}[\hat{P}_b|Y_k]$ at the k^{th} signal period is less than P_d.

Utilizing the fact that Y_k is Rician when conditioned on $\underline{X}$ as shown in (5.1), where $x = X(kT_s - \tau_1) = X(kT_s - T_s)$, FTS-ACM selects the largest candidate constellation whose resulting $E_{Y_k}[\hat{P}_b|Y_k]$ is less than P_d;

$$\sum_{j \in A} \frac{b_j}{k'_j} \int_0^{\infty} Q\left(\sqrt{\frac{d^2_{k-1,j} + |y_k|^2 \delta^2_j(c_k)}{2N_o}}\right) P(y_k|x)\, dy_k < P_d. \tag{5.4}$$

When evaluating the left side of (5.4), we employ an alternate definite integral form for the Gaussian Q-function [60] as $Q(x) = \frac{1}{\pi}\int_0^{\pi/2} \exp(-\frac{x^2}{2sin^2\theta})d\theta$ for $x \geq 0$. After this modification for the Q function, we utilize the integral table from [61] (6.614.2); then, the left side of (5.4) is simplified into a definite integral from 0 to $\pi/2$ with respect to θ which can be numerically computed; thus, we obtain (5.5),

$$\frac{1}{\pi}\exp(-\frac{|x|^2\rho^2}{1-\rho^2})\sum_{j\in A}\frac{b_j}{k'_j}\int_0^{\pi/2}\exp(-\frac{d^2_{k-1,j}}{4N_o sin^2\theta})\frac{\exp(\frac{|x|^2\rho^2/(1-\rho^2)}{\delta_j^2(c_k)(1-\rho^2)/(4N_o sin^2\theta)+1})}{\delta_j^2(c_k)(1-\rho^2)/(4N_o sin^2\theta)+1}d\theta < P_d. \tag{5.5}$$

In contrast to the stochastic ACM, FTS-ACM does not require energy adaptation when data is transmitted. However, no power is transmitted during outage periods, and this results in a reduced average transmit power. To solve this problem, we deal with the transmit power in two different ways. The first way (FTS-ACM1) is to simply save this non-transmitted power for the next transmitted symbol. But this leads to a time-varying transmission power and a lack of peak energy control, thus it is inefficient under the peak energy constraint. Therefore we propose a second way (FTS-ACM2) to handle the transmit power, which is to raise the fixed transmit power so that the desired average transmit power is achieved when the outage periods are considered. Those two kinds of the transmit power processing are compared in the simulations of FTS-ACM.

5.3 Numerical Results and Analysis

We use the standard rate-1/2, 64-state convolutional code as the nominal code. This code achieves $P_d = 10^{-5}$ at average received SNR $(SNR)_0 = 11.5$ dB over a Rayleigh fading channel. As discussed in section 5.2.1, the transmitter varies the per-symbol bit rate by concatenating adjacent trellis branches into one branch. We use rate-2/4, rate-3/6, rate-4/8 codes by concatenating 2 or 3 or 4 adjacent trellis branches together. Note that these trellis codes do not have parallel transition since a 64-state nominal code is used. In this chapter, we adopt 4-PSK, 16-QAM, 64-QAM, 256-QAM as candidate signal sets.

	rate-1/2	rate-2/4	rate-3/6	rate-4/8
Gray-labeled	20.00	4.00	1.33	0.47
set-partitioning	14.00	4.00	0.95	0.33

Table 5.1: Normalized d^2_{free} of the standard rate-1/2, 64-state trellis codes using Gray-labeled or set-partitioning labeled.

Using forward trellis search, Table 5.1 shows the normalized free distances for these trellis codes. We compare symmetric ultracomposite (SU) Gray-labeled and set-partitioning-labeled constellations [62]. Since SU Gray-labeled constellations achieve better normalized free distances for our encoder than set-partitioning-labeled constellations, all candidate constellations are symmetric ultracomposite Gray-labeled.

The time-varying channel used in our simulations is generated by filtering the independent in-phase and quadrature zero-mean Gaussian processes with a filter having the Doppler spectrum as in (5.6) [63],

$$S(f) = \frac{1}{\pi f_m} \frac{1}{\sqrt{1-(f/f_m)^2}}\,, \tag{5.6}$$

where f_m is the maximum Doppler frequency. The autocorrelation function is represented as $R_x(\tau) = J_0(2\pi f_m \tau)$ where $J_0(\cdot)$ is the 0^{th}-order Bessel function of the first kind. Thus the simulated channel model is temporally correlated and frequency-nonselective.

In Section 5.2.2, two methods to deal with the transmit power of FTS-ACM are proposed. Figure 5.4 compares FTS-ACM1 and FTS-ACM2 with $P_d = 10^{-5}$. We observe that FTS-ACM2 gives better rates as well as a lower peak power than FTS-ACM1. Therefore in the rest of this section, we employ FTS-ACM2 to compare with other schemes.

We first considered the stochastic ACM without threshold-adjustment. FTS-ACM2 achieves much better rate performance than this stochastic ACM, but this stochastic ACM obtains lower bit error probability than FTS-ACM2 for $\rho = 0.9$ and $\rho = 0.7$ where $\rho = J_0(2\pi f_m T_s)$. To make a fair comparison between them, we intentionally lower the stochastic ACM's thresholds so that both the stochastic ACM and FTS-ACM2 have similar bit error probability (both below P_d). Figure 5.5 shows that FTS-ACM still achieves rates about 0.4 bits higher than the threshold-adjusted stochastic ACM for $\rho = 0.9$ and $\rho = 0.7$.

We also compare FTS-ACM2 with the threshold-adjusted adaptive BICM. Figure 5.6 displays the comparison results. For similar P_b (both below P_d), for $\rho = 0.7$, FTS-ACM achieves rates about 0.2 bits per symbol higher than adaptive BICM; for $\rho = 0.9$, they achieve very similar rates. Note that adding the effect of threshold-adjustment to stochastic ACM and adaptive BICM must be done empirically. In contrast, the P_b estimate of the left side of the (5.4) is accurate enough that no empirical adjustment of FTS-ACM2 is required to achieve the performance in Figure 5.6.

5.4 Summary

Although adaptive TCM systems demonstrate large bandwidth efficiency gains over their nonadaptive counterparts in wireless systems which support users with low mobility, their rates plummet when the mobility of the users increases. In channels which support highly mobile users, the recently proposed adaptive BICM achieves a moderate bandwidth efficiency gain over previously proposed adaptive schemes and nonadaptive schemes of similar complexity. However, adaptive BICM

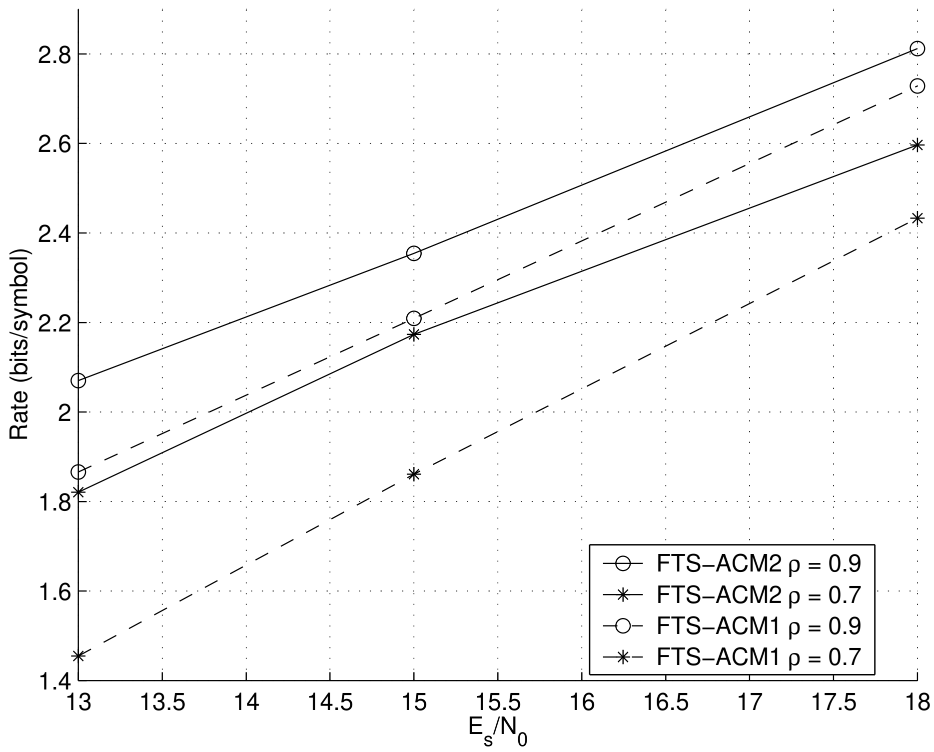


Figure 5.4: Simulation results comparison between FTS-ACM1 and FTS-ACM2 for $\rho = 0.9$ and $\rho = 0.7$ respectively.

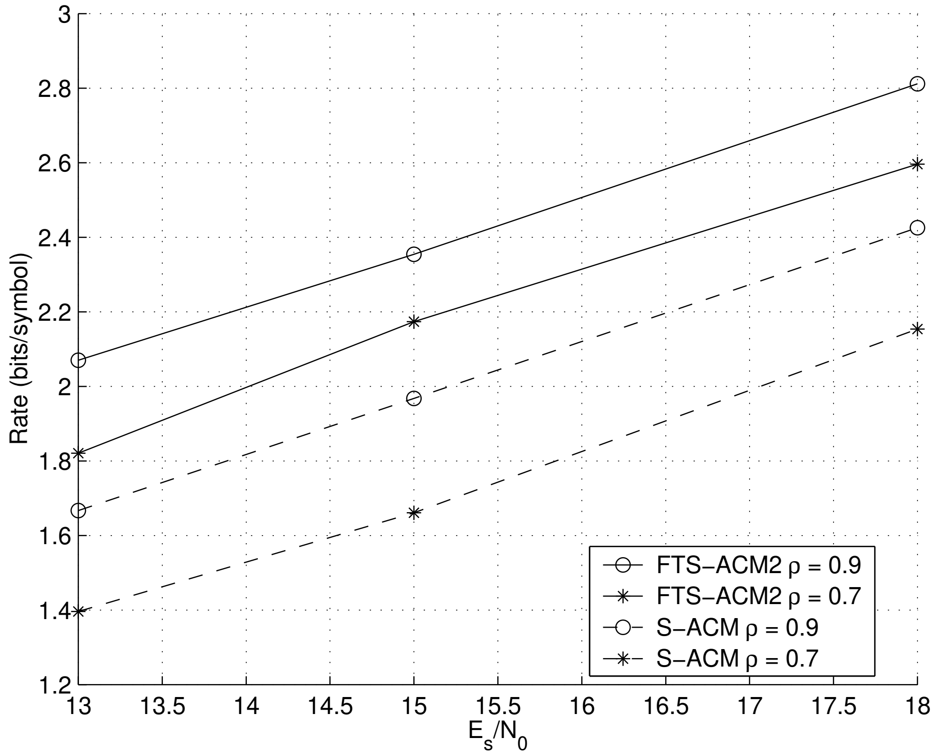


Figure 5.5: Simulation results comparison between FTS-ACM2 and the threshold-adjusted stochastic ACM for $\rho = 0.9$ and $\rho = 0.7$ respectively.

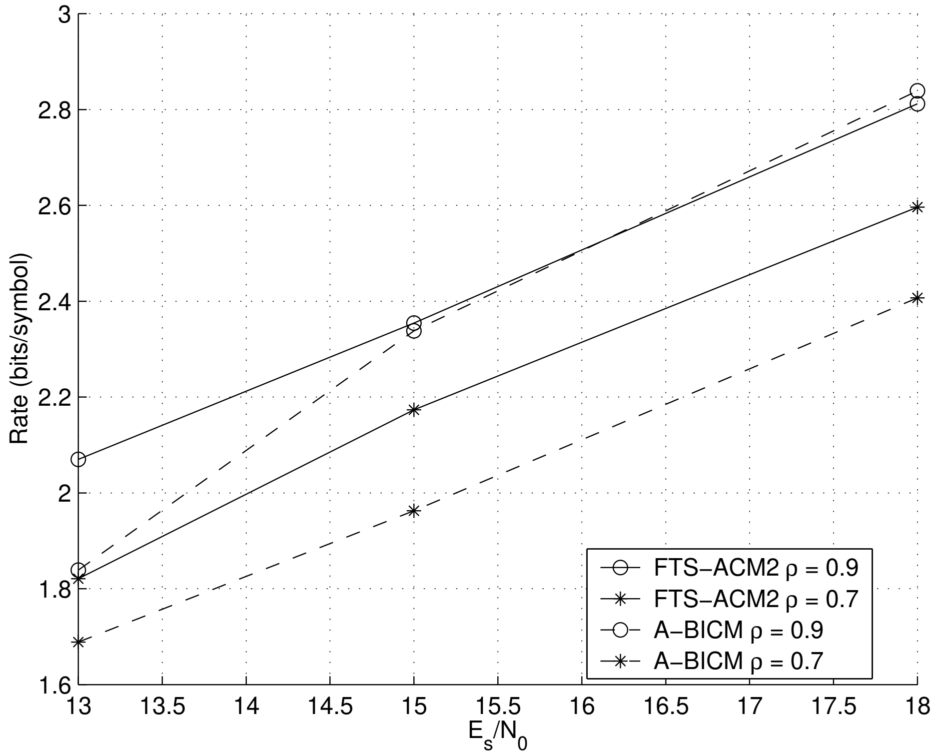


Figure 5.6: Simulation results comparison between FTS-ACM2 and the threshold-adjusted adaptive BICM for $\rho = 0.9$ and $\rho = 0.7$ respectively.

employs a bit interleaver and thus leads to long latency; therefore, it is not suitable for applications requiring very low latency.

We propose two adaptive coded modulation schemes, stochastic ACM and FTS-ACM, which avoid both uncoded bits and the use of interleaving. Stochastic ACM is motivated by work in [39], [41]. FTS-ACM employs a forward trellis search algorithm to adapt to the channel conditions. The numerical results show that FTS-ACM with a fixed transmitted power achieves a better bandwidth efficiency than the stochastic ACM and a better or equal bandwidth efficiency to adaptive BICM depending on the correlation of the fading. Thus, FTS-ACM is very attractive for systems with mobile users requiring very low latency transmissions.

Chapter 6

Conclusions

This thesis investigated techniques that improve the properties of adaptivity and robustness in communications systems. Adaptivity changes the transmission to match the estimated channel and traffic conditions while robustness allows the same transmission to work consistently well for a wide range of channel conditions. Research results have been presented on several specific topics related to the improvement of adaptivity and robustness.

Chapter 2 considers the design of trellis codes for periodic erasures and their applications. In periodic erasures where each channel is characterized by additive white Gaussian noise with a distinct periodic variation in signal-to-noise ratio, the design of trellis codes becomes a multi-criterion optimization problem. Two objective functions are proposed to search the optimal codes. Based on these objective functions, thirteen codes from searches that maximize the indicated objective functions are provided. Those trellis codes provide reliable communication over all periodic erasure patterns of a given period for which the number of unerased coded bits per period is at least equal to the number of information bits per period.

Furthermore, those trellis codes designed specifically for periodic erasures are also the optimal mother codes of generating families of rate-compatible, unequal error protection trellis codes by using periodic symbol-puncturing. As an application, Chapter 2 applies them to an embedded joint source-channel coding scheme of speech. The simulation results show that those codes are suitable for unequal error protection applications and perform well over a wide range of channel conditions.

The design of a trellis code involves selecting both a convolutional encoder and a labeled constellation. In the context of the optimal edge profile, Chapter 3 considers the labelings of cross constellations. Six distinct classes of quasi-Gray labeled 32-QAM are obtained. Chapter 3 also identifies 8-QAM and quasi-Gray labeled 32-QAM constellations that have the best edge profiles. Those optimal edge profile labelings offer better desired metrics than set-partitioning and Gray labeling in channels with puncturing or fading.

For time-varying channels, one possible method to achieve robust and spectrally efficient communication is to use adaptive coded modulation schemes. In systems supporting highly mobile users that can endure long-latency, adaptive bit-interleaved coded modulation (BICM) achieves a moderate bandwidth efficiency gain over previously proposed adaptive schemes and nonadaptive schemes of similar complexity. Chapter 4 presents a comparison of robustness to channel prediction errors between trellis codes and bit-interleaved coded modulation. This comparison shows that the structure of BICM is more robust to prediction errors than trellis codes modulation, thus motivating the use of adaptive BICM systems for channels supporting users with high mobility.

However, adaptive BICM requires a bit interleaver, which results in long latency. Therefore, in systems with highly mobile users requiring low-latency, adaptive BICM is not a good choice due to the use of a bit interleaver. Chapter 5 proposes two adaptive coded modulation (ACM) schemes, the stochastic ACM and FTS-ACM, for systems with highly mobile users. FTS-ACM uses a forward trellis search algorithm (FTS) to adapt to the current channel fading. Numerical results demonstrate that the proposed FTS-ACM scheme achieves a bandwidth efficiency gain equal or better than that of adaptive BICM. FTS-ACM is particularly attractive for low latency transmission applications.

Appendix A

Proofs of some lemmas in Chapter 3

The following three lemmas apply to any valid quasi-Gray-coded 32-QAM constellation labeling, with minimum distance edge labels $\mathbf{e_1}$, $\mathbf{e_2}$, $\mathbf{e_3}$, $\mathbf{e_4}$, $\mathbf{e_5}$, $\mathbf{e_6}$ and $\mathbf{e_6} = \mathbf{e_3} \oplus \mathbf{e_4} \oplus \mathbf{e_5}$.

Lemma 9: The edge profile structure must conform to the following table, assuming the minimum-distance is one.

Table A.1: The edge profile structure of a valid quasi-Gray-coded 32-QAM constellation labeling (*1 only possible here for the edge label $\mathbf{e_6} = \mathbf{e_3} \oplus \mathbf{e_4} \oplus \mathbf{e_5}$).

The number of minimum-distance basis vectors in edge labels	Possible d^2_{MIN}	The number of edge labels
1	1	$\binom{5}{1} = 5$
2, 4	2, 4, 8, 10	$\binom{5}{2} + \binom{5}{4} = 15$
3, 5	1*, 5, 9, 13	$\binom{5}{3} + \binom{5}{5} = 11$

Proof:

In any valid quasi-Gray coded 32-QAM constellation labeling, by Equation (3.10), some edge paths whose edge labels are the summation of two minimum-distance edge labels can be treated as the summation of four minimum-distance edge labels and vice versa. Similarly, some edge paths whose edge labels are the summation of three minimum-distance edge labels can be treated as the summation of five minimum-distance edge labels and vice versa. Therefore we group the paths containing two or four minimum-distance edge labels together and group the paths containing three or five minimum-distance edge labels together. Since no edge label has a squared distance greater than 13 in 32-QAM edge profiles (Table 3.3), we only need to consider possible paths of edge labels with squared distances less than or equal to 13.

The six possible paths whose edge labels are the summation of two or four minimum-distance edge labels are shown in Figure A.1, where $\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{j} \in \{1, 2, 3, 4, 5, 6\}$.

From Figure A.1, we have $d^2_{\text{MIN}}(E)$ equal to 2 (in (a), (c)) or 4 (in (b), (d)) or 8 (in (e)) or 10 (in (f)). Thus the possible $d^2_{\text{MIN}}(E)$ is 2, 4, 8 or 10, and the number of such edge labels is $\binom{5}{2} + \binom{5}{4} = 15$.

The six possible paths whose edge labels are the summation of three or five minimum-distance edge labels are shown in Figure A.2, where $\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{j}, \mathbf{k} \in \{1, 2, 3, 4, 5, 6\}$.

From Figure A.2, we have $d^2_{\text{MIN}}(E)$ equal to 1 (in (a)) or 5 (in (b),(d)) or 9 (in (c), (e)) or 13 (in (f)). Therefore the possible $d^2_{\text{MIN}}(E)$ is 1, 5, 9 or 13 and the number of such edge labels is $\binom{5}{3} + \binom{5}{5} = 11$. By Equation (3.10), $d^2_{\text{MIN}}(E) = 1$ is only possible here for the edge label $\mathbf{e_6} = \mathbf{e_3} \oplus \mathbf{e_4} \oplus \mathbf{e_5}$. Thus we can obtain the table and this lemma is proven. $\square$

Lemma 10: $\mathbf{e_1}$ and $\mathbf{e_2}$ can not both be used to label vertical edges or both be used to label horizontal edges. As a result, either $\mathbf{e_1}$ labels horizontal edges and

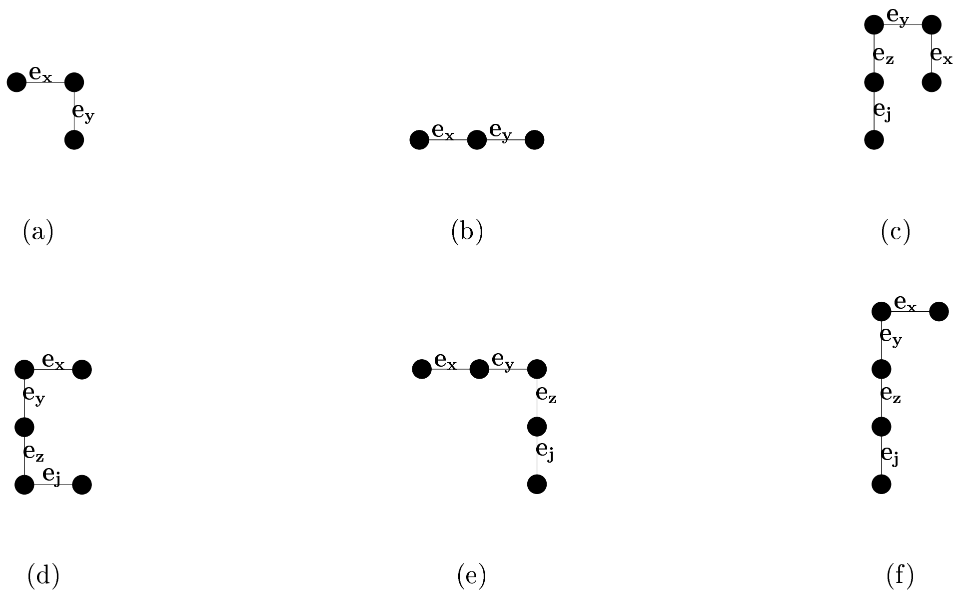


Figure A.1: Six possible paths for two or four minimum-distance vectors in edge labels.

$\mathbf{e_2}$ labels vertical edges or vice versa.

Proof:

Using the symmetry of horizontal edges and vertical edges we only need to show that $\mathbf{e_1}$ and $\mathbf{e_2}$ can not both label vertical edges.

Assume both $\mathbf{e_1}$ and $\mathbf{e_2}$ label vertical edges in 32-QAM, then by Lemma 8, they can not be used to label horizontal edges. Also all other vertical edges sharing the same two horizontal rows as the vertical edge labeled as $\mathbf{e_1}$ ($\mathbf{e_2}$) are labeled as $\mathbf{e_1}$ ($\mathbf{e_2}$). Since each vertical column with six constellation points must be labeled with at least three distinct minimum-distance edge labels by Lemma 5, then there is at least one vertical edge that is labeled with neither $\mathbf{e_1}$ nor $\mathbf{e_2}$. Therefore we have the subconstellations in Figure A.3 to be labeled without using either $\mathbf{e_1}$ or $\mathbf{e_2}$.

In Fig. A.3(a) and Fig. A.3(b), we have to label the subconstellations with

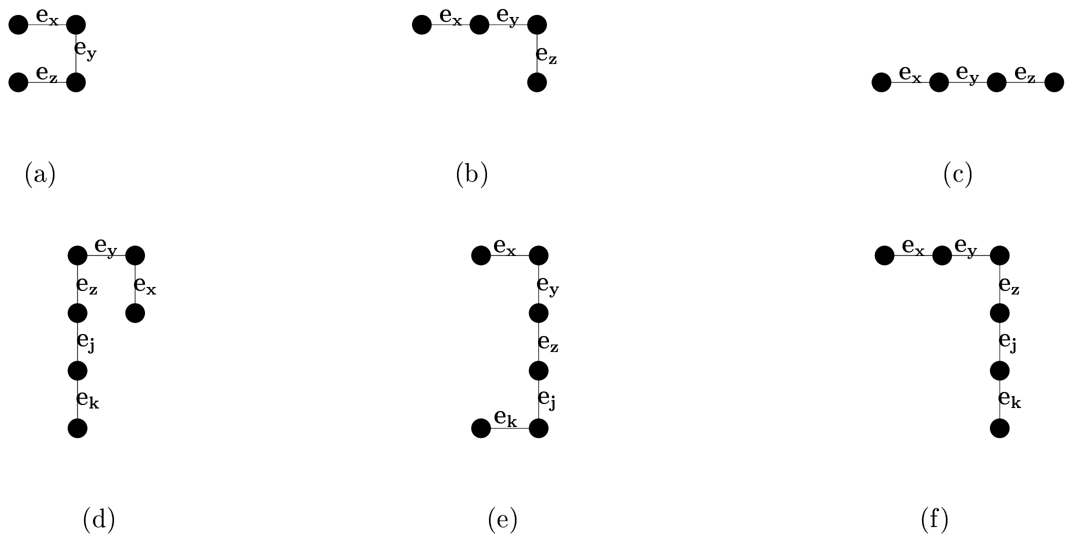


Figure A.2: Six possible paths for three or five minimum-distance vectors in edge labels.

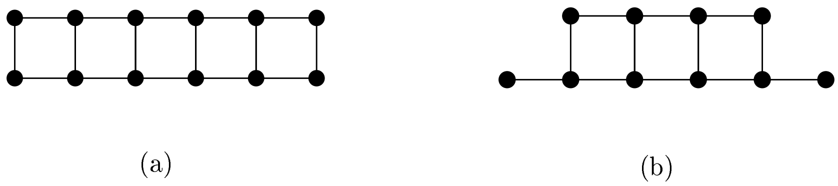


Figure A.3: subconstellations with no edges labeled by either $\mathbf{e_1}$ or $\mathbf{e_2}$.

12 distinct points (in Fig. A.3(a)) or 10 distinct points (in Fig. A.3(b)) by using $\mathbf{e_3}$, $\mathbf{e_4}$, $\mathbf{e_5}$ and $\mathbf{e_6}$. Restricted by Equation (3.10), only three of $\mathbf{e_3}$, $\mathbf{e_4}$, $\mathbf{e_5}$ and $\mathbf{e_6}$ are linearly independent and thus these vectors provide at most 8 distinct point labels by Lemma 4. Therefore it is impossible to label the subconstellations Fig. A.3 (a)(b), meaning that the assumption is not true, and the lemma is proven. $\square$

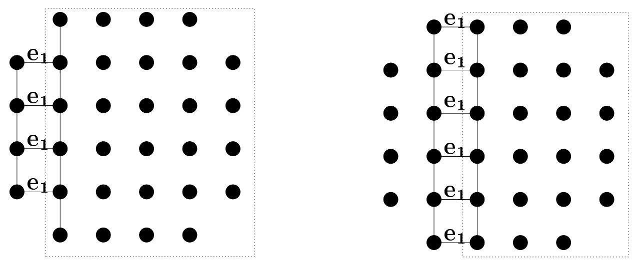
Lemma 11: If $\mathbf{e_1}$ appears exactly once in a six-point horizontal row, then $\mathbf{e_1}$ must appear in the central position; $\mathbf{e_1}$ can not appear twice in a six-point horizontal row. The same labeling rules are followed by $\mathbf{e_2}$ appearing in a six-point vertical column.

Proof:

Using the symmetry of $\mathbf{e_1}$ and $\mathbf{e_2}$, and of horizontal rows and vertical columns, we only need to consider the case of $\mathbf{e_1}$ appearing in horizontal rows.

By Lemma 10, if $\mathbf{e_1}$ appears in horizontal rows, then $\mathbf{e_2}$ must appear in vertical columns.

Assume $\mathbf{e_1}$ labels exactly one edge label and does not appear in the central position. In this case, we must label constellations as in Fig. A.4(a) or Fig. A.4(b), and all edges inside the dotted rectangular boxes must be labeled by $\mathbf{e_2}$, $\mathbf{e_3}$, $\mathbf{e_4}$, $\mathbf{e_5}$ and $\mathbf{e_6}$, which contain only four linearly independent vectors.



(a) (b)

Figure A.4: Constellation labeling structures with $\mathbf{e_1}$ appearing exactly once but not in the central position in horizontal rows.

We must label 28 points in Fig. A.4(a), or 22 points in Fig. A.4(b), by using four linearly independent vectors. However, by Lemma 4, four linearly independent vectors can label at most 16 different points. Thus the assumption is not true, so if $\mathbf{e_1}$ appears once in a six-point horizontal row, then $\mathbf{e_1}$ must appear in the central position.

Assume $\mathbf{e_1}$ appears twice in a six-point horizontal row, Fig. A.5 shows four possible ways to label this horizontal row.

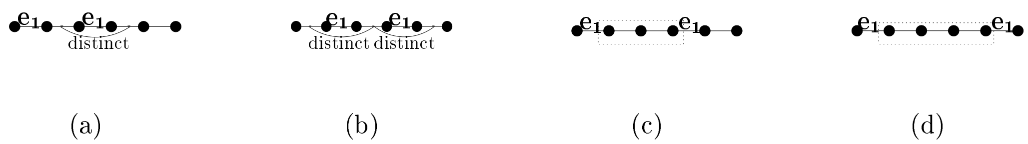


Figure A.5: Four possible labeling ways for $\mathbf{e_1}$ appearing twice in a six-point horizontal row.

We need to label 18 points in Fig. A.5(c) or 24 points in Fig. A.5(d) by using $\mathbf{e_2}$ through $\mathbf{e_6}$, which can label at most 16 points. Thus, these two cases are impossible.

Following the labeling schemes of Fig. A.5(a)-(b), the subconstellation labeling structures are illustrated as Fig. A.6(a)-(b) respectively. In Fig. A.6(a), arbitrarily starting from point **a**, 17 distinct hexagon points must be labeled by using $\mathbf{e_2}$ through $\mathbf{e_6}$, i. e., 17 distinct nonzero edges starting from point **a** must be labeled by linear combinations of $\mathbf{e_2}$ through $\mathbf{e_6}$. By Lemma 5, 17 distinct nonzero edges need at least five linearly independent vectors, while there are only four linearly independent vectors among $\mathbf{e_2}$ through $\mathbf{e_6}$. Therefore, Fig. A.6(a) is invalid. Similarly, we can prove that Fig. A.6(b) is also invalid. Thus Fig. A.5(a) and Fig. A.5(b) are invalid as well. Since all cases have been covered, the lemma is proven. $\square$

Lemma 14: Set-partitioned 32-QAM has at least eight distinct edge labels for minimum-distance edges.

Proof: Continued from Section IIIC.

CASE B: consider the case where all $\mathbf{h_2}$ coset edges have the same 2 MSBs.

(a) (b)

Figure A.6: The subconstellation labeling structures when $\mathbf{e_1}$ appears as Fig. A.5 (a) or (b).

Fig. A.7 shows the procedure of the proof of CASE B.

First, consider Fig. A.8.

Case B1: All $\mathbf{v_1}$ coset edges have the same 2 MSBs, this case is isomorphic to case A2b (where all $\mathbf{h_1}$ coset edges have the same 2 MSBs and all $\mathbf{v_2}$ coset edges have the same 2 MSBs). Then the argument is completed.

Case B2: Consider the cases where at least $2\mathbf{v_1}$s exist.

Since all $\mathbf{h_2}$ coset edges have the same 2 MSBs, $1\mathbf{h_2}$ and at least $2\mathbf{h_1}$s exist by Lemma 5 (the horizontal row has a total of six different points).

(B2a): If there are three distinct $\mathbf{v_2}$ edge labels, then we get $3\mathbf{v_2}$s + $2\mathbf{v_1}$s + $1\mathbf{h_2}$ + $2\mathbf{h_1}$s, the argument is complete.

(B2b): Consider cases where $\mathbf{v_2}$ coset edge labels have fewer than three distinct $\mathbf{v_2}$ coset edge labels.

(B2b1): all $\mathbf{h_2}$ coset edges have the same 2 MSBs, at least $2\mathbf{v_1}$s exist, $\mathbf{qv_2} \neq \mathbf{pv_2}$ and $\mathbf{fv_2} = \mathbf{qv_2}$ or $\mathbf{fv_2} = \mathbf{pv_2}$, i.e., $2\mathbf{v_2}$s exist.

For this case, the set-partition labeling structure is shown in Fig. A.9.

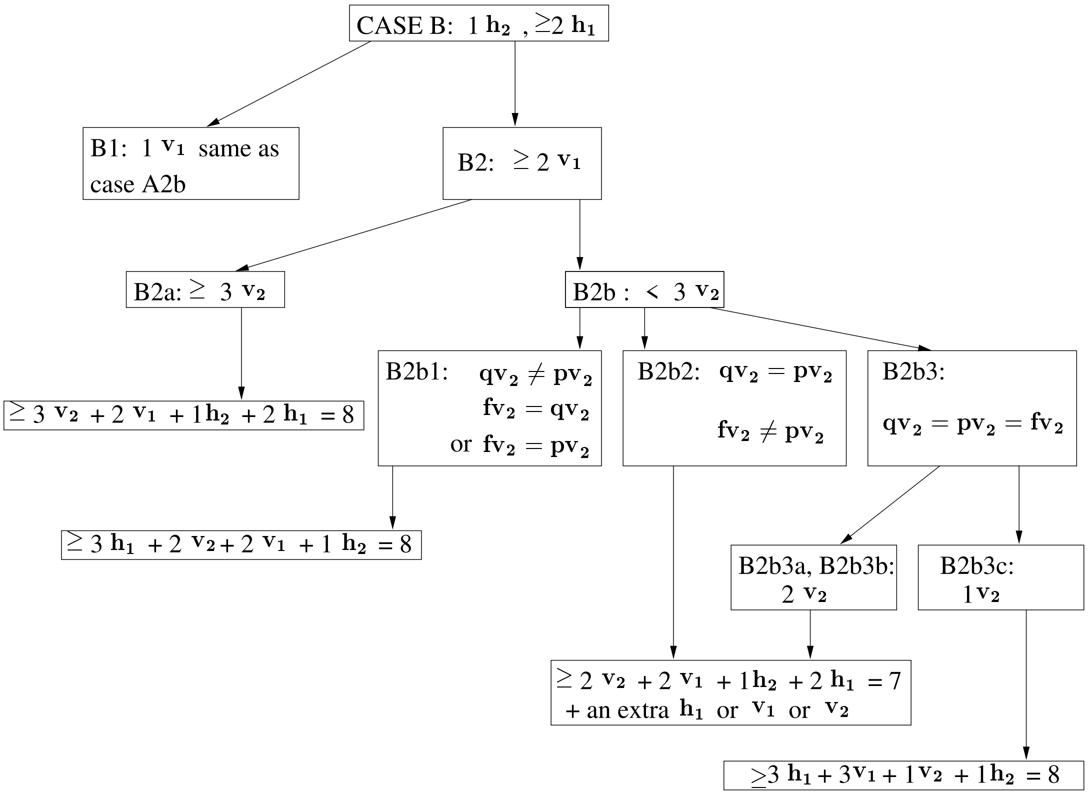


Figure A.7: The procedure of the proof of Lemma 14.

Since $\mathbf{a_1} \oplus \mathbf{a_2} = \mathbf{a_3} \oplus \mathbf{a_4} \neq \mathbf{0}$, we get

$$\mathbf{qv_2} \oplus \mathbf{ev_1} \oplus \mathbf{fv_2} \oplus \mathbf{gv_1} = \mathbf{ih_1} \oplus \mathbf{jh_1} \neq \mathbf{0}\,. \tag{A.1}$$

Since $\mathbf{b_1} \oplus \mathbf{b_2} = \mathbf{b_3} \oplus \mathbf{b_4} \neq \mathbf{0}$, we get

$$\mathbf{ev_1} \oplus \mathbf{fv_2} \oplus \mathbf{gv_1} \oplus \mathbf{pv_2} = \mathbf{lh_1} \oplus \mathbf{mh_1} \neq \mathbf{0}\,. \tag{A.2}$$

Comparing equations (A.1) and (A.2), since $\mathbf{qv_2} \neq \mathbf{pv_2}$, we get $0 \neq \mathbf{ih_1} \oplus \mathbf{jh_1} \neq \mathbf{lh_1} \oplus \mathbf{mh_1} \neq \mathbf{0}$, i.e., $3\mathbf{h_1}$s exist. Therefore we get $3\mathbf{h_1}$s + $2\mathbf{v_2}$s + $2\mathbf{v_1}$s + $1\mathbf{h_2}$, the argument is complete.

(B2b2): all $\mathbf{h_2}$ coset edges have the same 2 MSBs, at least $2\mathbf{v_1}$s exist, $\mathbf{qv_2} = \mathbf{pv_2}$ and $\mathbf{fv_2} \neq \mathbf{pv_2}$.

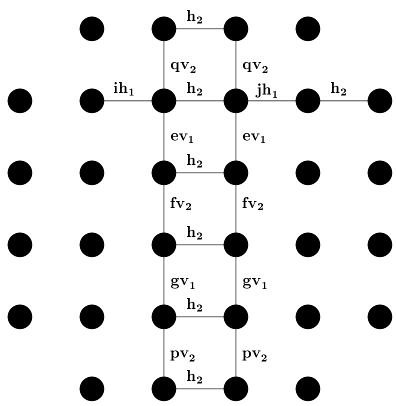


Figure A.8: All $\mathbf{h_2}$ coset edges have the same 2 MSBs.

Since $\mathbf{fv_2} \neq \mathbf{pv_2}$, at least $2\mathbf{v_2}$s exist.

For this case, the set-partition labeling structure is shown in Fig. A.10 and $\mathbf{a_1}$, $\mathbf{a_2}$, $\mathbf{a_3}$, $\mathbf{a_4}$, $\mathbf{e_1}$, $\mathbf{e_2}$ and $\mathbf{e_3}$ are distinct points.

In this case, we know that there are 1 $\mathbf{h_2}$, 2 distinct $\mathbf{h_1}$s ($\mathbf{ih_1}$ and $\mathbf{jh_1}$), 2 distinct $\mathbf{v_1}$s and 2 distinct $\mathbf{v_2}$ ($\mathbf{fv_2}$ and $\mathbf{pv_2}$). So there are a total of seven distinct minimum-distance edge labels. Next we will show that we can not set-partitioned label 32-QAM by using only these seven distinct minimum-distance edge labels, i. e., at least eight distinct minimum-distance edge labels are needed to set-partitioned label 32-QAM. Assume we can set-partitioned label 32-QAM by using only these seven distinct minimum-distance edge labels (Assumption II).

Under Assumption II and Lemma 1, in each six-point horizontal row, for the two unlabeled horizontal edges, one must be labeled as $\mathbf{ih_1}$, the other as $\mathbf{jh_1}$.

In Fig. A.10, since $\mathbf{a_1} \oplus \mathbf{a_2} = \mathbf{a_3} \oplus \mathbf{a_4} \neq \mathbf{0}$, we get

$$\mathbf{pv_2} \oplus \mathbf{ev_1} \oplus \mathbf{fv_2} \oplus \mathbf{gv_1} = \mathbf{ih_1} \oplus \mathbf{jh_1}\,. \tag{A.3}$$

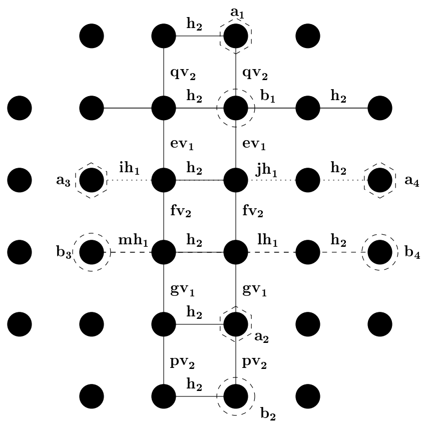


Figure A.9: All $\mathbf{h_2}$ coset edges have the same 2 MSBs, at least $2\mathbf{v_1}$s exist, $\mathbf{qv_2} \neq \mathbf{pv_2}$ and $\mathbf{fv_2}$ is either $\mathbf{qv_2}$ or $\mathbf{pv_2}$.

Since $\mathbf{e_1} \oplus \mathbf{e_3} \neq \mathbf{e_1} \oplus \mathbf{e_2} \neq \mathbf{0}$ in Fig. A.10, we get

$$\mathbf{kv_1} \oplus \mathbf{wv_2} \oplus \mathbf{lv_1} \oplus \mathbf{uv_2} \neq \mathbf{ih_1} \oplus \mathbf{jh_1}\,. \tag{A.4}$$

Fig. A.11 shows the cases considered for the proof that Assumption II is invalid in case B2b2.

(B2b2a). If $\mathbf{ev_1} \neq \mathbf{gv_1}$, then in this case, assuming the two distinct $\mathbf{v_1}$s are $\mathbf{ev_1}$ and $\mathbf{gv_1}$:

(B2b2ai). If $\mathbf{uv_2} = \mathbf{wv_2}$, under Assumption II and Lemma 2, we get

$$\mathbf{uv_2} = \mathbf{pv_2} \textbf{ or } \mathbf{uv_2} = \mathbf{pv_2} \oplus \mathbf{ih_1} \oplus \mathbf{jh_1}\,. \tag{A.5}$$

$$\mathbf{wv_2} = \mathbf{fv_2} \textbf{ or } \mathbf{wv_2} = \mathbf{fv_2} \oplus \mathbf{ih_1} \oplus \mathbf{jh_1}\,. \tag{A.6}$$

Since $\mathbf{uv_2} = \mathbf{wv_2}$, then from Equations (A.5) and (A.6), we obtain

$$\mathbf{pv_2} \oplus \mathbf{fv_2} \oplus \mathbf{ih_1} \oplus \mathbf{jh_1} = \mathbf{0}\,. \tag{A.7}$$

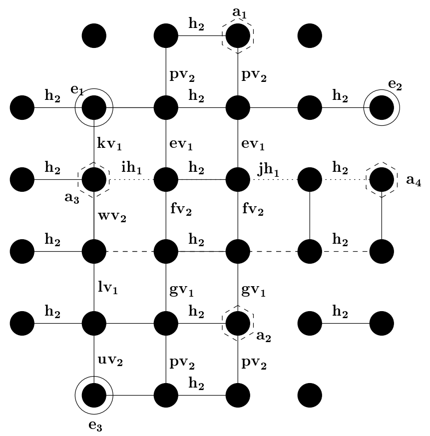


Figure A.10: All $\mathbf{h_2}$ coset edges have the same 2 MSBs, at least $2\mathbf{v_1}$ exist, $\mathbf{qv_2} = \mathbf{pv_2}$, $\mathbf{fv_2} \neq \mathbf{pv_2}$.

Adding Equations (A.3) and (A.7), we get $\mathbf{ev_1} \oplus \mathbf{gv_1} = \mathbf{0}$, which contradicts $\mathbf{ev_1} \neq \mathbf{gv_1}$. Therefore, this case is invalid.

(B2b2aii). If $\mathbf{uv_2} \neq \mathbf{wv_2}$ and $\mathbf{kv_1} \neq \mathbf{lv_1}$, under Assumption II, then

$$\mathbf{kv_1} \oplus \mathbf{wv_2} \oplus \mathbf{lv_1} \oplus \mathbf{uv_2} = \mathbf{ev_1} \oplus \mathbf{pv_2} \oplus \mathbf{gv_1} \oplus \mathbf{fv_2} \neq \mathbf{0}\,. \tag{A.8}$$

Comparing Equations (A.3), (A.4) and (A.8), they are contradictory, thus this case is invalid.

(B2b2aiii). If $\mathbf{uv_2} \neq \mathbf{wv_2}$, $\mathbf{kv_1} = \mathbf{lv_1}$, by Assumption II and Lemma 2, we get

$$\mathbf{kv_1} = \mathbf{ev_1} \textbf{ or } \mathbf{kv_1} = \mathbf{ev_1} \oplus \mathbf{ih_1} \oplus \mathbf{jh_1}\,. \tag{A.9}$$

$$\mathbf{lv_1} = \mathbf{gv_1} \textbf{ or } \mathbf{lv_1} = \mathbf{gv_1} \oplus \mathbf{ih_1} \oplus \mathbf{jh_1}\,. \tag{A.10}$$

Since $\mathbf{kv_1} = \mathbf{lv_1}$, then from Equations (A.9) and (A.10), we obtain

$$\mathbf{ev_1} \oplus \mathbf{gv_1} = \mathbf{ih_1} \oplus \mathbf{jh_1}\,. \tag{A.11}$$

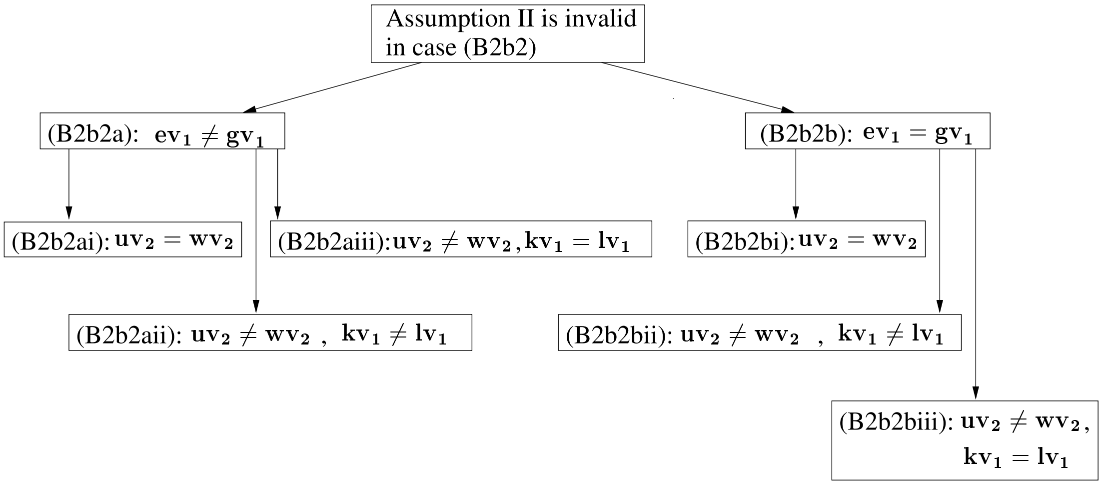


Figure A.11: Cases considered for the proof of invalid Assumption II in case B2b2.

Adding equations (A.3) and (A.11), we get $\mathbf{fv_2} \oplus \mathbf{pv_2} = \mathbf{0}$, which contradicts the fact that $\mathbf{fv_2}$ and $\mathbf{pv_2}$ are two distinct edge labels. Therefore, this case is invalid.

(B2b2b). If $\mathbf{ev_1} = \mathbf{gv_1}$, then Equation (A.3) becomes

$$\mathbf{pv_2} \oplus \mathbf{fv_2} = \mathbf{ih_1} \oplus \mathbf{jh_1}\,. \tag{A.12}$$

(B2b2bi). If $\mathbf{uv_2} = \mathbf{wv_2}$, then by Lemma 1, $\mathbf{kv_1} \neq \mathbf{lv_1}$.

By Assumption II and Lemma 2, we get

$$\mathbf{ev_1} = \mathbf{kv_1} \textbf{ or } \mathbf{ev_1} = \mathbf{kv_1} \oplus \mathbf{ih_1} \oplus \mathbf{jh_1}\,. \tag{A.13}$$

$$\mathbf{gv_1} = \mathbf{lv_1} \textbf{ or } \mathbf{gv_1} = \mathbf{lv_1} \oplus \mathbf{ih_1} \oplus \mathbf{jh_1}\,. \tag{A.14}$$

Since $\mathbf{ev_1} = \mathbf{gv_1}$ and $\mathbf{kv_1} \neq \mathbf{lv_1}$, then from Equations (A.13) and (A.14), we obtain

$$\mathbf{kv_1} \oplus \mathbf{lv_1} \oplus \mathbf{ih_1} \oplus \mathbf{jh_1} = \mathbf{0}\,. \tag{A.15}$$

Also since $\mathbf{uv_2} = \mathbf{wv_2}$, Equation (A.4) becomes

$$\mathbf{kv_1} \oplus \mathbf{lv_1} \neq \mathbf{ih_1} \oplus \mathbf{jh_1} \,. \tag{A.16}$$

Equations (A.15) and (A.16) contradict each other. Therefore, this case is invalid.

(B2b2bii). If $\mathbf{uv_2} \neq \mathbf{wv_2}$ and $\mathbf{kv_1} \neq \mathbf{lv_1}$:

Since $\mathbf{uv_2} \neq \mathbf{wv_2}$, then under Assumption II, one must equal $\mathbf{fv_2}$, the other must equal $\mathbf{pv_2}$.

Since $\mathbf{ev_1} = \mathbf{gv_1}$ and $\mathbf{kv_1} \neq \mathbf{lv_1}$, by Assumption II and Lemma 2, we have Equation (A.15).

Adding equations (A.12) and (A.15), we get

$$\mathbf{kv_1} \oplus \mathbf{fv_2} \oplus \mathbf{lv_1} \oplus \mathbf{pv_2} = \mathbf{0} \,, \mathbf{i.\,e.}, \mathbf{kv_1} \oplus \mathbf{wv_2} \oplus \mathbf{lv_1} \oplus \mathbf{uv_2} = \mathbf{0} \,, \tag{A.17}$$

which contradicts the fact that $\mathbf{e_1}$ and $\mathbf{e_3}$ are two distinct points. Therefore, this case is invalid.

(B2b2biii). If $\mathbf{uv_2} \neq \mathbf{wv_2}$, $\mathbf{kv_1} = \mathbf{lv_1}$:

Since $\mathbf{uv_2} \neq \mathbf{wv_2}$, then under Assumption II, one must equal $\mathbf{fv_2}$, the other must equal $\mathbf{pv_2}$.

Since $\mathbf{kv_1} = \mathbf{lv_1}$, then Equation (A.4) becomes

$$\mathbf{wv_2} \oplus \mathbf{uv_2} \neq \mathbf{ih_1} \oplus \mathbf{jh_1} \,, \mathbf{i.\,e.}, \mathbf{fv_2} \oplus \mathbf{pv_2} \neq \mathbf{ih_1} \oplus \mathbf{jh_1} \,, \tag{A.18}$$

which contradicts Equation (A.12). Therefore, this case is invalid.

Thus, under Assumption II, no set-partitioned labelings for 32-QAM are valid. Therefore Assumption II is not true. Thus set-partitioned 32-QAM needs at least eight distinct minimum-distance edge labels for case (B2b2).

(B2b3): all $\mathbf{h_2}$ coset edges have the same 2 MSBs, at least two distinct $\mathbf{v_1}$ and $\mathbf{qv_2} = \mathbf{pv_2} = \mathbf{fv_2}$.

Since $\mathbf{qv_2} = \mathbf{pv_2} = \mathbf{fv_2}$, by Lemma 2, we get $\mathbf{ev_1} \neq \mathbf{gv_1}$.

For this case, the set-partition labeling structure is shown in Fig. A.12 and $\mathbf{a_1}$, $\mathbf{a_2}$, $\mathbf{a_3}$, $\mathbf{a_4}$, $\mathbf{c_1}$, $\mathbf{c_2}$ and $\mathbf{c_3}$ are distinct points.

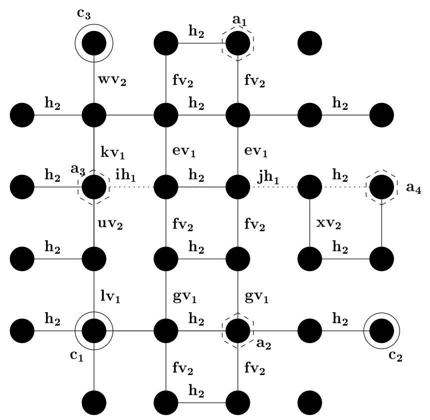


Figure A.12: All $\mathbf{h_2}$ coset edges have the same 2 MSBs, $\mathbf{qv_2} = \mathbf{pv_2} = \mathbf{fv_2}$, $\mathbf{ev_1} \neq \mathbf{gv_1}$.

Since $\mathbf{a_1} \oplus \mathbf{a_2} = \mathbf{a_3} \oplus \mathbf{a_4} \neq \mathbf{0}$(see Fig. A.12), we have

$$\mathbf{ev_1} \oplus \mathbf{gv_1} = \mathbf{ih_1} \oplus \mathbf{jh_1} \neq \mathbf{0}\,. \tag{A.19}$$

Consider the following cases:

(B2b3a). If $\mathbf{kv_1} = \mathbf{lv_1}$, then by Lemma 1, we get $\mathbf{wv_2} \neq \mathbf{uv_2}$, i. e., at least two distinct $\mathbf{v_2}$s exist. Now, we have 1 $\mathbf{h_2}$, 2 distinct $\mathbf{h_1}$s (assuming are $\mathbf{ih_1}$ and $\mathbf{jh_1}$), 2 distinct $\mathbf{v_1}$s (assuming are $\mathbf{ev_1}$ and $\mathbf{gv_1}$) and 2 distinct $\mathbf{v_2}$s. So there are a total of seven distinct minimum-distance edge labels. Using the similar argument to case (B2b2), we can show that no set-partitioned labelings for 32-QAM are valid by using only these seven distinct minimum-distance edge labels. Therefore, for this case, at least eight distinct minimum-distance edge labels are needed for set-partitioned 32-QAM.

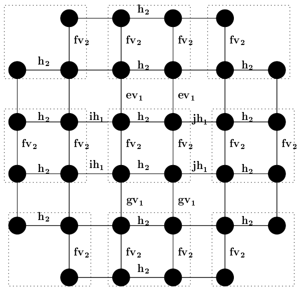
(B2b3b). If $\mathbf{kv_1} \neq \mathbf{lv_1}$ and $\mathbf{wv_2} \neq \mathbf{uv_2}$, then similar to (B2b3a), we have a total of seven distinct minimum-distance edge labels. Using the similar argument to case (B2b2), we can prove that for this case, at least eight distinct minimum-distance edge labels are needed for set-partitioned 32-QAM labelings.

For other cases where we have two distinct $\mathbf{v_2}$s, i. e., have a total of seven distinct minimum-distance edge labels. Using the similar argument to case (B2b2), we can prove that for these cases, at least eight distinct minimum-distance edge labels are needed for set-partitioned 32-QAM labelings.

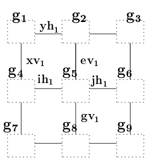
Still, there remains one case where all $\mathbf{v_2}$ coset edges have the same 2 MSBs.

(B2b3c). If all $\mathbf{v_2}$ coset edges have the same 2 MSBs.

The set-partition labeling structure for this case is shown in Fig. A.13(a). All unlabeled edges must be labeled by using only $\mathbf{v_1}$ and $\mathbf{h_1}$ coset edge labels. Using the similar discussion to the proof of Case 4 in Theorem 2 (see Fig. 3.14), we can obtain the grouped labeling structure shown in Fig. A.13(b), where grouped points $\mathbf{g_1}$ through $\mathbf{g_9}$ must be labeled by using $\mathbf{v_1}$ and $\mathbf{h_1}$ coset edge labels.



(a)



(b)

Figure A.13: All $\mathbf{h_2}$ coset edges have the same 2 MSBs, all $\mathbf{v_2}$ coset edges have the same 2 MSBs, $\mathbf{ev_1} \neq \mathbf{gv_1}$: (a) Set-partition labeling structure. (b) The grouped constellation labeling structure.

From Equation (A.19), we have $\mathbf{ev_1} \oplus \mathbf{gv_1} = \mathbf{ih_1} \oplus \mathbf{jh_1} \neq \mathbf{0}$, therefore only three of them are linearly independent, which means these vectors can label at most 7 nonzero edges starting from a point by Lemma 4. Since Fig. A.13(b) has a total of 8 nonzero edge labels starting from a point, we need at least an extra vector $\mathbf{v_1}$ or $\mathbf{h_1}$ to label Fig. A.13(b) by Lemma 4. If $\mathbf{xv_1}$ is the extra distinct edge label having $\mathbf{v_1}$ as the 3 LSBs, similar to earlier discussion, it is easy to show that another extra $\mathbf{yh_1}$ also must appear in Fig. A.13(b) edge labels. Now we have $3\mathbf{h_1}$s + $3\mathbf{v_1}$s + $1\mathbf{v_2}$ + $1\mathbf{h_2}$, therefore there are at least eight distinct edge labels for minimum-distance edges in set-partitioned 32-QAM.

Since all cases are covered, thus set-partitioned 32-QAM has at least eight distinct minimum-distance edge labels for CASE B. $\square$

Bibliography

[1] T. N. Zogakis, J. T. Aslanis Jr., and J. M. Cioffi, "A Coded and Shaped Discrete Multitone System," *IEEE Trans. Commun.*, vol. 43, PP. 2941-49, December 1995.

[2] Y. S. Leung, S. G. Wilson and J. W. Ketchum, "Multifrequency Trellis Coding with Low Delay for Fading Channels," *IEEE Trans. on Comm.*, vol. 41, no. 10, pp. 1450-59, October 1993.

[3] R. D. Wesel and J. M. Cioffi, "Fundamentals of Coding for Broadcast OFDM," *in Proceedings of 29th Asilomar Conf. on Signals, Systems, and Computers*, Pacific Grove, CA, November 1995, pp. 2-6

[4] R. D. Wesel, "Trellis Code Design for Correlated Fading and Achievable Rates for Tomlinson Harashima Precoding," Ph.D. dissertation, Stanford Univ. Stanford, CA, Aug. 1996.

[5] R. D. Wesel and J. M. Cioffi, "Joint Trellis Code and Interleaver Design," in *proceedings of the Communication Theory Mini Conference at Globecom 97*, Phoenix, AZ, November 3-8, 1997, pp. 939-43.

[6] R. D. Wesel, X. Liu and J. M. Cioffi, "Trellis Codes for Compound Periodic Gaussian Channels," *in Proceedings of Int. Symp. on Information Theory*, Cambridge, MA, August, 1998, pp.462.

[7] R. D. Wesel and J. M. Cioffi, "Trellis Codes for Periodic Interleavers," *IEEE Comm. Let.*, vol. 3, no. 4, pp.103-5, April 1999.

[8] E. Malkamaki and H. Leib, " Rate 1/n Convolutional Codes with Interleaving Depth of n over a Block Fading Rician Channel," *in Proceedings of IEEE 47th Vehicular Technology Conf.*, Phoenix, AZ, 1997, pp. 2002-06.

[9] E. Malkamaki and H. Leib, "Coded Diversity on Block-Fading Channels," *IEEE Trans. Inform. Theory*, vol. 45, pp. 771-81, March 1999.

[10] E. Malkamaki and H. Leib, "Evaluating the Performance of Convolutional Codes over Block Fading Channels," *IEEE Trans. Inform. Theory*, vol. 45, PP. 1643-46, July 1999.

[11] R. Knopp and P. A. Humblet, "Maximizing Diversity on Block-Fading Channels," *in Proceedings of ICC'97*, June 1997, pp. 647-51.

[12] W. L. Root and P. P. Varaiya, "Capacity of Classes of Gaussian Channels," SIAM J. Appl. Math, vol. 16, pp. 1350-93, November 1968.

[13] K. J. Larsen, "Comments on "An Efficient Algorithm for Computing Free Distance," *IEEE Trans. Inform. Theory*, vol IT-19, pp. 577-79, July 1973.

[14] D. Divsalar and M. K. Simon, "Multiple Trellis Coded Modulation (MTCM)," *Proc. IEEE Global Telecommunications Conf.*, pp. 1088-1094, 1986.

[15] G. D. Forney Jr., "Convolutional Codes I: Algebraic Structure," *IEEE Trans. Inform. Theory*, vol IT-16, pp. 720-38, November 1970.

[16] G. Ungerboeck, "Channel Coding with Multilevel/Phase Signals," *IEEE Trans. on Inform. Theory*, vol. IT-28, no. 1, pp. 55-67, January 1982.

[17] A. Lapidoth, "The Performance of Convolutional Codes on the Block Erasure Channel Using Various Finite Interleaving Techniques," *IEEE Trans. Inform. Theory,* vol. 40, pp. 1459-73, September 1994.

[18] R. D. Wesel and J. M. Cioffi, "Trellis Codes Design for Periodic Interleavers," *IEEE Commun. Lett.,* vol. 3, pp. 103-105, April 1999.

[19] R. D. Wesel, X. Liu and W. Shi, "Trellis Codes for Periodic Erasures," *IEEE Transactions on Communications,* vol. 48, no. 6, pp. 938-47, June 2000.

[20] R. D. Wesel, C. Komninakis, and X. Liu, " Towards Optimality in Constellation Labeling," in *proceedings of the Communication Theory Mini Conference at Globecom 97*, Phoenix, AZ, November 3-8, 1997.

[21] X. Liu and R. D. Wesel, "Profile Optimal 8-QAM and 32-QAM Constellations," in *Thirty-Sixth Annual Allerton Conference on Communications, Control, and Computing*, September 23-25, 1998.

[22] J. B. Anderson and K. Balanchandran, "Decision Depths of Convolutional codes," *IEEE Trans. Inform. Theory*, vol. 35, pp. 455-59, march 1989.

[23] R. D. Wesel, X. Liu, and W. Shi, "Periodic Symbol Puncturing of Trellis Codes," *in Proceedings of 31st Asilomar Conf. Signals, Systems, and computers*, Pacific Grove, CA, November 1997, pp. 172-76.

[24] G. Caire, E. Leonardi, and E. Viterbo, "Concatenated Coding for Packet Transmission over the Collision Channel," *in Proceedingd of ISIT'98*, Cambridge, MA, 1998, pp. 288.

[25] A. Bernard, X. Liu, R. Wesel and A. Alwan, "Channel Adaptive Joint Source-Channel Coding of Speech," *in Proeedings. of the 32nd Asilomar Conference on Signals, Systems, and Computers*, November, 1998.

[26] A. Bernard, X. Liu, R. D. Wesel and A. Alwan, "Embedded Joint Source-Channel Coding of Speech Using Symbol Puncturing of Trellis Codes," *in Proceedings of IEEE ICASSP*, March, 1999.

[27] A. Bernard, X. Liu, R. D. Wesel and A. Alwan, "Adaptive Multi-Rate Transmissio n of Speech Using Rate-Compatible Trellis Codes and Embedded Souce Coding," *submitted to IEEE Transactions on Communication*, July 2000.

[28] J. Hagenauer, "Rate-Compatible Punctured Convolutional Codes and their Applications," *IEEE Trans. on Comm.*, vol. 36, no. 4, pp. 389-400, April 1988.

[29] D. Divsalar and M. K. Simon, "The Design of Trellis Coded MPSK for Fading Channels: Performance criteria," *IEEE Trans. on Comm.*, vol. 36, no. 9, pp. 1004-12, September 1988.

[30] S. G. Wilson and Y. S. Leung, "Trellis Coded Phase Modulation on Rayleigh Channels," *in Proceedings of ICC-87*, June 1987, pp. 739-43.

[31] R. D. Wesel, X. Liu, and W. Shi, "Trellis Codes for Periodic Erasures," *IEEE Trans. on Comm.*, vol. 48, no. 6, pp. 938-47, June 2000.

[32] F. Gray, "Pulse Code Communications," *U. S. Patent 2632058*, March 1953.

[33] G. D. Forney, Jr.,"Coset Codes - Part I: Introduction and Geometrical Classification," *IEEE Trans. on Inform. Theory*, vol. 34, no. 5, pp. 1123-41, September 1988.

[34] R. D. Wesel, X. Liu, J. M. Cioffi and C. Komninakis, "Constellation Labeling for Linear Encoders," *IEEE Trans. on Inform. Theory*, submitted August 2000.

[35] G. Ungerboeck, "Trellis-Coded Modulation with Redundant Signal Sets: Parts I and II," *IEEE Comm. Mag.*, vol. 25, no. 2, pp. 5-20, February 1987.

[36] J. Du, Y. Kamino, H. Sasaoka and B. Vucetic, "New 32-QAM Trellis Codes for Fading Channels," *Eur. Trans. on Telecom.*, vol. 4, no. 3, pp. 335-41, May-June 1993.

[37] G. Caire, G. Taricco and E. Biglieri, "Bit Interleaved Coded Modulation," *IEEE Trans. on inform. Theory*, vol. 44, no. 3, pp. 927-946, May 1998.

[38] D. Goeckel, "Adaptive Coding for Time-Varying Channels Using Outdated Fading Estimates," *Proc. Allerton Conf. Communications, Control, and Computing*, 1997.

[39] D. Goeckel, "Adaptive Coding for Time-Varying Channels Using Outdated Fading Estimates," *IEEE Trans. Commun.*, vol. 47, no. 6, pp. 844-55, June 1999.

[40] J. Cavers, "Variable-Rate Transmission for Rayleigh Fading Channels," *IEEE Trans. Commun.*, Vol. 20, pp. 15-22, February 1972.

[41] P. Ormeci, D. Goeckel, and R. D. Wesel, "Adaptive Bit-Interleaved Coded Modulation for Time-Varying Channels Using Outdated Fading Estimates,"

33rd Ann. Conference on information sciences and systems, Baltimore, MD.,March 14-17, 1999.

[42] A. Goldsmith, "Capacity and Dynamic Resource Allocation in Broadcast Fading Channels,"*Proc. Allerton Conf. Communications, Control, and Computing*, pp. 915-924, 1995.

[43] G. Caire, G. Taricco, and E. Biglieri, "Optimum Power Control Over Fading Channels," *IEEE Trans. Inform. Theory*, Vol. 45, No. 5, pp. 1468-1489, July 1999.

[44] C. Schlegel and D. Costello, Jr., "Bandwidth Efficient Coding for Fading Channels: Code Construction and Performance Analysis," *IEEE Journal Select. Areas Commun.*, vol. 7, pp. 1356-68, December 1989.

[45] E. Biglieri, "High-Level Modulation and Coding for Nonlinear Satellite channels," *IEEE Trans. on Comm.*, Vol. 32, pp. 616-626, May 1984.

[46] M. Rouanne and D. J. Costello, Jr., "An Algorithm for Computing the Distance Spectrum of Trellis Codes," *J. of Sel. Areas in Comm.*, 7(6):929-940, August 1989.

[47] E. Zehavi and J. K. Wolf, "On the Performance Evaluation of Trellis codes," *IEEE Trans. on Info. Th.*, 33(2):196-202, March 1987.

[48] R. D. Wesel, "Reduced Complexity Trellis Code Transfer Function Computation," *Communication Theory Miniconference in conjunction with ICC99*, June 6-10, 1999.

[49] R. D. Wesel and X. Liu, "Analytic Techniques for Periodic Trellis Codes," in *Thirty-Sixth Annual Allerton Conference on Communications, Control, and Computing*, September 23-25, 1998.

[50] D. Kucukyavuz and M. P. Fitz, "New Views of Transfer Function Based Performance Analysis of Coded Modulations," In *Thirty-Three Asilomar Conference on Signals, Systems, and Computers*, Oct. 1999.

[51] E. Zehavi, "8-PSK Trellis Codes for a Rayleigh Channel," *IEEE Trans. on Comm.*, 40(5):873-884, May 1992.

[52] B.Vucetic, "An Adaptive Coding Scheme for Time Varying Channels," *IEEE Trans. Commun.*, Vol. 39, pp. 653-663, May 1991.

[53] S. Alamouti and S. Kallel, "Adaptive Trellis Coded Multiple-Phase-Shift Keying for Rayleigh Fading Channels," *IEEE Trans. Commun.*, Vol. 42, pp. 2305-2314, June 1994.

[54] A. Goldsmith and S. Chua, "Variable-Rate Variable-Power MQAM for Fading Channels," *IEEE Trans. Commun.* , Vol. 45, pp. 1218-1230, October 1997.

[55] M. Alouini and A. Goldsmith, "Adaptive M-QAM Modulation over Nakagami Fading Channels," *Proc. IEEE Global Communications Conf. - Communication Theory Miniconf.*, pp. 218-223, November 1997.

[56] P. Bello, "Characterization of Randomly Time-Variant Linear Channels," *IEEE Trans. Commun. Systems*, Vol. 11, pp. 360-393, December 1963.

[57] D. Goeckel, "Adaptive Coding for Fading Channels using Outdated Fading Estimates," *Proc. 1998 IEEE 48^{th} Vehicular Tech. Conf.,* pp. 1925-1929, May 1998.

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

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

[60] J. W. Craig, "A New, Simple, and Exact Result for Calculating the Probability of Error for Two-Dimensional Signal Constellations," in *Proc. IEEE Milit. Commun. Conf. MILCOM'91*, McLean, VA, Oct. 1991, pp. 571-575.

[61] I. Gradshteyn and I. Rhyzhik, *Table of Integrals, Series, and Products*, Academic Press, New York, 1980.

[62] R. D. Wesel and X. Liu, "Edge Profile Optimal Constellation Labeling," *in Proceedings of ICC'00*, New Orleans, LA, June 2000.

[63] R. L. Peterson, R. E. Ziemer and D. E. Borth, *Introduction to Spread-Spectrum Communications*, Prentice-Hall, Inc., 1995.