Coding for the Rayleigh Fading Channel

A thesis
submitted in fulfilment
of the requirements for the Degree of

Doctor of Philosophy

in Electrical and Electronic Engineering
from the University of Canterbury
Christchurch, New Zealand

Robert van Nobelen
B.E. (Hons 1), M.E.

February 1996

Abstract.

The problem of effective coding for the Rayleigh fading channel is addressed. A signal transmitted over the Rayleigh fading channel experiences quasi-periodic deep fades in the signal amplitude, and random phase shifts. The fades in signal amplitude are the cause of error events and limit the bit error rate of an uncoded system to about 10^{-4} at a signal-to-noise ratio of 30 dB. For applications where a lower error rate is desired, more sophisticated methods of transmission and detection are required. We present an extensive generalised analysis of the probability of error of maximum likelihood sequence estimation techniques for the Rayleigh fading channel.

This analysis provides the criteria for good code design for the Rayleigh fading channel and we extend the recently developed area of geometrically uniform (GU) codes for the AWGN channel, to the Rayleigh fading channel, and present the results of searches for good GU trellis codes.

The concept of geometric uniformity is extended to set of points to form geometrically uniform partitions of signal sets. The GU partitions readily allow powerful multi-level codes to be defined with good distance properties. Multi-level codes have the advantage of outperforming trellis codes in terms of decoding complexity at the cost of greater decoding delay. Good multi-level codes over GU partitions are presented and compared with similarly performing trellis codes.

Finally a system is presented which combines the techniques of multiple symbol differential detection (MSDD) with multi-level coding to obtain a good probability of error performance without assuming coherent detection, or ideal channel state information.

Contents

Acknowledgements.

I wish to thank my supervisor, Professor Des P. Taylor, for his excellent supervision of this work, the continued support, both financially and morally, the careful proofreading of this thesis, and the many pints of beer at the staff club.

Special thanks are due to all the people of the comms. lab, Matt, Brian, Anthony, John, Richard, Leon, Nick, Steve, Katharine, Peter, Thawatt, and Perapol, for their company.

Preface.

Communication is fundamental to human existence and nothing is more indicative of this need than the phenomenal growth of The Internet and the World Wide Web. For any form of communication we need a medium or channel for the propagation of information. The most common modern medium is the telephone channel which traditionally carried audio signals in analogue form. With the advent of computers and digital techniques, most information traffic is now digital and this has opened the need for research into a new area of difficult and interesting engineering problems. The number of applications of digital traffic is limited only by the imagination, but we can readily divide it into two classes. The first class can tolerate a relatively high error rate (from 10^{-3} to 10^{-4}) and examples of this class are digitised voice and video images. The occasional incorrectness of the data does in general not affect the quality of communications significantly. On the other hand, the 2nd class of traffic ideally requires perfect integrity of data (in practice an error rate of about 10^{-9}), and an example is computer information. In this thesis we study the communications channel, known as the Rayleigh fading channel. With a simple uncoded modulation format, digital data transmitted across the Rayleigh fading channel typically, as a result of the quasi-periodic deep fades experienced by the signal, experiences an error rate of about 10^{-4} and therefore is only suited for the class of applications discussed that can tolerate a high error rate. With the technique of coding it is possible, at the expense of complexity, to vastly improve on the error rate without sacrificing the data rate, increasing the bandwidth or increasing the power.

The work reported in this thesis is specifically concerned with trellis coding and multi-level coding techniques based on the concept of geometric uniformity of the code sequence set. The property of geometric uniformity implies the code performance may be evaluated by considering only one codeword and therefore reducing search evaluation complexity significantly.

The first chapter describes the nature of the fading channel and a bound on its capacity. The capacity bound shows that it should be possible to achieve a performance much better than uncoded modulation. Chapter two describes and analyses the technique of maximum likelihood sequence estimation (MLSE) to improve upon

uncoded modulation. These results supply the necessary conditions for good codes for the Rayleigh fading channel. Chapter three introduces the concepts of signal space coding and decoding. In chapter four we extend the definition of geometric uniformity and apply this definition to discover GU good trellis codes for the Rayleigh fading channel. Chapter five extends the concept of geometric uniformity to a set of points to form geometrically uniform partitions, which are applied to construct multi-level codes in chapter six. Multi-level codes trade off performance for decoding complexity, such that for a given level of decoding complexity they still outperform maximum likelihood trellis codes. Chapter seven combines multiple symbol differential detection (MSDD) with multi-level codes into a system that can tolerate a high speed of fading compared to conventional differential detection, while still maintaining a low error rate. The conclusions are presented in chapter eight. The contributions considered original are those of chapters two, four, five, six and seven.

The work reported in this thesis was performed during the period of February 1993 to February 1996 and the following papers have been published, presented, or are in preparation.

R. van Nobelen. "Trellis codes for mobile radio." Telecom of New Zealand research contract final report 1, Jul. 1993

R. van Nobelen. "Trellis codes for mobile radio." Telecom of New Zealand research contract final report 2, Jul. 1994

R. van Nobelen and D.P. Taylor. "Analysis of the pairwise error probability of non-interleaved codes on the Rayleigh fading channel." In *Globecom '94*, Mini-conf. rec., San Francisco, pp. 181-185, Nov. 1994.

R. van Nobelen and D.P. Taylor. "Geometrically uniform trellis codes for the Rayleigh fading channel." In *Globecom '95*, conf. rec., Singapore, vol. 3, pp. 1820-1824, Nov. 1995

R. van Nobelen and D.P. Taylor. "Analysis of the pairwise probability of non-interleaved codes on the Rayleigh fading channel." Accepted for publication in the *IEEE Trans. Commun.*

R. van Nobelen and D.P. Taylor. "Multi-level codes based on geometrically uniform partitions." To be submitted.

R. van Nobelen and D.P. Taylor. "Multi-level codes for differential detection on the

Chapter 1

The Rayleigh fading channel

1.1 The fading channel model.

An RF signal of carrier frequency f_c that is transmitted from a base station to a mobile unit in a typical urban or rural environment exhibits extreme variation in both phase and amplitude. These effects are due to the motion of the vehicle and the scattering of the signal by interfering objects such as trees, poles and buildings. Three components in the received faded signal can be identified at the mobile antenna: the direct line-of-sight component, the specular component and the diffuse component. The combined direct and specular components are usually referred to as the *coherent received component* and the diffuse component referred to as the *noncoherent received component.* In this study of the terrestrial mobile channel, it is assumed that the received coherent component is totally blocked, i.e. there are no line of sight or specular components. If a received coherent component does exist then the channel is referred to as *Rician.* The channel is modelled as shown in figure 1.1. The mobile is travelling at a velocity v, and at any point in time receives N

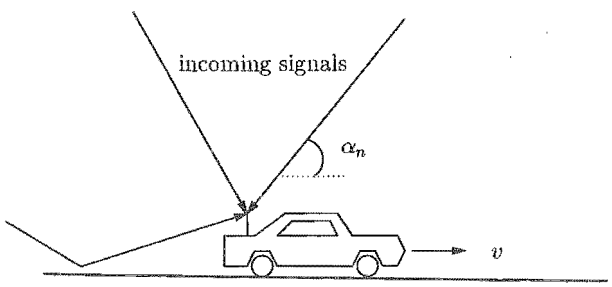


Figure 1.1: Travelling mobile.

incoming multipath signals each with a different incidence angle α_n. This received signal can be represented by a linear superposition of waves of random phase each of whose frequencies is affected by a Doppler shift f_n of the carrier frequency. This shift is a function of the mobile velocity, carrier frequency and angle of incidence to the antenna and can be expressed as

$$f_n = \frac{v}{\lambda} \cos \alpha_n \tag{1.1}$$

where α_n is the incidence angle of the mobile to the n^{th} incoming wave, v is the velocity of the mobile and λ is the wavelength of the transmitted carrier frequency. The maximum Doppler shift occurs when the incoming wave heads straight for the mobile ($\alpha_n = 0$) and is defined as

$$f_D = \frac{v}{\lambda}. \tag{1.2}$$

The received signal $e(t)$ is written as the sum of N incident waves.

$$e(t) = E \sum_{n=1}^{N} e_n cos(2\pi f_c t + \theta_n), \tag{1.3}$$

where

$$\theta_n = 2\pi f_n t + \phi_n \tag{1.4}$$

and the ϕ_n are random phase angles uniformly distributed from 0 to 2π. The e_n are zero mean and normalised such that the ensemble average $E[\sum_{n=1}^{N} e_n^2] = 1$. Therefore $E^2/2$ represents the mobile's average received signal power.

To analyse the statistics of the received signal, equation (1.3) is expressed in bandpass form

$$e(t) = T_c(t) \cos 2\pi f_c t - T_s(t) \sin 2\pi f_c t, \tag{1.5}$$

where

$$T_c(t) = E \sum_{n=1}^{N} e_n \cos(2\pi f_n t + \phi_n), \tag{1.6}$$

$$T_s(t) = E \sum_{n=1}^{N} e_n \sin(2\pi f_n t + \phi_n), \tag{1.7}$$

For large N, the central limit theorem states that $T_c(t)$ and $T_s(t)$ become Gaussian random processes. Denoting $T_c(t)$ and $T_s(t)$ by T_c and T_s respectively for fixed t, the random variables T_c and T_s have zero mean and equal variance of

$$E[T_c^2] = E[T_s^2] = \frac{E^2}{2} \tag{1.8}$$

where the expectation is taken over the random variables α_n, ϕ_n and e_n. T_c and T_s are also uncorrelated i.e.

$$E[T_c T_s] = 0. \tag{1.9}$$

The probability density functions of the Gaussian variables T_c and T_s are of the form

$$p(x) = \frac{1}{\sqrt{2\pi b}} e^{-x^2/2b} \quad , \quad -\infty < x < \infty \tag{1.10}$$

where $b = E^2/2$ is the mean power, and $x = T_c$ or T_s. It can be shown that the probability density function for fixed t of the envelope,

$$u = \sqrt{T_c^2 + T_s^2} \tag{1.11}$$

of $e(t)$ is given by

$$p(u) = \begin{cases} \frac{u}{b} e^{-u^2/2b}, & u \geq 0 \\ 0, & u < 0 \end{cases} \tag{1.12}$$

which is the Rayleigh distribution [51]. The probability density of the phase

$$\phi = \tan^{-1} \frac{T_s}{T_c}, \tag{1.13}$$

of $e(t)$ is uniform and is given by

$$p(\phi) = \tfrac{1}{2\pi}, \quad 0 \leq \phi < 2\pi. \tag{1.14}$$

The autocorrelation function of $u(t)$, assuming isotropic scattering, can be shown to be [43]

$$E[u(t)u(t+\tau)] = \frac{E^2}{2} J_0(2\pi f_D \tau) \tag{1.15}$$

where $J_0(\cdot)$ is the zero-order Bessel function.

By normalising the maximum Doppler frequency f_D to the symbol rate $1/T$ the figure $f_D T$ (called the normalised Doppler frequency) indicates the rate of fading

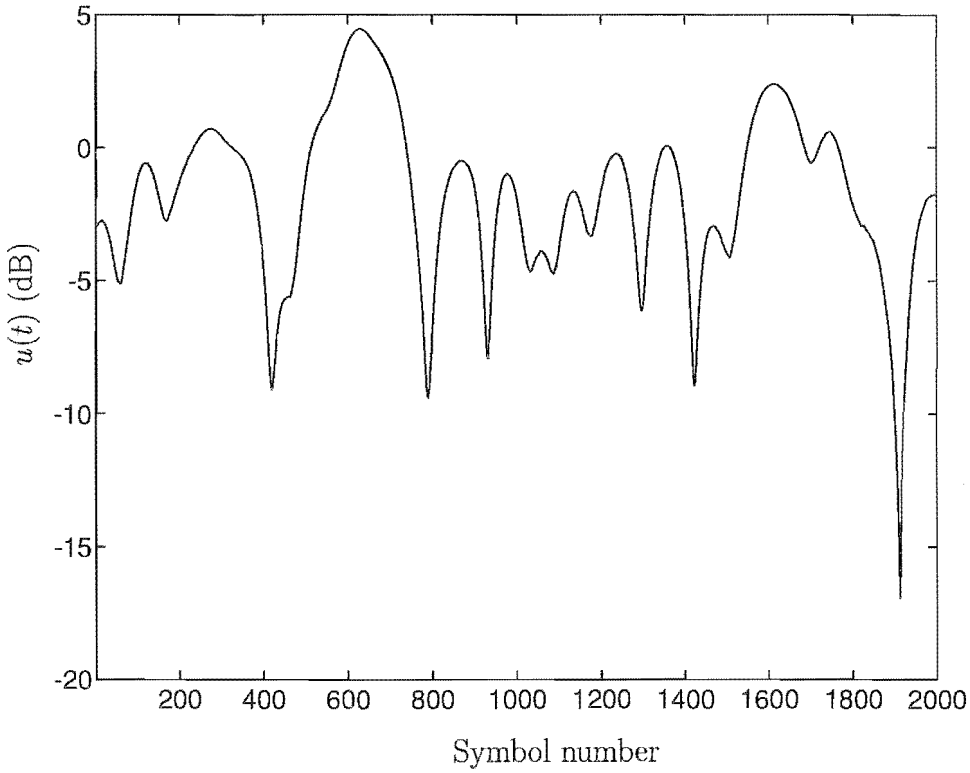


Figure 1.2: Simulated faded carrier amplitude $u(t)$ for $f_D T = 0.01$.

and the number of symbols which are significantly correlated. Values of $f_D T$ ranging from 0.001 to 0.1 are understood to mean very slow to very fast fading respectively. Figures 1.2 and 1.3 show simulated fading envelopes $u(t)$ for values of $f_D T$ of 0.01 and 0.1 respectively. The typical behaviour of the amplitude of the Rayleigh fading process is an oscillatory motion with sudden rapid deep fades occurring at almost regular intervals. The depth of the fades can easily be as much as 20 dB and these are the cause of most error events in a communication system. The analysis of the travelling mobile shows that the phase and amplitude of the received carrier signal varies with time. The electronics of the front end receiver contributes a thermal noise component to the received signal. We may write the overall system in complex baseband form such that if $c(t)$ is the transmitted complex baseband signal then the received baseband signal $r(t)$ after passing through the channel is modelled by the equation

$$r(t) = u(t)c(t) + n(t) \tag{1.16}$$

where $u(t)$ is now the complex baseband equivalent of the fading process, i.e. a

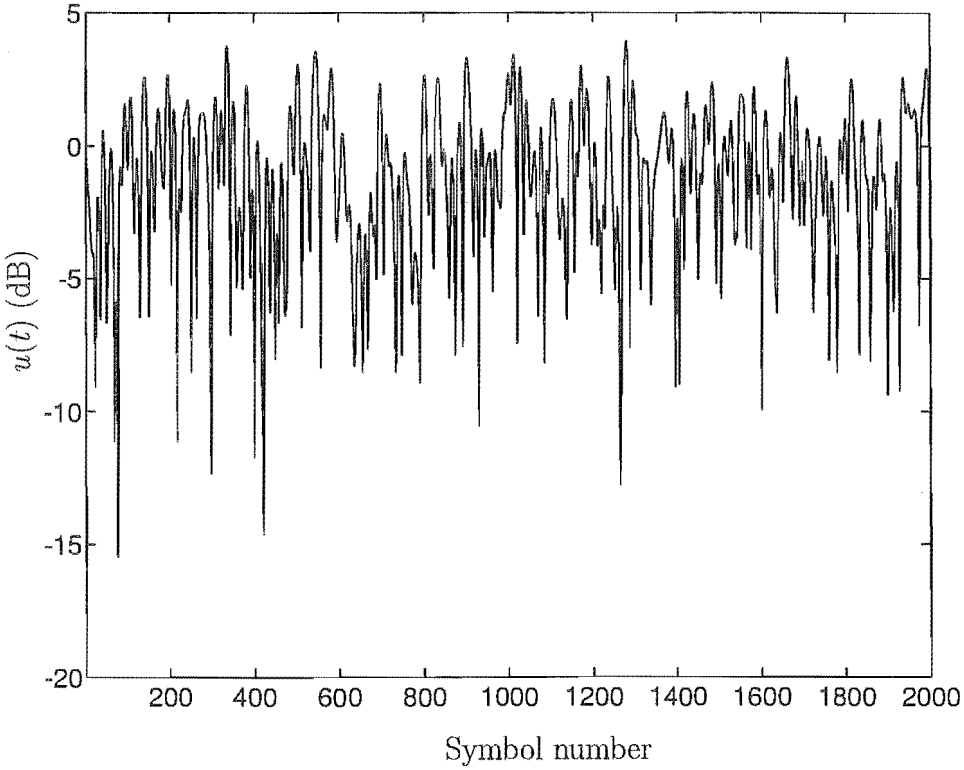


Figure 1.3: Simulated faded carrier amplitude $u(t)$ for $f_D T = 0.1$.

complex Gaussian random variable, and $n(t)$ represent complex additive Gaussian noise. Equation (1.16) is the channel model used for the remainder of this work. The effect of $u(t)$ is that it randomly rotates the phase and scales the amplitude of the received signal $c(t)$. The instantaneous value of $u(t)$ is referred to as the *channel state*. Clearly the receiver requires information about the channel state for successful detection. Channel state information (CSI) may be obtained through pilot tones, pilot symbols and differential detection. Often for the purpose of analysis ideal CSI is assumed, and represents the best we can hope for in practice. More about channel state estimation is presented in chapters 2 and 7.

1.2 Channel capacity.

In 1948 Shannon derived the now famous bound on the capacity of the additive white Gaussian noise (AWGN) channel [66]. This bound relates the capacity C of a bandlimited Gaussian channel to the bandwidth and the SNR of the system and is

given by

$$C = B\log_2(1+\Gamma) \text{ bits/second} \tag{1.17}$$

where B is the channel bandwidth and $\Gamma = E_s/N$ is the signal-to-noise ratio of the system. This equation tells us the absolute best that the system can provide given the channel parameters Γ and B. It shows that the channel has infinite capacity as Γ approaches infinity, and that the channel capacity does not become infinite when the bandwidth B becomes infinite because the noise power increases with increasing bandwidth. If $N = N_0B$, where N_0 is the noise power per Hertz, then in the limit as B tends towards infinity equation (1.17) becomes :

$$\lim_{B\to\infty} C = \frac{1}{\log_e 2}\frac{E_s}{N_0} \approx 1.44\frac{E_s}{N_0}. \tag{1.18}$$

This defines the Shannon limit of -1.6 dB below which error free communication is

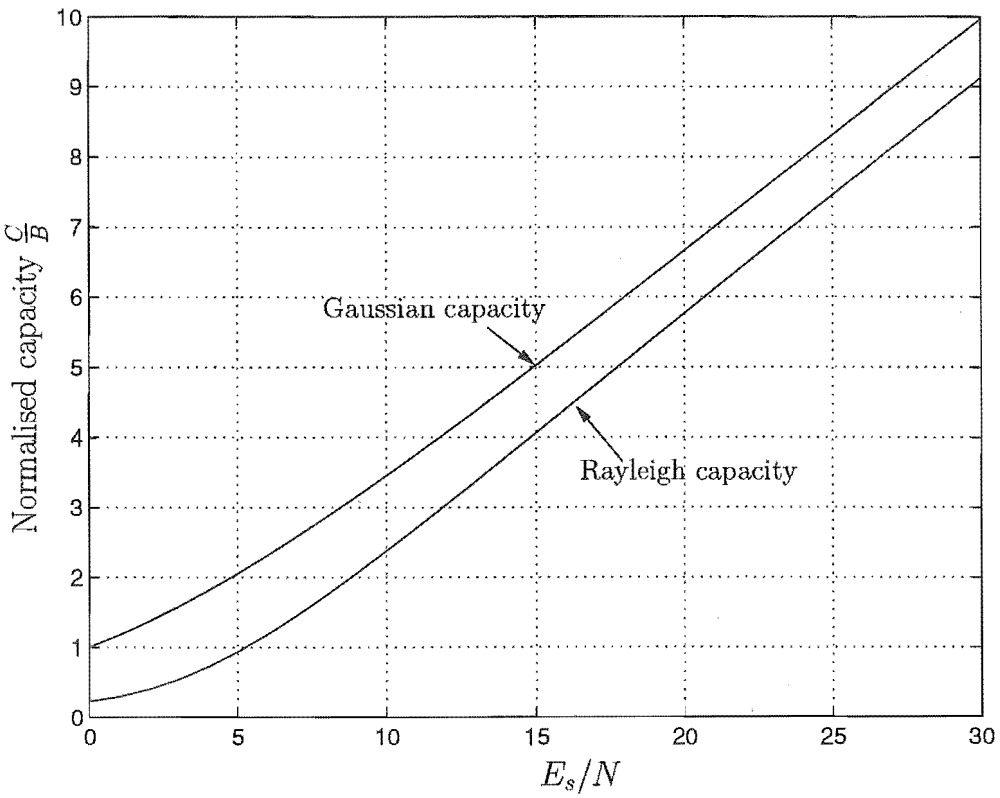


Figure 1.4: Channel capacity in the Gaussian and Rayleigh fading channels.

not possible at any bandwidth. The Rayleigh fading channel may be thought of as a conventional AWGN channel with a time varying signal-to-noise ratio (SNR) that is

Rayleigh distributed. A capacity on the Rayleigh fading channel may be obtained by averaging the instantaneous AWGN capacity over the Rayleigh distribution. W.C.Y. Lee performed this derivation resulting in the following average capacity of the Rayleigh fading channel for $\Gamma > 2$ [46]:

$$\bar{C} = B \log_2 e \cdot e^{-1/\Gamma}(-\gamma + \ln \Gamma + \frac{1}{\Gamma}). \tag{1.19}$$

where γ is Euler's constant given by $\gamma = 0.5772\ldots$. In figure 1.4 we have plotted the normalised channel capacity bounds of equations (1.17) and (1.19) of the AWGN and Rayleigh fading channels respectively. We see that the SNR required on the Rayleigh fading channel to achieve the same normalised capacity as the Gaussian channel is only about 2.5 dB higher for large SNR. For an unlimited bandwidth B the capacity of the Rayleigh fading channel is

$$\lim_{B\to\infty} \bar{C} = \log_2 e \frac{E_s}{N_0} \approx 1.44 \frac{E_s}{N_0} \tag{1.20}$$

and is the same as that for the Gaussian channel.

1.3 Uncoded performance.

We will compute here the probability of error of uncoded BPSK transmitted across the Rayleigh fading channel. This is a one-dimensional modulation and the two signal points with average energy E_s are drawn in the IQ plane in figure 1.5. By

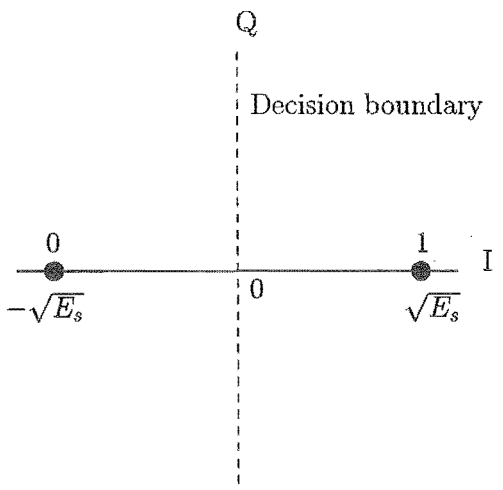


Figure 1.5: A BPSK signal constellation in the IQ plane.

symmetry the decision boundary is at $r = 0$. If we assume coherent detection then the received point r is given by

$$r = u + n \tag{1.21}$$

where u is the transmitted point multiplied by the Rayleigh fading process random variable and n is additive white Gaussian noise with variance $N_0/2$. The probability density function of u is given by

$$f(u) = \frac{u}{E_s} e^{-u^2/2E_s}, \quad u \geq 0 \tag{1.22}$$

and the pdf of n is

$$f(n) = \frac{1}{\sqrt{\pi N_0}} e^{-n^2/N_0} \tag{1.23}$$

The probability density function of r is the convolution of the pdf's of u and n and is given by

$$\begin{aligned} f(r) &= \int_{-\infty}^{\infty} u(x)n(r-x)dx \\ &= \frac{E_s}{\sqrt{\pi N_0}} \int_0^{\infty} x e^{-x^2/2E_s} e^{-x^2/N_0} dx \end{aligned} \tag{1.24}$$

The probability of confusing the signal point 0 for signal point 1 is the probability that $r < 0$ and is

$$\begin{aligned} p(r<0) &= \int_{-\infty}^{0} f(r)dr \\ &= \frac{E_s}{\sqrt{\pi N_0}} \int_{-\infty}^{0} \int_0^{\infty} x e^{-x^2/2E_s} e^{-x^2/N_0} dx dr \\ &= \frac{1}{2E_s} \int_0^{\infty} x e^{-x^2/E_s} - x\text{erf}(\frac{x}{\sqrt{N_0}}) e^{-x^2/2E_s} dx \\ &= \frac{1 + 2E_s/N_0 - \sqrt{2Es/N_0 + 4(E_s/N_0)^2}}{2 + 4E_s/N_0} \\ &\leq \frac{1}{4E_s/N_0} \end{aligned} \tag{1.25}$$

which shows that the probability of error of an uncoded binary PSK system is inversely proportional to the SNR of the system. In comparison with the AWGN transmitting BPSK the probability of error is given by [31]

$$p_{err} = \frac{1}{2}\text{erfc}\left(\sqrt{\frac{E_s}{N_0}}\right) \tag{1.26}$$

and decreases exponentially with increasing SNR. In figure 1.6 we have plotted the probability of error curves of coherent BPSK operating on the AWGN channel and Rayleigh fading channel. Drawn also are the capacity limits for error-free transmission of 1 bit/symbol/Hertz for one-dimensional modulation obtained from equations (1.17) and (1.19) for the AWGN and Rayleigh fading channels respectively. Note that equations (1.17) and (1.19) need to be halved for a one-dimensional modulation. From figure 1.6, uncoded BPSK operating on the Rayleigh fading channel performs

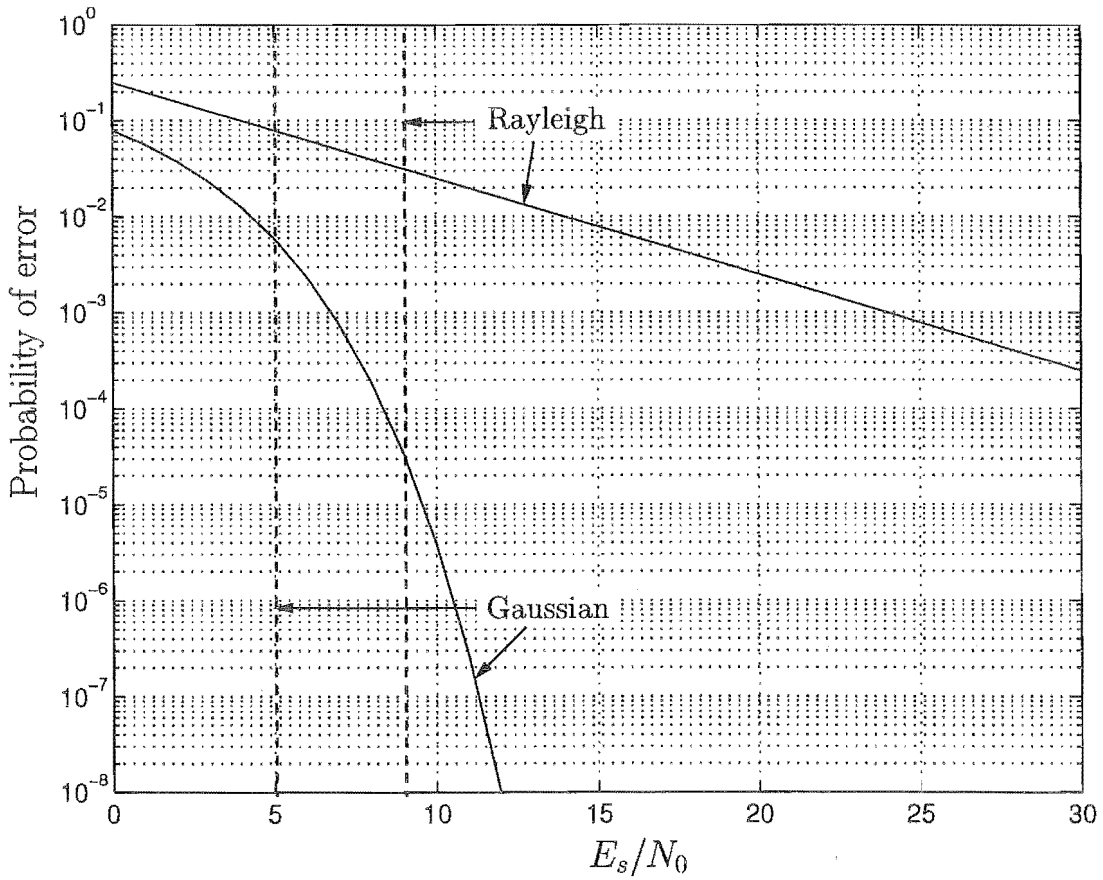


Figure 1.6: Probabilities of error of uncoded coherent BPSK on the Gaussian and Rayleigh fading channels. The dashed lines show the capacity of each channel.

extremely poorly compared to the theoretical channel capacity bound. At an error rate of 10^{-4} the uncoded performance is over 22 dB away from the capacity compared to about 4 dB for the Gaussian channel. It has now been well established that capacity can almost be achieved for the Gaussian channel by the trellis coding techniques pioneered by Ungerboeck [75]. Clearly the potential for coding techniques

for the Rayleigh fading channel is enormous.

1.4 Summary.

We have presented the model of the Rayleigh fading channel. Rayleigh fading is a multiplicative interference characterised by quasi-periodic deep fades in the strength of the received signal. The speed of variation of the process is a function of the speed of the mobile and is normalised to the symbol rate to give the normalised fade rate parameter $f_D T$. The work by W.C.Y. Lee derives the capacity of the Rayleigh fading channel and we have shown that simple uncoded modulation performs extremely poorly compared to the theoretical capacity of the channel. This result justifies the need for investigation into superior methods of signal transmission and detection.

Chapter 2

Maximum likelihood sequence estimation.

In section 1.3 of chapter 1 we discussed the performance of an uncoded system operating on the Rayleigh fading channel making symbol by symbol decisions. The difference between the performances of coherently detected PSK and the theoretical channel capacity is not only large, but also rapidly diverging with increasing SNR. Clearly more sophisticated methods of decoding need to be considered. Maximum likelihood sequence estimation (MLSE) is a technique which makes a decision based on as much of the information available at the receiver as possible. This information typically consists of the received signal, the statistics of the random processes affecting it, and the set of possible transmitted sequences (the code). Generally it is assumed the codeword *a priori* probability of transmission is equiprobable. Divsalar and Simon [22] first derived a ML decoder for the Rayleigh fading channel assuming ideal channel state information (CSI), and infinite interleaving and provided an upperbound on the pairwise error performance of the decoder. The work by Cavers and Ho [13] extended the ML decoder to include a channel state estimate, but still assumed infinite interleaving. Their derivation resulted in an exact expression for the pairwise probability of error of the system. Ho and Fung considered the effect of finite interleaving combined with differential detection to establish a channel estimate [32]. Other authors considered other special cases, some of which in combination with trellis coding [15],[18],[25], [27],[70]. In this chapter we derive a generalised form of a MLSE decoder for the Rayleigh fading channel. The model takes into ac-

count the amplitude and correlation statistics of the multiplicative Rayleigh fading process as described in chapter 1, the additive white Gaussian noise (AWGN), and assumes an unbiased, noisy channel state estimate is available at the receiver. We then derive an exact analytical expression on the pairwise probability of error and analyse this expression further to derive tight upperbounds on the error performance for a number of special cases.

2.1 System model.

The block diagram of the system model used in this study is shown in figure 2.1. The input to the system is a sequence of binary digits to be transmitted, denoted $\mathbf{a} = (a_1, a_2, \ldots, a_n)$, where each $a_k \in \{0, 1\}$. For the purposes of the model it is assumed that the binary stream is independent and identically distributed. This means the probabilities of a 0 or 1 occurring are the same and the outcome is uncorrelated with the previous input history. Real world data typically is correlated and the redundancy may be removed by an appropriate compression algorithm such as Huffman coding [34]. The encoder maps the sequence $\mathbf{a}$ to a sequence $\mathbf{s}_i$ of symbols drawn from a signal constellation S. The constellation S may be multi-dimensional. The sequence $\mathbf{s}_i$ is mapped onto a sequence of channel symbols $\mathbf{c}_i = (c_{i1}, c_{i2}, \ldots, c_{iL})$, where each c_{ik} is a point in the complex plane and is normalised such that the expectation $\frac{1}{2}E[|c_{ik}|^2]$ over all codewords equals 1. The normalisation ensures the average energy of the channel symbols is unity and allows for fair comparison of the distance properties of the underlying signal constellation. Typical encoding schemes are trellis codes, block codes, and multi-level codes. Special forms of trellis and multi-level coding appropriate for the Rayleigh fading channel are discussed in chapters 4, 6 and 7. The sequence of baseband symbols $\mathbf{c}_i$ is modulated onto the carrier frequency to give the base band equivalent time-domain signal

$$c(t) = \sum_{k=1}^{L} c_{ik} p(t - kT) \tag{2.1}$$

where T is the symbol period and $p(t)$ is the complex impulse response of a pulse shaping filter that satisfies the Nyquist criterion for zero inter-symbol-interference (ISI). The energy of the pulse is normalised such that

$$\int_{-\infty}^{\infty} |p(t)|^2 dt = 1. \tag{2.2}$$

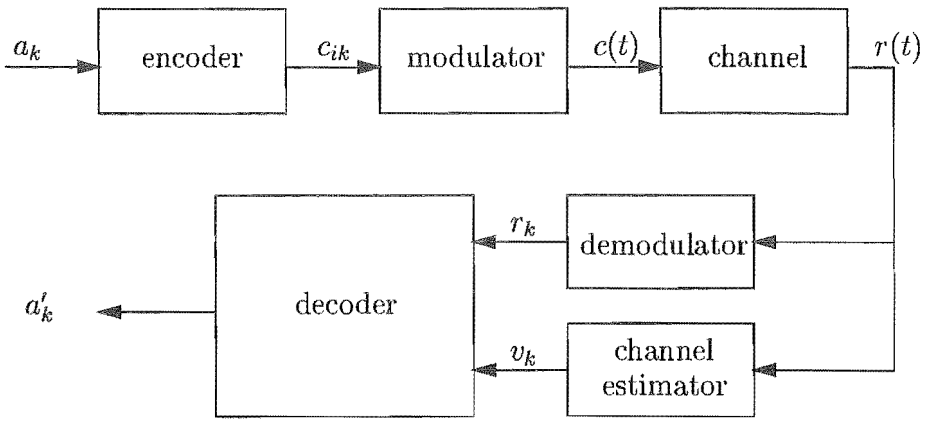


Figure 2.1: System model block diagram.

The received signal

$$r(t) = c(t)u(t) + n(t) \tag{2.3}$$

is described by the channel model of chapter 1 and is passed through a filter matched to the pulse $p(t)$ and sampled every T seconds to produce the output sequence $\mathbf{r} = (r_1, r_2, \ldots, r_L)$. Details of optimal pulse shaping and filtering techniques are subjects in themselves and are not considered. Each sample r_k can be described in terms of the corresponding transmitted symbol c_{ik} by

$$r_k = c_{ik}u_k + n_k. \tag{2.4}$$

The n_k's are statistically independent identically distributed complex Gaussian variables with a variance,

$$\sigma_n^2 = \frac{1}{2}E[|n_k|^2] = N_0, \tag{2.5}$$

the one-sided power spectral density of the additive white Gaussian noise. The u_k's are a sequence of correlated, zero mean, complex Gaussian random variables and represent the fading process experienced by the transmitted sequence. The variance of the u_k's is

$$\sigma_u^2 = \frac{1}{2}E[|u_k|^2] = E_s. \tag{2.6}$$

The ratio E_s/N_0 represents the average signal-to-noise ratio (SNR) of the signal. As described in chapter 1 the amplitude of the complex Gaussian variables u_k is Rayleigh distributed and the autocorrelation of the u_k's is

$$\rho(m) = \frac{1}{2}E[u_k u_{k+m}^*] = E_s J_0(2\pi m f_D T), \tag{2.7}$$

where f_D is the maximum Doppler frequency, T is the symbol period, and $J_0(\bullet)$ is the zero order Bessel function. The quantity $f_D T$ is called the *normalised Doppler frequency* and is a measure of the ratio of the maximum Doppler shift f_D experienced by the mobile's carrier frequency to the symbol rate. The channel estimator of figure 2.1 extracts information about the complex gain u_k of the channel from the received signal $r(t)$. The output of the estimator is assumed to be a sequence of zero mean complex Gaussian random variables, $\mathbf{v} = (v_1, v_2, \ldots, v_L)$, which estimate the true channel state sequence $\mathbf{u} = (u_1, u_2, \ldots, u_L)$. No attempt is made in the present analysis to specify a method of channel estimation. This allows the analysis to be used for a broad class of channel estimators.

The received signal sequence, $\mathbf{r} = (r_1, r_2, \ldots, r_L)$, and the channel state estimate, $\mathbf{v} = (v_1, v_2, \ldots, v_L)$, are input to a maximum likelihood decoder which selects the most probable estimate $\mathbf{c}_j$ of the transmitted sequence $\mathbf{c}_i$. The metric for such a decoder is derived in the following section.

2.2 The maximum likelihood decoder.

Let the set of all possible codewords of the encoding scheme be denoted by

$$\{\mathbf{c}_j = (c_{j1}, c_{j2}, \ldots, c_{jL})\}. \tag{2.8}$$

A maximum likelihood decoder will select the code sequence $\mathbf{c}_j$ for which the *a posteriori* probability $P(\mathbf{c}_j|\mathbf{r}, \mathbf{v})$ is the largest, i.e. given the received signal sequence $\mathbf{r}$ and the channel state estimate sequence $\mathbf{v}$, the decoder chooses the sequence $\mathbf{c}_j$ which is the most likely to have been transmitted. This is equivalent to choosing the codeword $\mathbf{c}_j$ with the largest conditional probability density function,

$$p(\mathbf{r}, \mathbf{v}|\mathbf{c}_j). \tag{2.9}$$

To proceed, we write the received vector $\mathbf{r}$ in matrix form as

$$R = C_j U + N, \tag{2.10}$$

where the $L \times 1$ matrix

$$R = (r_1, r_2, \ldots, r_L)^{\mathrm{T}}, \tag{2.11}$$

now represents the received sequence (the symbol T is the matrix transpose operator). The transmitted sequence is represented by the $L \times L$ matrix

$$C_j = \begin{pmatrix} c_{j1} & & & \\ & c_{j2} & & \\ & & \ddots & \\ & & & c_{jL} \end{pmatrix}, \tag{2.12}$$

where the off-diagonal entries are zero. The channel state vector $\mathbf{u}$, the channel state estimate vector $\mathbf{v}$, and the noise vector $\mathbf{n}$ are similarly represented by the matrices U, V and N respectively.

$$U = (u_1, u_2, \ldots, u_L)^{\mathrm{T}}, \tag{2.13}$$

$$V = (v_1, v_2, \ldots, v_L)^{\mathrm{T}}, \tag{2.14}$$

and

$$N = (n_1, n_2, \ldots, n_L)^{\mathrm{T}}. \tag{2.15}$$

The length L vectors, U, V and N, of zero mean complex Gaussian variables have associated with them the auto-correlation and cross-correlation matrices, Φ_{UU}, Φ_{NN}, Φ_{VV}, and Φ_{UV}, Φ_{VN} respectively. The matrices Φ_{VV}, Φ_{UV}, and Φ_{VN} are dependent on the method of channel estimation and are derived for the cases of ideal CSI in section 2.4.1, and pilot tone aided detection in section 2.7. The matrices Φ_{UU} and Φ_{NN} do not depend on the method of channel state estimation. The element in the i^{th} row and j^{th} column of Φ_{UU} represents the autocorrelation of the channel's fading process. These values are found from equation (2.7) to give the real Toeplitz matrix

$$\Phi_{UU} = \begin{pmatrix} \rho(0) & \rho(1) & \cdots & \rho(L-1) \\ \rho(1) & \rho(0) & \cdots & \rho(L-2) \\ \vdots & \vdots & \ddots & \vdots \\ \rho(L-1) & \rho(L-2) & \cdots & \rho(0) \end{pmatrix}. \tag{2.16}$$

The additive noise terms n_k of equation (2.4) are assumed to be statistically independent and identically distributed complex Gaussian variables with the autocorrelation matrix

$$\Phi_{NN} = N_0 I \tag{2.17}$$

where I is the $L \times L$ identity matrix, and N_0 is the variance of n_k as defined in equation (2.5). We want to find the joint conditional probability density function $p(R, V|\mathbf{c}_j)$, of R and V. This is equivalent to finding the probability density function (pdf) of the vector of $2L$ random variables

$$W = \begin{pmatrix} R \\ V \end{pmatrix} \tag{2.18}$$

From equation (2.10), for fixed $\mathbf{c}_j$, R is a vector of zero-mean Gaussian variables. By assumption the channel estimate V is a zero mean Gaussian variable and therefore W is a vector of zero-mean Gaussian variables. The general form of a zero-mean multi-variate complex Gaussian probability density function is [51]

$$p(W|\mathbf{c}_j) = \frac{e^{-\frac{1}{2}W^\dagger \Phi_{WW}^{-1}|\mathbf{c}_j W}}{(2\pi)^{2L}\det(\Phi_{WW}|\mathbf{c}_j)}, \tag{2.19}$$

where $\dagger$ denotes the Hermitian transpose. The matrix

$$\Phi_{WW}|\mathbf{c}_j = \begin{pmatrix} \Phi_{RR}|\mathbf{c}_j & \Phi_{RV}|\mathbf{c}_j \\ \Phi_{VR}|\mathbf{c}_j & \Phi_{VV}|\mathbf{c}_j \end{pmatrix} \tag{2.20}$$

is the autocorrelation of W. The autocorrelation of R is

$$\begin{aligned} \Phi_{RR}|\mathbf{c}_j &= \frac{1}{2}E[RR^\dagger] \\ &= \frac{1}{2}E[(C_jU + N)(C_jU + N)^\dagger] \\ &= C_j\Phi_{UU}C_j^\dagger + \Phi_{NN} \end{aligned} \tag{2.21}$$

since U and N are uncorrelated. The cross correlation of R and V is

$$\begin{aligned} \Phi_{RV}|\mathbf{c}_j &= \frac{1}{2}E[RV^\dagger] \\ &= \frac{1}{2}E[(C_jU + N)V^\dagger] \\ &= C_j\Phi_{UV}|\mathbf{c}_j + \Phi_{NV}|\mathbf{c}_j \end{aligned} \tag{2.22}$$

A maximum likelihood receiver is one which selects the codeword $\mathbf{c}_j$ which maximises the conditional density of equation (2.19) for the received vector R and channel estimate vector V. Using the property that the exponential function of equation (2.19) decreases monotonically, we can take the natural log, and arrive at an equivalent decoding metric

$$M_d(\mathbf{c}_j) \;=\; W^\dagger\Phi_{WW}^{-1}|\mathbf{c}_j W + 2\ln\det(\Phi_{WW}|\mathbf{c}_j). \tag{2.23}$$

The maximum likelihood decoder selects the codeword $\mathbf{c}_j$ for which the metric of equation (2.23) is a minimum.

2.3 The pairwise error event probability.

Let the transmitted codeword be $\mathbf{c}_i$. The decoder will pick the erroneous codeword $\mathbf{c}_j$ if

$$M_d(\mathbf{c}_j) < M_d(\mathbf{c}_i), \tag{2.24}$$

or equivalently if

$$W^\dagger \Phi_{WW}^{-1}|\mathbf{c}_j W + 2\ln\det(\Phi_{WW}|\mathbf{c}_j) < W^\dagger \Phi_{WW}^{-1}|\mathbf{c}_i W + 2\ln\det(\Phi_{WW}|\mathbf{c}_i) \tag{2.25}$$

where M_d is the decoding metric of equation (2.23). The probability of an error event can be written as

$$P(D < \delta) \tag{2.26}$$

where the decision variable D is

$$D = W^\dagger(\Phi_{WW}^{-1}|\mathbf{c}_j - \Phi_{WW}^{-1}|\mathbf{c}_i)W \tag{2.27}$$

The parameter δ is defined as

$$\delta = 2\ln\frac{\det(\Phi_{WW}|\mathbf{c}_i)}{\det(\Phi_{WW}|\mathbf{c}_j)}, \tag{2.28}$$

and corresponds to a decision threshold. We can write the decision variable D as the quadratic form

$$D = W^\dagger F W \tag{2.29}$$

where

$$F = \Phi_{WW}^{-1}|\mathbf{c}_j - \Phi_{WW}^{-1}|\mathbf{c}_i \tag{2.30}$$

Note that $\Phi_{WW}^{-1}|\mathbf{c}_i$ and $\Phi_{WW}^{-1}|\mathbf{c}_j$ are Hermitian matrices and F is Hermitian also. The result of [63, Appendix B] can now be applied to express the two sided Laplace transform characteristic function of the decision variable D as

$$\begin{aligned}\Phi_D(s) &= \frac{1}{\det(I + 2s\Phi_{WW}|\mathbf{c}_i F)} \\ &= \frac{1}{\det(I + 2s(\Phi_{WW}|\mathbf{c}_i \Phi_{WW}^{-1}|\mathbf{c}_j - I))}\end{aligned} \tag{2.31}$$

where $\Phi_{WW}|\mathbf{c}_i$ and $\Phi_{WW}|\mathbf{c}_j$ are found from equation (2.20). The pairwise event probability can be found by the appropriate integration of the inverse Laplace transform of $\Phi_D(s)$ (which gives the probability density function of D). However, following [13] it is simpler to calculate

$$P(\mathbf{c}_i \to \mathbf{c}_j) = \begin{cases} -\sum \text{Residue}\left[e^{\delta s}\Phi_D(s)/s\right]_{RP_{\text{poles}}}, & \delta \leq 0 \\ \sum \text{Residue}\left[e^{\delta s}\Phi_D(s)/s\right]_{LP_{\text{poles}}}, & \delta > 0 \end{cases} \tag{2.32}$$

instead. The notation RP_{poles} and LP_{poles} refers to the right hand plane poles and left hand plane poles of $\Phi_D(s)$ respectively. To calculate (2.32) it is useful to note that the $2L$ poles p_i, of equation (2.31), are related to the eigenvalues λ_i of $\Phi_{WW}F$ by

$$p_i = \frac{-1}{2\lambda_i} \tag{2.33}$$

and equation (2.31) can be written as

$$\Phi_D(s) = \prod_{i=1}^{2L} \frac{-p_i}{s - p_i}. \tag{2.34}$$

If $\lambda_i = 0$ then

$$\frac{-p_i}{s - p_i} = 1 \tag{2.35}$$

and the number of poles of $\Phi_D(s)$ is reduced accordingly.

2.4 Channel estimation methods.

In chapter 1 we explained the need for the acquisition of channel state information for the purpose of decoding the incoming signal. In section 2.4.1 we analyse the performance of the system on the assumption that ideal channel state information is available, i.e. the receiver knows the exact state of the channel at any time. Although this is an unrealistic assumption, it is useful for determining a limit on the performance of the system to which practical techniques may be compared. The common techniques used for estimating the channel state in practice are pilot tones, pilot symbols and differential detection. Pilot tones are analysed in section 2.7 while chapter 7 is devoted to the analysis of differential detection. Pilot symbols have been shown to perform similarly to pilot tones and are not considered here [15].

2.4.1 Ideal channel state information (CSI).

The assumption of ideal channel state information allows many simplifications of the general decoding metric for the Rayleigh fading channel derived in section 2.2. The generalised metric reproduced from equation (2.23) is

$$M_d(\mathbf{c}_j) = W^\dagger \Phi_{WW}^{-1}|\mathbf{c}_j W + 2\ln\det(\Phi_{WW}|\mathbf{c}_j). \tag{2.36}$$

With ideal CSI the receiver has perfect knowledge of the channel, and hence

$$V = U \tag{2.37}$$

In this case, the correlation matrices dependent on the method of channel estimation are simply

$$\Phi_{VV}|\mathbf{c}_j = \Phi_{UU}, \tag{2.38}$$

$$\Phi_{UV}|\mathbf{c}_j = \Phi_{UU}, \tag{2.39}$$

$$\Phi_{NV}|\mathbf{c}_j = 0 \tag{2.40}$$

With these values, the autocorrelation matrix of W required for the decoding metric reduces to

$$\Phi_{WW}|\mathbf{c}_j = \begin{pmatrix} C_j\Phi_{UU}C_j^\dagger + \Phi_{NN} & C_j\Phi_{UU} \\ \Phi_{UU}C_j^\dagger & \Phi_{UU} \end{pmatrix} \tag{2.41}$$

and the complexity of the decoding metric of equation (2.36) reduces significantly. Simplifying the second term:

$$\begin{aligned} 2\ln\det(\Phi_{WW}|\mathbf{c}_j) &= 2\ln\det\begin{pmatrix} C_j\Phi_{UU}C_j^\dagger + \Phi_{NN} & C_j\Phi_{UU} \\ \Phi_{UU}C_j^\dagger & \Phi_{UU} \end{pmatrix} \\ &= 2\ln\left\{\det\begin{pmatrix} C_j & I \\ I & 0 \end{pmatrix}\det\begin{pmatrix} \Phi_{UU} & 0 \\ 0 & \Phi_{NN} \end{pmatrix}\det\begin{pmatrix} C_j^\dagger & I \\ I & 0 \end{pmatrix}\right\} \end{aligned} \tag{2.42}$$

but

$$\det\begin{pmatrix} C_j & I \\ I & 0 \end{pmatrix} = -1 \tag{2.43}$$

and

$$\det \begin{pmatrix} \Phi_{UU} & 0 \\ 0 & \Phi_{NN} \end{pmatrix} = N_0^L \det \Phi_{UU} \tag{2.44}$$

hence

$$2 \ln \det \Phi_{WW}|\mathbf{c}_j = 2L \ln(N_0^L \det \Phi_{UU}) \tag{2.45}$$

which is independent of $\mathbf{c}_j$ and the term $2 \ln \det(\Phi_{WW}|\mathbf{c}_j)$ may be omitted. The first term of the metric reduces as

$$\begin{aligned} W^\dagger \Phi_{WW}^{-1}|\mathbf{c}_j W &= \begin{pmatrix} R^\dagger & V^\dagger \end{pmatrix} \begin{pmatrix} C_j \Phi_{UU} C_j^\dagger + \Phi_{NN} & C_j \Phi_{UU} \\ \Phi_{UU} C_j^\dagger & \Phi_{UU} \end{pmatrix}^{-1} \begin{pmatrix} R \\ V \end{pmatrix} \\ &= \begin{pmatrix} R^\dagger & V^\dagger \end{pmatrix} \begin{pmatrix} \Phi_{NN}^{-1} & -\Phi_{NN}^{-1} C_j \\ -C_j^\dagger \Phi_{NN}^{-1} & \Phi_{UU}^{-1} + C_j \Phi_{NN}^{-1} C_j^\dagger \end{pmatrix} \begin{pmatrix} R \\ V \end{pmatrix} \end{aligned} \tag{2.46}$$

where the inverse has been found using the matrix identity described in appendix A. Expanding the matrix product gives

$$\begin{aligned} W^\dagger \Phi_{WW}^{-1}|\mathbf{c}_j W &= \frac{1}{N_0}(R^\dagger R - R^\dagger C_j V - V^\dagger C_j^\dagger R + V^\dagger C_j^\dagger C_j V) + V^\dagger \Phi_{UU}^{-1} V \\ &= \frac{1}{N_0}(R - C_j V)^\dagger (R - C_j V) + V^\dagger \Phi_{UU}^{-1} V \end{aligned} \tag{2.47}$$

The last term and the factor $\frac{1}{N_0}$ are independent of $\mathbf{c}_j$ and may be discarded to give the decoding metric for ideal CSI:

$$\begin{aligned} M_d(\mathbf{c}_j) &= (R - C_j V)^\dagger (R - C_j V) \\ &= \sum_{k=1}^{L} |r_k - v_k c_{jk}|^2, \end{aligned} \tag{2.48}$$

i.e. simply a Euclidean distance metric which is independent of $f_D T$ and is suitable for use with the Viterbi decoding algorithm. Note that with ideal channel state information, the knowledge of the correlation of the channel does not aid in the decision process.

Ideal CSI error performance.

We examine the variation of the error performance of an ideal CSI system as a function of the sequence of symbols, SNR and the fade rate $f_D T$. Equation (2.32) for

the evaluation of the pairwise probability of error simplifies. The decision threshold parameter δ equals zero since the determinants of $\Phi_{WW}|\mathbf{c}_i$ and $\Phi_{WW}|\mathbf{c}_j$ are equal as shown by equation (2.45). The matrix F of equation (2.30) reduces to

$$F = \begin{pmatrix} \mathbf{0} & \Phi_{NN}^{-1}(C_i - C_j) \\ (C_i^\dagger - C_j^\dagger)\Phi_{NN}^{-1} & C_j\Phi_{NN}^{-1}C_j^\dagger - C_i\Phi_{NN}^{-1}C_i^\dagger \end{pmatrix} \tag{2.49}$$

and the matrix $\Phi_{WW}|\mathbf{c}_j$ is as described by equation (2.41). The pairwise probability of error is given by

$$P(\mathbf{c}_i \to \mathbf{c}_j) = -\sum \text{Residue}\left[\Phi_D(s)/s\right]_{RP_{\text{poles}}} \tag{2.50}$$

where the poles of $\Phi_D(s)$ are related to the eigenvalues of $\Phi_{WW}|\mathbf{c}_j F$ by equation (2.33). From equation (2.48) it is clear that if $c_{ik} = c_{jk}$, no contribution is made towards the decision variable D of equation (2.27). This means for every $c_{ik} = c_{jk}$, $\Phi_{WW}F$ has a pair of zero eigenvalues and the number of poles of equation (2.50) is reduced by two. Hence the number of pairs of poles of $\Phi_D(s)$ is equal to the number of symbols different between the sequences $\mathbf{c}_i$ and $\mathbf{c}_j$. This number is simply the Hamming distance between $\mathbf{c}_i$ and $\mathbf{c}_j$ and is denoted by the symbol l. We define η to be the set of indices k such that $c_{ik} \neq c_{jk}$, i.e.

$$\eta = \{k : c_{ik} \neq c_{jk}, 1 \leq k \leq L\} \tag{2.51}$$

and the order of η is

$$|\eta| = l. \tag{2.52}$$

As an example of the behaviour of the pairwise error probability with ideal CSI, consider two message sequences drawn from a 4-PSK constellation, the all-zeroes sequence

$$\mathbf{s}_i = \{0, 0, \ldots, 0\} \tag{2.53}$$

and a sequence of the following form

$$\mathbf{s}_j = \{s_1, 0, \ldots, 0, s_L\} \tag{2.54}$$

i.e. the sequence consists of a non-zero start symbol s_1, followed by $L-2$ zeroes and a non-zero end symbol s_L. The 4-PSK sequences $\mathbf{s}_i$ and $\mathbf{s}_j$ map onto the channel

symbol sequences $\mathbf{c}_i$ and $\mathbf{c}_j$ respectively as

$$c_{ik} = e^{\frac{j\pi s_{ik}}{2}} \tag{2.55}$$

$$c_{jk} = e^{\frac{j\pi s_{jk}}{2}} \tag{2.56}$$

As discussed, if $c_{ik} = c_{jk}$ then no contribution is made towards the decoding metric, and the dimensionality of the problem is reduced accordingly. The matrices from which the probability of error may be derived are the message sequence matrices

$$C_i = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \tag{2.57}$$

$$C_j = \begin{pmatrix} c_{j1} & 0 \\ 0 & c_{jL} \end{pmatrix} \tag{2.58}$$

and the channel correlation matrix:

$$\Phi_{UU} = E_s \begin{pmatrix} 1 & \rho_\beta \\ \rho_\beta & 1 \end{pmatrix} \tag{2.59}$$

The parameter ρ_β is the correlation between the fading experienced by the symbols c_{i0} and c_{iL} and is related to L and $f_D T$ by

$$\rho_\beta = J_0(2\pi\beta f_D T) \tag{2.60}$$

where $\beta = L - 1$, the distance between non-equal symbols in the sequences. Figure 2.2 is a plot of the probability of the decoder confusing the sequence $\mathbf{c}_i$ for the sequence $\mathbf{c}_j$ as a function of the SNR for $\rho_\beta = 1.0$, $\rho_\beta = 0.99$, $\rho_\beta = 0.95$ and $\rho_\beta = 0.0$. A number of important observations can be made. For a value of $\rho_\beta = 1.0$ (i.e. the fading is completely correlated), the probability of error of the system decreases at a rate proportional to the inverse of the SNR, the same at that of an uncoded system (see section 1.3). This case is analysed in detail in section 2.4.1. For a value of $\rho_\beta = 0$ (i.e. there is no correlation between the fading affecting the symbols), the performance improves and, as we show in section 2.4.1, the probability of error now falls off inversely with the *square* of the SNR. For the two values of ρ_β between 0.0 and 1.0 the asymptotic behaviour (i.e. the slope of the curve) with increasing SNR is the same as $\rho_\beta = 0.0$, however there is significant loss with respect to $\rho_\beta = 0.0$. Section 2.5 quantifies this loss analytically.

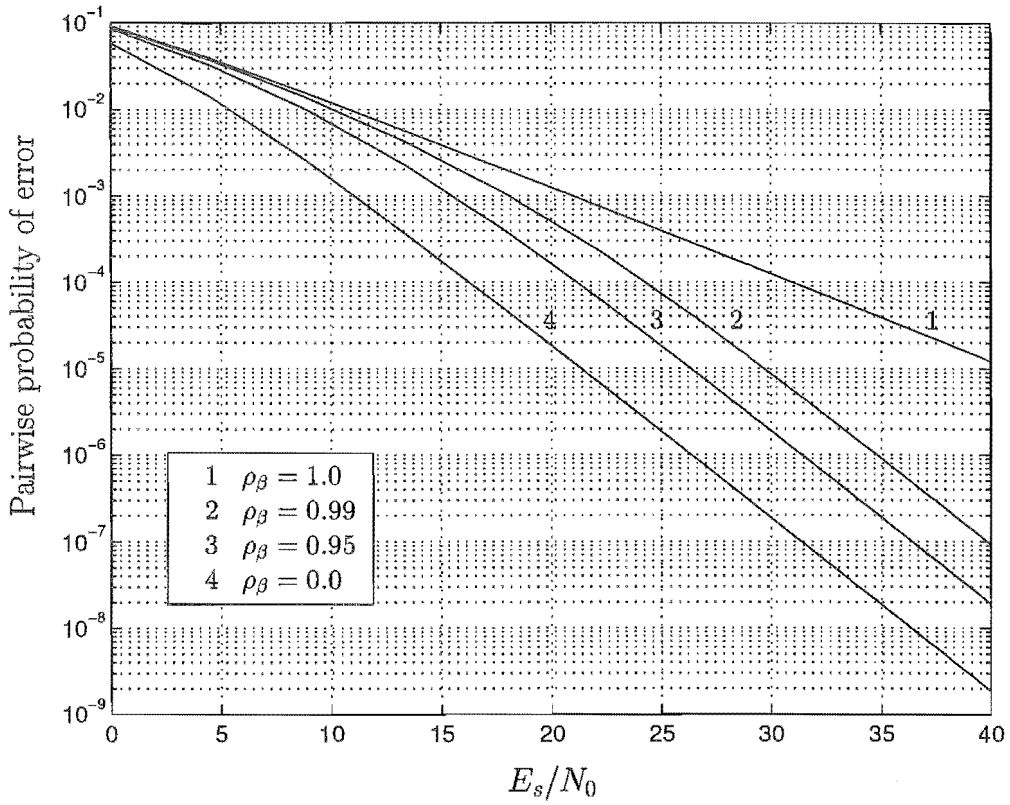


Figure 2.2: Pairwise probability of error versus SNR, for a two symbol sequence, with varying symbol correlation values of ρ_β.

To study the transitional behaviour as ρ_β varies from 0.0 to 1.0, figure 2.3 plots the probability of error as a function of ρ_β for a fixed SNR of 20 dB. Clearly the error probability is minimised for $\rho_\beta = 0.0$ and maximised for $\rho_\beta = 1.0$. Significant improvements are realised for only small deviations away from $\rho_\beta = 1.0$ as is evident in figure 2.2. Alternatively the probability of error may be plotted as a function of $\beta f_D T$ (see figure 2.4), which is related to ρ_β by equation (2.60). The system operates at the two distinct error levels. For values of $\beta f_D T > 3 \times 10^{-1}$ the performance is determined by the inverse of the square of the SNR and for values of $\beta f_D T < 10^{-2}$ the performance is only inversely proportional to the SNR. Clearly it is advantageous to operate the system at high values of $\beta f_D T$. For reasons such as increased ISI, carrier synchronisation problems, and channel state recovery, it is difficult to operate a practical system at high values of $f_D T$ (fast fading). The alternative is to increase the value of β by interleaving as described in section 2.6. The trade-off is an increase in the overall transmission delay of the system. In the next sections we analyse the

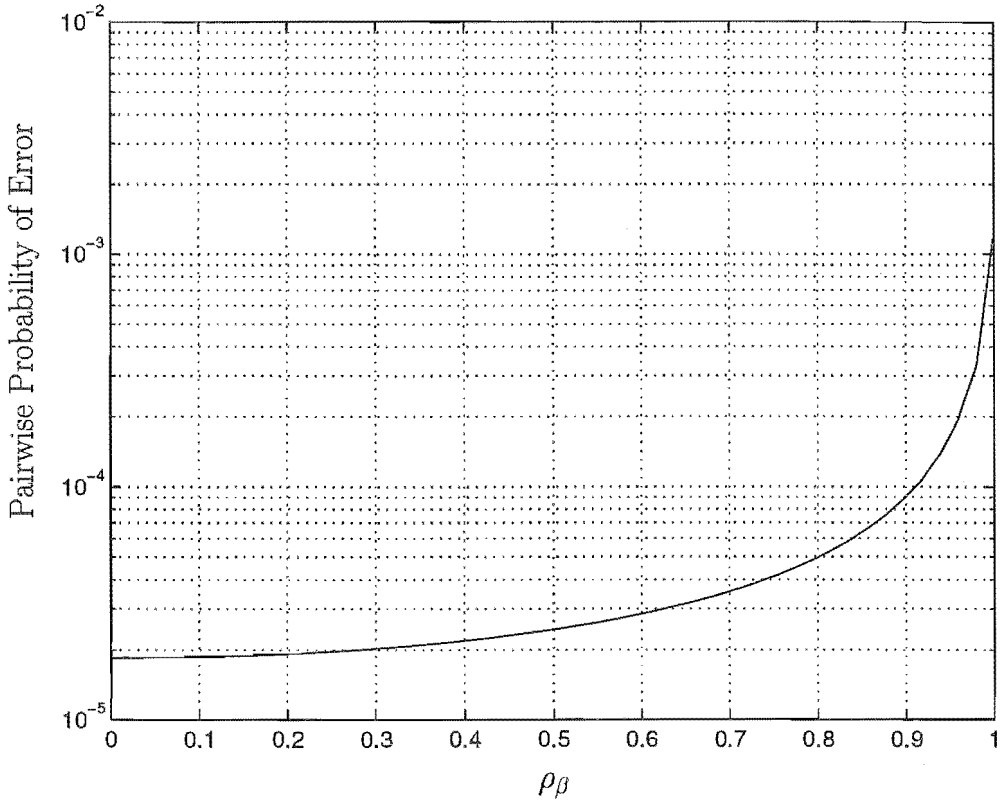


Figure 2.3: Pairwise probability of error as a function of ρ_β of a two symbol sequence. E_s/N_0=20 dB.

ideal CSI system for the cases of highly correlated fading, uncorrelated fading, and partial fading.

Completely correlated fading with ideal CSI.

If the fading is completely correlated, the channel, at any instant, looks like a Gaussian channel where the SNR is drawn from a Rayleigh distribution. We analyse this limiting case in detail, assuming the receiver has ideal CSI. For such a system, the channel correlation matrix Φ_{UU} is

$$\Phi_{UU} = E_s \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{pmatrix}. \tag{2.61}$$

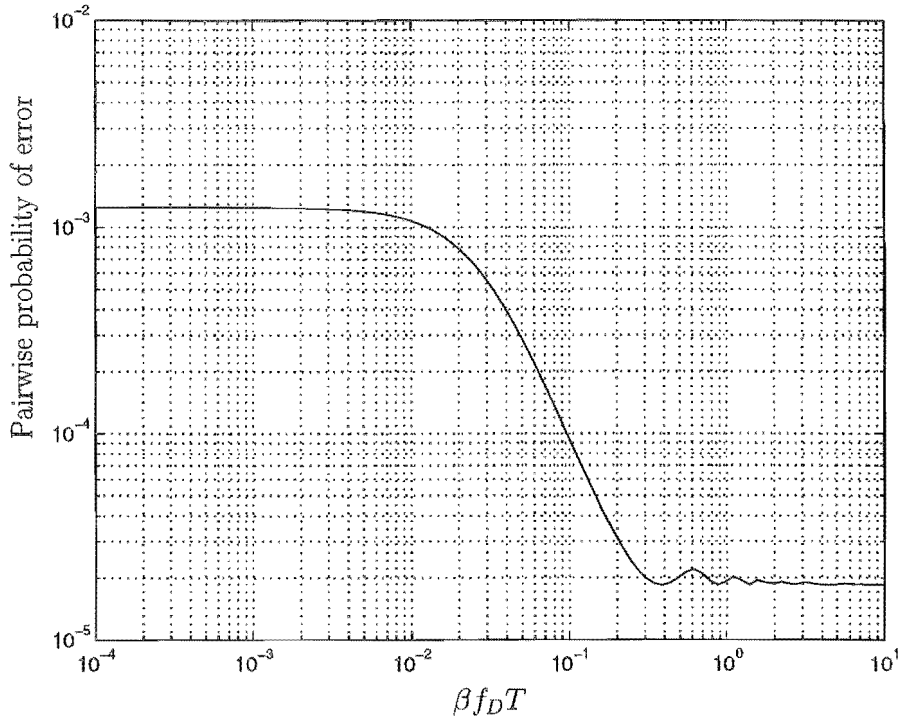


Figure 2.4: Pairwise probability of error as a function of $\beta f_D T$ of a two symbol sequence.

i.e. the fading affects all symbols equally. This is equivalent to a very slowly fading channel where $f_D T \to 0$. Since we are assuming ideal CSI, the ML decoding metric is the Euclidean distance as defined by equation (2.48). To determine the probability of error we examine equation (2.50). After some manipulation we can show that there are only two non-zero eigenvalues of the matrix $\Phi_{WW} F$. They are given by

$$\lambda_{1,2} = \frac{\frac{E_s}{N_0} d^2 \mp \sqrt{\frac{E_s}{N_0} d^2 (\frac{E_s}{N_0} d^2 + 4)}}{2} \tag{2.62}$$

where $d^2 = \sum_{k=1}^{L} |c_{ik} - c_{jk}|^2$, the squared Euclidean distance between codewords $\mathbf{c}_i$ and $\mathbf{c}_j$, and E_s/N_0 is the average signal-to-noise ratio of the system. The remaining eigenvalues are zero, do not contribute to $\Phi_D(s)$, as shown by equation (2.35), and are discarded. Substituting the eigenvalues into equation (2.50) and noting that only $p_1 = \frac{-1}{2\lambda_1}$ lies in the right hand plane, results in the exact expression for the pairwise probability of error for ideal CSI and correlated fading.

$$P(\mathbf{c}_i \to \mathbf{c}_j) = \frac{2}{4 + \frac{E_s}{N_0} d^2 + d\sqrt{(\frac{E_s}{N_0})^2 d^2 + 4\frac{E_s}{N_0}}}, \tag{2.63}$$

For a high SNR we can tightly upper bound the probability of error by

$$P(\mathbf{c}_i \to \mathbf{c}_j) \leq \frac{1}{d^2 \frac{E_s}{N_0}} \tag{2.64}$$

which agrees exactly with the upper bound given by Divsalar and Simon [22, equation (34)] for $K = 0$ and $C = 1$. We see that for very highly correlated fading, the pairwise error event probability only decreases inversely with the signal-to-noise ratio independently of the length of the codewords. The inverse decrease in error probability with increasing SNR is the same as the behaviour of an uncoded system as shown in chapter 1. Next we examine the uncorrelated fading case.

Uncorrelated fading with ideal CSI.

For an uncorrelated fading system, the channel autocorrelation matrix is

$$\Phi_{UU} = E_s \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix} \tag{2.65}$$

i.e. the fading affecting any pair of transmitted symbols is independent. The maximum likelihood decoding metric is Euclidean distance. To calculate the pairwise probability of error we need the eigenvalues of $\Phi_{WW}F$. With some manipulation we can show that the l pairs of non-zero eigenvalues of $\Phi_{WW}F$ are given by

$$\lambda_k, \lambda_{k+L} = \frac{\frac{E_s}{N_0} d_k^2 \mp d_k \sqrt{(\frac{E_s}{N_0})^2 d_k^2 + 4\frac{E_s}{N_0}}}{2}, \quad k \in \eta \tag{2.66}$$

where $d_k^2 = |c_{ik} - c_{jk}|^2$. The remaining eigenvalues for $k \notin \eta$ are zero and are discarded. From equation (2.33) the poles of $\Phi_D(s)$ are

$$p_k, p_{k+L} = \frac{1}{\frac{E_s}{N_0} d_k^2 \pm d_k \sqrt{(\frac{E_s}{N_0})^2 d_k^2 + 4\frac{E_s}{N_0}}}, \quad k \in \eta \tag{2.67}$$

Clearly p_k lies in the right hand plane and p_{k+L} lies in the left hand plane. The pairwise probability of error may be written from equation (2.50) as

$$p(\mathbf{c}_i \to \mathbf{c}_j) = -S_r \prod_{k \in \eta} p_k p_{k+L} \tag{2.68}$$

where

$$S_r = \sum_{i \in \eta} \lim_{s \to p_i} \frac{s - p_i}{s} \prod_{k \in \eta} \frac{1}{(s - p_k)(s - p_{k+L})} \tag{2.69}$$

is the sum of the residues of the poles in the right hand plane. The product of the pair of poles p_k and p_{k+L} is simply

$$p_k p_{k+L} = -\frac{1}{4\frac{E_s}{N_0}d_k^2} \tag{2.70}$$

and now

$$p(\mathbf{c}_i \to \mathbf{c}_j) = -S_r \prod_{k\in\eta} \frac{-1}{4\frac{E_s}{N_0}d_k^2} \tag{2.71}$$

For high SNR, $p_k \to \frac{1}{2}$, and $p_{k+L} \to -0$, and as shown in appendix B, S_r for these limits is upper bounded by

$$S_r \leq \frac{(-4)^l(2l-1)!}{l!(l-1)!} \tag{2.72}$$

where $l = |\eta|$. Combining equations (2.71) and (2.72) gives an upperbound on the pairwise probability of error of a sequence transmitted over an uncorrelated Rayleigh fading channel with perfect CSI as

$$p(\mathbf{c}_i \to \mathbf{c}_j) \leq \frac{(2l-1)!}{l!(l-1)!} \prod_{k\in\eta} \frac{1}{\frac{E_s}{N_0}d_k^2}. \tag{2.73}$$

The pairwise error probability is inversely proportional to the SNR raised to the power of the Hamming distance l, and inversely proportional to the product distance defined by

$$d_p^2 = \prod_{k\in\eta} d_k^2 \tag{2.74}$$

This is a significant improvement over both the cases of uncoded modulation and completely correlated fading. Note that the upperbound by Divsalar and Simon [22], for the identical system, is

$$p(\mathbf{c}_i \to \mathbf{c}_j) \leq 4^l \prod_{k\in\eta} \frac{1}{\frac{E_s}{N_0}d_k^2} \tag{2.75}$$

Using quite a different technique to the one presented here, Dingman [18] derives an upperbound, again for the same system as

$$p(\mathbf{c}_i \to \mathbf{c}_j) \leq \frac{1}{\pi}\int_0^{\frac{\pi}{2}} \sin^{2l}(\phi)d\phi \prod_{k\in\eta} \frac{4}{\frac{E_s}{N_0}d_k^2} \tag{2.76}$$

He did not show however that the closed form expression for the integral is

$$\frac{1}{\pi}\int_0^{\frac{\pi}{2}} \sin^{2l}(\phi)d\phi = \frac{(2l-1)!}{4^l l!(l-1)!} \tag{2.77}$$

which when combined with equation (2.76) gives the same bound as equation (2.73). The bound of equation (2.73) is tighter than that of equation (2.75) which can be shown by induction. Assuming $b_i = \frac{(2i-1)!}{i!(i-1)!} < 4^i$ we will show that $b_{i+1} \leq 4^{i+1}$

$$\begin{aligned} b_i &\leq 4^i \\ \frac{(2i-1)!}{i!(i-1)!} &\leq 4^i \\ \frac{(2i-1)!}{i!(i-1)!} \times \frac{2i(2i+1)}{(i+1)i} &\leq 4^i \times \frac{2i(2i+1)}{(i+1)i} \\ b_{i+1} &\leq 4^i \times (4 - \frac{2}{i+1}) \\ b_{i+1} &\leq 4^{i+1} \end{aligned} \tag{2.78}$$

And for $i = 1$, $b_i = 1 < 4$ hence the inequality is true for all $i \geq 1$ and the bound is tighter than that of equation (2.75). To demonstrate the tightness of the upperbound of equation (2.73) consider 3 pairs of sequences using antipodal signalling, with increasing Hamming distance.

$$\mathbf{c}_i^1 = \{1\}, \ \mathbf{c}_j^1 = \{-1\} \tag{2.79}$$

$$\mathbf{c}_i^2 = \{1, 1\}, \ \mathbf{c}_j^2 = \{-1, -1\} \tag{2.80}$$

$$\mathbf{c}_i^3 = \{1, 1, 1\}, \ \mathbf{c}_j^3 = \{-1, -1, -1\} \tag{2.81}$$

In figure 2.5 we have plotted the exact pairwise probability of error and the corresponding upperbound for the pairs of sequences $\mathbf{c}_i^n$ and $\mathbf{c}_j^n$. The solid lines show the exact pairwise probability for the three pairs of symbol sequences and the corresponding bounds of equation (2.73) are shown in dashed lines. At high SNR the bounds almost precisely match the exact calculations. The case of $n = 1$ represents uncoded BPSK transmission, which has a diversity of one. Clearly higher diversity symbol sequences vastly improve the performance of the system, especially at lower error rates. It must be kept in mind that such gains in performance are only possible by ensuring that the fading affecting the symbols in the transmitted sequence is *uncorrelated*. The next section analyses the relative loss in performance of the system due to non-zero correlation in the channel.

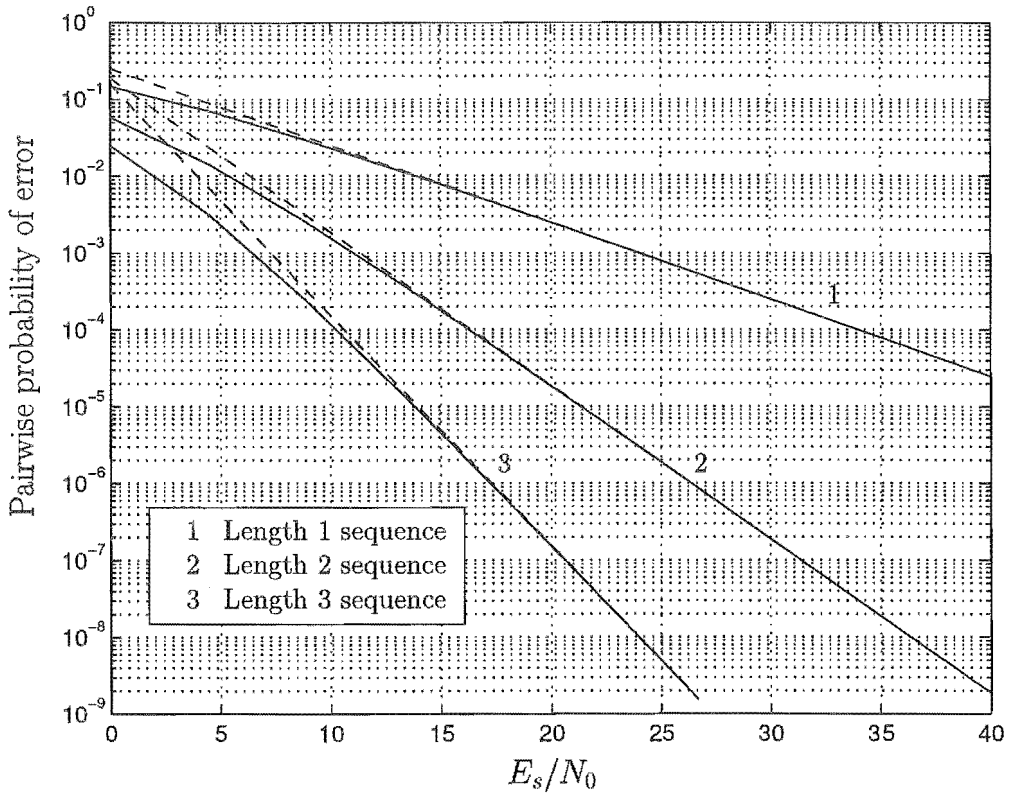


Figure 2.5: Exact pairwise probability of error in solid lines and corresponding upperbounds in dashed lines. Curves 1, 2 and 3 are length 1, 2 and 3 sequences respectively.

2.5 Partially correlated fading with ideal CSI.

In the example of section 2.4.1 we observed that the asymptotic behaviour of a system with some channel correlation is the same as that of an uncorrelated system, but with some loss in performance. We now show that this is indeed the case and quantify this loss analytically for an ideal CSI system. The channel correlation matrix is given by equation (2.16) and reproduced here for convenience:

$$\Phi_{UU} = \begin{pmatrix} \rho(0) & \rho(1) & \cdots & \rho(L-1) \\ \rho(1) & \rho(0) & \cdots & \rho(L-2) \\ \vdots & \vdots & \ddots & \vdots \\ \rho(L-1) & \rho(L-2) & \cdots & \rho(0) \end{pmatrix}. \tag{2.82}$$

The correlation parameters $\rho(n)$ are defined by equation (2.7). The maximum likelihood decision metric is Euclidean distance. For arbitrary SNR, the exact probability of error can only be computed by summing the residues of the right hand

plane poles of $\Phi_{WW}F$ as defined by equation (2.50). For high SNR however, this expression simplifies significantly. Equation (2.50) is written in the form used in the previous section, i.e.

$$p(\mathbf{c}_i \to \mathbf{c}_j) = -S_r \prod_{k\in\eta} p_k p_{k+L} \tag{2.83}$$

The product of the poles is, as shown in appendix C, is given by

$$\prod_{k\in\eta} p_k p_{k+L} = \frac{1}{4^l \det(\Phi'_{WW}F')} \tag{2.84}$$

where Φ'_{WW} and F' correspond to Φ_{WW} and F respectively with the rows and columns i and $i+L$ deleted for $1 \le i \le L$, $i \notin \eta$. From equation (2.49) the determinant of F' is

$$\det F' = \frac{(-1)^l}{N_0^{2l}} \prod_{k\in\eta} |c_{ik} - c_{jk}|^2 \tag{2.85}$$

and from equation (2.45) the determinant of Φ'_{WW} is

$$\det \Phi'_{WW} = (N_0 E_s)^l \det \Phi_\rho \tag{2.86}$$

where

$$\Phi_\rho = \Phi_{UU}/E_s. \tag{2.87}$$

For high SNR, the poles of $\Phi_D(s)$ tend toward $p_k \to \frac{1}{2}$ and $p_{k+L} \to -0$ and combining the equation with the result from appendix B, the probability of error for correlated fading and ideal CSI is bounded by

$$p(\mathbf{c}_i \to \mathbf{c}_j) \le \frac{1}{\det \Phi_\rho} \frac{(2l-1)!}{l!(l-1)!} \prod_{k\in\eta} \frac{1}{\frac{E_s}{N_0} d_k^2} \tag{2.88}$$

which is identical to that of uncorrelated fading divided by $\det \Phi_\rho$. Figure 2.6 shows the application of the bound to the example of section 2.4.1, figure 2.2. The solid lines are the exact calculations, and the dashed lines are the upper bound of equation (2.88).

The loss in performance δ_ρ, measured in dB of SNR, relative to uncorrelated fading is

$$\Delta_\rho = -10 l^{-1} \log_{10} \det \Phi_\rho \tag{2.89}$$

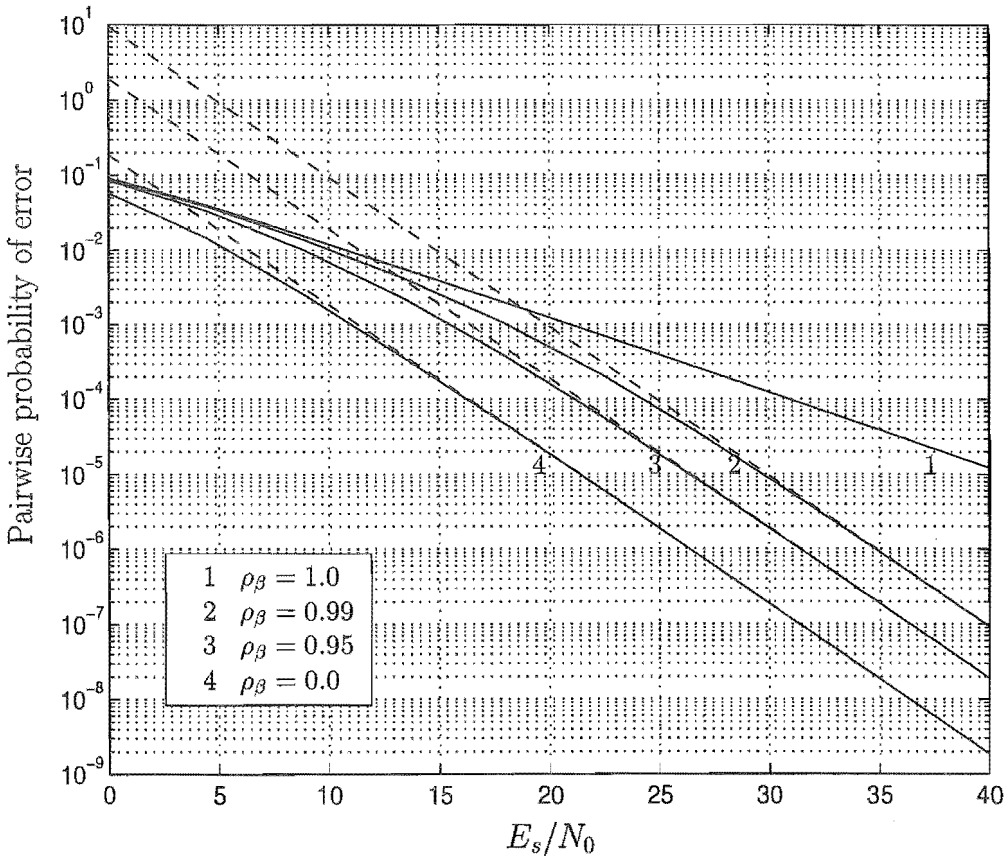


Figure 2.6: The illustration of a tight upperbound on the pairwise probability of error over a correlated Rayleigh fading channel assuming ideal CSI. The sequences have Hamming distance of two.

For example, in section 2.4.1, figure 2.2, we plotted the performance of an ideal CSI system transmitting two possible symbol sequences as a function of SNR. The matrix Φ_ρ for this example is

$$\Phi_\rho = \begin{pmatrix} 1 & \rho_\beta \\ \rho_\beta & 1 \end{pmatrix} \tag{2.90}$$

and the determinant is

$$\det \Phi_\rho = 1 - \rho_\beta^2 \tag{2.91}$$

The asymptotic loss for $\rho_\beta = 0.99$ is

$$\begin{aligned} \Delta_\rho|_{\rho_\beta=0.99} &= -\frac{10}{2}\log_{10}(1 - 0.99^2) \\ &\approx 8.51 \text{ dB} \end{aligned} \tag{2.92}$$

and for $\rho_\beta = 0.95$ is

$$\begin{aligned}\Delta_\rho|_{\rho_\beta=0.95} &= -\frac{10}{2}\log_{10}(1-0.95^2) \\ &\approx 5.05\ \text{dB} \end{aligned} \tag{2.93}$$

which agree with the losses observed in figure 2.2. In figure 2.7 we have plotted a family of curves describing the loss in dB as a function of $\beta f_D T$ for $l = 2$, $l = 3$ and $l = 4$. Note that the loss experienced by longer sequences is greater than shorter sequences. Again these curves emphasise the need for independently faded

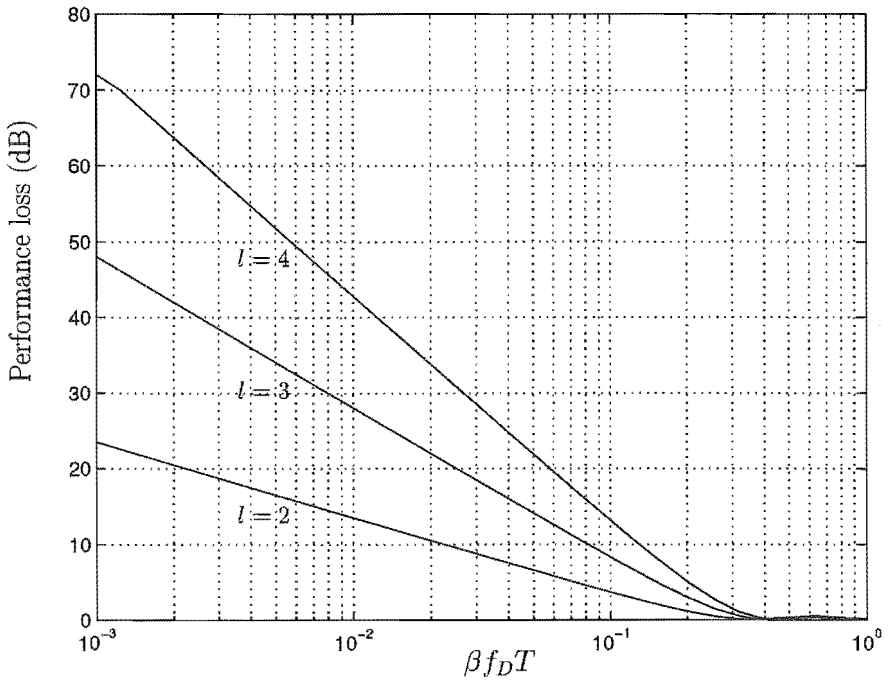


Figure 2.7: Performance loss, measured in dB of SNR, of a correlated system compared to an uncorrelated system as a function of $\beta f_D T$. Ideal CSI assumed.

symbols. The next section explains the technique of interleaving and deinterleaving to effectively achieve this objective.

2.6 Interleaving

The analysis of the pairwise probability of error of an ideal CSI system shows that the fading affecting the symbols in the message sequence must be independent to realise the full potential of coding. The correlation of the fading affecting two symbols in

a sequence is given by

$$\frac{1}{2}E[u_{jk}u_{jk+\beta}] = J_0(2\pi\beta f_D T) \tag{2.94}$$

where β is the distance, in symbols, between the two symbols. As shown by the example in section 2.4.1, values of $\beta f_D T > 3 \times 10^{-1}$ are sufficiently high for the fading affecting the symbols to be regarded as uncorrelated. Therefore we require

$$\beta f_D T > 3 \times 10^{-1} \tag{2.95}$$

or equivalently

$$\beta > \frac{3 \times 10^{-1}}{f_D T} \tag{2.96}$$

For example, for a maximum $f_D T$ of 0.01, β must be greater than 30. This implies that for the fading affecting the symbols in the transmitted sequence to be uncorrelated, there needs to be a spacing in time between the symbols of at least 30 symbols. We describe the technique of *block interleaving* to achieve this objective. A block interleaver consists of a buffer with α rows and β columns as shown in figure 2.8. The symbols c_{ik} fill the interleaver row by row, while the output sequence of

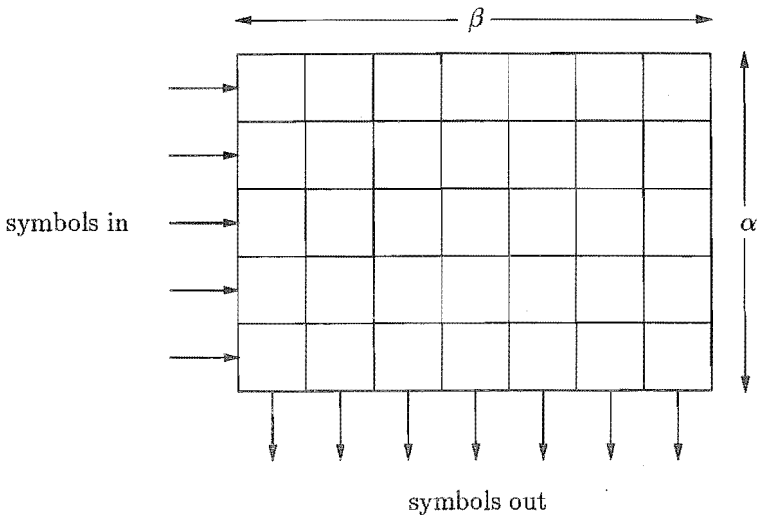


Figure 2.8: An $\alpha \times \beta$ block interleaver.

the interleaver is obtained by reading out the buffer's content column by column. Therefore a sequence of length α has all of it's symbols spaced by β symbols at the output. At the receiver there is a corresponding deinterleaver which inverts the

interleaving process. A deinterleaver is implemented in the same manner as an interleaver but with α and β reversed. Figure 2.9 illustrates the mapping between the input sequence and output sequence of a block interleaver for $\alpha = 3$ and $\beta = 4$.

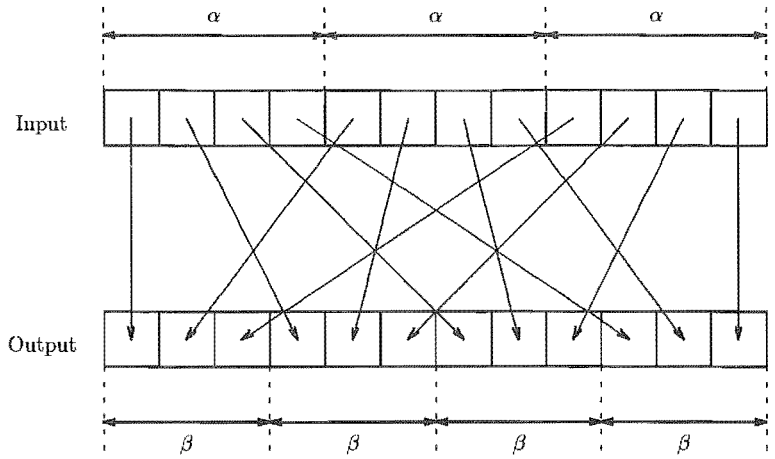


Figure 2.9: Mapping between input and output of a block interleaver for $\alpha = 4$, and $\beta = 3$.

2.6.1 Interleaver delay

The necessity for interleaving and deinterleaving to achieve optimal coding gains introduces a delay in the overall transmission time of the system. To fill and empty the interleaver and deinterleaving buffers takes $2\alpha\beta$ symbols. Therefore the minimum transmission delay measured in symbols associated with an interleaver deinterleaver pair to achieve uncorrelated fading for coding is

$$\delta = \frac{6 \times 10^{-1} l}{f_D T} \text{ symbols} \tag{2.97}$$

For example, for $l = 4$ and $f_D T = 0.01$, the minimum delay is 240 symbols.

2.7 Pilot tone aided detection.

The assumption in the previous sections of ideal CSI is useful for establishing lower bounds on the performance of the communications system, but is not necessarily indicative of how a real system will perform. As discussed in section 2.1, we need to know the state of the fading process in order to recover the transmitted data. One method used in practice to obtain an estimate of the channel's fading process

is to transmit a known pilot tone along with the data bearing signal. Such a pilot tone is inserted into a spectral null of the power spectrum of the data bearing signal and is extracted using a narrowband filter with a frequency response wide enough to allow the fading process to pass through undistorted. The receiver therefore has an estimate at each point in time of the fading process, assuming the fading is flat across the frequency band of the transmitted signal. The tradeoff of this technique is that some of the total power is taken up by the pilot tone. With a pilot tone, the sampled baseband representation of the system is

$$r'_k = A(c_k + p_t)u_k + n_k \tag{2.98}$$

where c_k is the data symbol, p_t is the pilot tone and A is a constant to normalise to unit transmission energy. If we define γ to be the ratio of the pilot tone energy to the total energy E_s transmitted per symbol then

$$p_t = \sqrt{\gamma} \tag{2.99}$$

and

$$A = \frac{1}{\sqrt{\gamma + 1}} \tag{2.100}$$

The received signal is filtered into two components, namely the data bearing component

$$r_k = \frac{c_k}{\sqrt{\gamma + 1}} u_k + n_k \tag{2.101}$$

and the pilot tone component which provides the channel estimate

$$v_k = \sqrt{\frac{\gamma}{\gamma + 1}} u_k + \eta_k. \tag{2.102}$$

B_p is the bandwidth of the frequency response function $H(f)$ of the pilot tone zonal filter

$$H(f) = \begin{cases} 1 & -B_p \leq f \leq B_p \\ 0 & \text{elsewhere} \end{cases} \tag{2.103}$$

To allow all of the fading process to pass through the filter we have

$$B_p \geq f_{D_{\max}} \tag{2.104}$$

where $f_{D_{\max}}$ is the maximum Doppler frequency experienced by the mobile. The variance of the noise component of the channel state estimate is related to N_0 by

$$\frac{1}{2}E[\eta_k] = B_p T N_0 \tag{2.105}$$

With a pilot tone the system covariance matrices differ. The autocorrelation of the received samples r_k now reflects the reduced energy

$$\Phi_{RR} = \frac{1}{\gamma+1} C_j \Phi_{UU} C_j^{\dagger} + \Phi_{NN} \tag{2.106}$$

The covariance matrix of the fading process u_k and the pilot tone aided channel estimate v_k is

$$\Phi_{UV}|\mathbf{c}_j = \sqrt{\frac{\gamma}{\gamma+1}}\Phi_{UU}. \tag{2.107}$$

and the covariance between r_k and v_k is

$$\Phi_{RV}|\mathbf{c}_j = \frac{\sqrt{\gamma}}{\gamma+1}\Phi_{UU}. \tag{2.108}$$

The autocorrelation matrix Φ_{VV} equals

$$\Phi_{VV}|\mathbf{c}_j = \frac{\gamma}{\gamma+1}\Phi_{UU} + \Phi_{\eta\eta} \tag{2.109}$$

where $\Phi_{\eta\eta}$ is the covariance matrix of the additive estimation noise of equation (2.102). Note that the η_k's represent bandpass filtered white Gaussian noise sampled at a rate of $1/T$ and are not independent. The ith row and jth column of the covariance matrix equals

$$\Phi_{\eta\eta}(i,j) = B_p T N_0 \text{sinc}((i-j)B_p T) \tag{2.110}$$

where $\text{sinc}(x) = \frac{\sin(\pi x)}{\pi x}$. The covariance matrix $\Phi_{NV}|\mathbf{c}_j$, between the channel estimate v_k and the additive white Gaussian noise n_k is zero because η_k and n_k are samples of output noise processes of two filters whose frequency responses do not overlap.

The generalised decoding metric does not simplify for pilot aided detection as it does for ideal channel information and the combination of equations (2.32), (2.106), (2.107) and (2.109) may be used to compute the pairwise probability of error of a pilot tone assisted system.

Ideal interleaving with pilot tone aided detection.

We have shown that interleaving is required to take full advantage of the gains offered by coding. We now analyse the case of ideally interleaved pilot tone aided decoding. With ideal interleaving the fading affecting the symbols in the message sequence is uncorrelated and the channel correlation matrix is simply

$$\Phi_{UU} = E_s \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix}. \tag{2.111}$$

The process of interleaving and deinterleaving also decorrelates the additive white noise affecting the channel state estimate sequence and

$$\Phi_{\eta\eta} = B_p T N_0 I \tag{2.112}$$

With these parameters, the matrices $\Phi_{RR}|\mathbf{c}_j$, $\Phi_{RV}|\mathbf{c}_j$ and $\Phi_{VV}|\mathbf{c}_j$ reduce to diagonal forms

$$\Phi_{RR}|\mathbf{c}_j = \begin{pmatrix} \frac{E_s|c_{j1}|^2}{\gamma+1} + N_0 & 0 & \cdots & 0 \\ 0 & \frac{E_s|c_{j2}|^2}{\gamma+1} + N_0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{E_s|c_{jL}|^2}{\gamma+1} + N_0 \end{pmatrix}, \tag{2.113}$$

$$\Phi_{RV}|\mathbf{c}_j = \begin{pmatrix} \frac{\sqrt{\gamma}E_s c_{j1}}{\gamma+1} & 0 & \cdots & 0 \\ 0 & \frac{\sqrt{\gamma}E_s c_{j2}}{\gamma+1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{\sqrt{\gamma}E_s c_{jL}}{\gamma+1} \end{pmatrix}, \tag{2.114}$$

and

$$\Phi_{VV}|\mathbf{c}_j = \begin{pmatrix} \frac{\gamma E_s}{\gamma+1} + f_D T N_0 & 0 & \cdots & 0 \\ 0 & \frac{\gamma E_s}{\gamma+1} + f_D T N_0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{\gamma E_s}{\gamma+1} + f_D T N_0 \end{pmatrix}, \tag{2.115}$$

and combine to form $\Phi_{WW}|\mathbf{c}_j$. The inverse of $\Phi_{WW}|\mathbf{c}_j$ is required for the generalised decoding metric of equation (2.23) and is found by applying the matrix inverse

identity of appendix A:

$$\Phi_{WW}^{-1}|\mathbf{c}_j = \begin{pmatrix} a_1 & & 0 & -b_1 & & 0 \\ & \ddots & & & \ddots & \\ 0 & & a_L & 0 & & -b_L \\ -b_1^* & & 0 & c_1 & & 0 \\ & \ddots & & & \ddots & \\ 0 & & -b_L^* & 0 & & c_L \end{pmatrix} \tag{2.116}$$

where the elements a_k, b_k and c_k equal

$$a_k = \frac{\gamma E_s/N_0 + f_D T + \gamma f_D T}{f_k} \tag{2.117}$$

$$b_k = \frac{\sqrt{\gamma} c_{jk} E_s/N_0}{f_k} \tag{2.118}$$

$$c_k = \frac{E_s/N_0 |c_{jk}|^2 + \gamma + 1}{f_k} \tag{2.119}$$

and

$$f_k = \frac{E_s}{N_0}(\gamma + |c_{jk}|^2 f_D T) + f_D T(\gamma + 1) \tag{2.120}$$

The determinant of $\Phi_{WW}|\mathbf{c}_j$ is

$$\det \Phi_{WW}|\mathbf{c}_j = \prod_{k=1}^{L} \frac{f_k}{N_0^2(\gamma+1)} \tag{2.121}$$

Substituting equations (2.116) and (2.121) into the generalised decoding metric of equation (2.23) gives

$$\begin{aligned} M_d(\mathbf{c}_j) &= \sum_{k=1}^{L} a_k |r_k|^2 - b_k r_k v_k^* - b^* r_k^* v_k + c_k |v_k|^2 + 2 \ln \frac{f_k}{N_0^2(\gamma+1)} \\ &= \sum_{k=1}^{L} a_k |r_k - \frac{b_k}{a_k} v_k|^2 + (c_k - \frac{|b_k|^2}{a_k}) |v_k|^2 + 2 \ln \frac{f_k}{N_0^2(\gamma+1)} \\ &= \sum_{k=1}^{L} a_k |r_k - \frac{b_k}{a_k} v_k|^2 + (\frac{f_k(\gamma+1)}{a_k}) |v_k|^2 + 2 \ln f_k - 2 \ln N_0^2(\gamma+1) \end{aligned} \tag{2.122}$$

The terms $\frac{f_k(\gamma+1)}{a_k}$ and $2 \ln N_0^2(\gamma+1)$ are independent of c_{jk} and may be discarded. The maximum likelihood decoding metric for ideally interleaved pilot tone aided detection is

$$M_d(\mathbf{c}_j) = \sum_{k=1}^{L} a_k |r_k - \frac{b_k}{a_k} v_k|^2 + 2 \ln f_k \tag{2.123}$$

which is suitable for Viterbi decoding. If the signalling format is PSK then $|c_{jk}|^2 = 1$ and the metric of equation (2.123) reduces to

$$M_d(\mathbf{c}_j) = \sum_{k=1}^{L} |r_k - \frac{b_k}{a_k} v_k|^2 \tag{2.124}$$

The probability of error for ideally interleaved pilot tone aided decoding may be derived using equation (2.32). The characteristic polynomial $p(x)$ of the matrix $\Phi_{WW}F$ is

$$p(x) = \prod_{k\in\eta} (a'_k x^2 + b'_k x + c'_k) \tag{2.125}$$

where the coefficients a'_k, b'_k and c'_k are

$$a'_k = (\gamma+1)(\frac{E_s}{N_0}(\gamma + |c_{jk}|^2 f_D T) + f_D T(\gamma+1)) \tag{2.126}$$

$$b'_k = -\frac{E_s}{N_0}(f_D T(|c_{ik}|^2 - |c_{jk}|^2)(\gamma+1) + \frac{E_s}{N_0}\gamma d_k^2) \tag{2.127}$$

$$c'_k = -\gamma(\frac{E_s}{N_0})^2 d_k^2 \tag{2.128}$$

where $d_k^2 = |c_{ik} - c_{jk}|^2$. From the eigenvalues of equation (2.125) the poles of $\Phi_D(s)$ are

$$p_k, p_{k+L} = \frac{a_k}{b_k \mp \sqrt{b_k^2 - 4a_k c_k}} \tag{2.129}$$

The pole p_k lies in the left hand plane and p_{k+L} lies in the right hand plane. The decision threshold parameter δ is in general non-zero and given by

$$\delta = 2\ln \prod_{k\in\eta} \frac{\gamma\frac{E_s}{N_0} + f_D T(\gamma + \frac{E_s}{N_0}|c_{ik}|^2 + 1)}{\gamma\frac{E_s}{N_0} + f_D T(\gamma + \frac{E_s}{N_0}|c_{jk}|^2 + 1)} \tag{2.130}$$

and the pairwise probability of error may be computed using these equations.

Ideally interleaved PSK with pilot tone aided detection.

The symbols of a PSK constellation have unit energy and the expressions for the pairwise probability of error for a pilot tone aided, ideally interleaved system simplify. With $|c_{ik}|^2 = |c_{jk}|^2 = 1$, equations (2.127), (2.128) and (2.128) defining the characteristic polynomial simplify to

$$a'_k = (\gamma+1)(\frac{E_s}{N_0}(\gamma + f_D T) + f_D T(\gamma+1)) \tag{2.131}$$

$$b'_k = -\gamma(\frac{E_s}{N_0})^2 d_k^2 \tag{2.132}$$

$$c'_k = -\gamma(\frac{E_s}{N_0})^2 d_k^2 \tag{2.133}$$

and the parameter δ equals 0. With these simplifications, the exact pairwise probability of error may be written as

$$p(\mathbf{c}_i \to \mathbf{c}_j) = -S_r \prod_{k\in\eta} p_k p_{k+L} \tag{2.134}$$

where

$$S_r = \sum_{i\in\eta} \lim_{s\to p_i} \frac{s-p_i}{s} \prod_{k\in\eta} \frac{1}{(s-p_k)(s-p_{k+L})} \tag{2.135}$$

the sum of the residues of the poles in the right hand plane. The product of the pair of poles p_k and p_{k+L} is

$$p_k p_{k+L} = -\frac{-(\gamma+1)(\frac{E_s}{N_0}(\gamma+f_D T)+f_D T(\gamma+1))}{4\gamma(E_s/N_0)^2 d_k^2} \tag{2.136}$$

and the exact pairwise probability of error is

$$p(\mathbf{c}_i \to \mathbf{c}_j) = S_r \prod_{k\in\eta} \frac{-(\gamma+1)(\frac{E_s}{N_0}(\gamma+f_D T)+f_D T(\gamma+1))}{4\gamma(E_s/N_0)^2 d_k^2} \tag{2.137}$$

For high SNR the poles p_k and p_{k+L} tend toward

$$p_k \to \frac{-(\gamma+1)(\gamma+f_D T)}{2\gamma d_k^2 \frac{E_s}{N_0}} \tag{2.138}$$

$$p_{k+L} \to \frac{1}{2}, \tag{2.139}$$

Using the bound on S_r from equation (2.72) we have the following bound on the pairwise error probability of a pilot tone aided PSK system with ideal interleaving

$$p(\mathbf{c}_i \to \mathbf{c}_j) \le \frac{(2l-1)!}{l!(l-1)!} \prod_{k\in\eta} \frac{1}{\frac{\gamma}{(\gamma+1)(\gamma+f_D T)} d_k^2 \frac{E_s}{N_0}} \tag{2.140}$$

The optimal pilot tone ratio.

The upperbound of equation (2.140) on the pairwise probability of error for pilot tone aided decoding is a function of the parameter γ, the ratio of the pilot tone power to the data signal power. We may optimise this parameter to minimise the probability of error of the system. From the upperbound expression for the pairwise probability of error, we need to maximise

$$\phi(\gamma) = \frac{\gamma}{(\gamma+1)(\gamma+f_D T)} \tag{2.141}$$

with respect to γ. Differentiating and setting the result to zero gives

$$\frac{d\phi(\gamma)}{d\gamma} = \gamma_{\text{opt}}^2 - f_D T = 0 \tag{2.142}$$

which gives the optimal value of γ as

$$\gamma_{\text{opt}} = \sqrt{f_D T} \tag{2.143}$$

i.e. at high SNR, with PSK signalling, the optimal pilot tone ratio is only a function of the fade rate $f_D T$. With the optimal value of γ the pairwise probability of error of ideally interleaved PSK with pilot tone aided detection is

$$p(\mathbf{c}_i \to \mathbf{c}_j) \leq \frac{(2l-1)!}{l!(l-1)!} \prod_{k \in \eta} \frac{(\sqrt{f_D T}+1)^2}{d_k^2 \frac{E_s}{N_0}} \tag{2.144}$$

The loss compared to ideally interleaved ideal channel state information is

$$= 20 \log_{10}(\sqrt{f_D T}+1) \tag{2.145}$$

Figure 2.10 shows the loss of pilot tone aided detection as a function of $f_D T$ for the optimal power ratio γ. Even at relatively fast fading ($f_D T = 0.1$) the loss

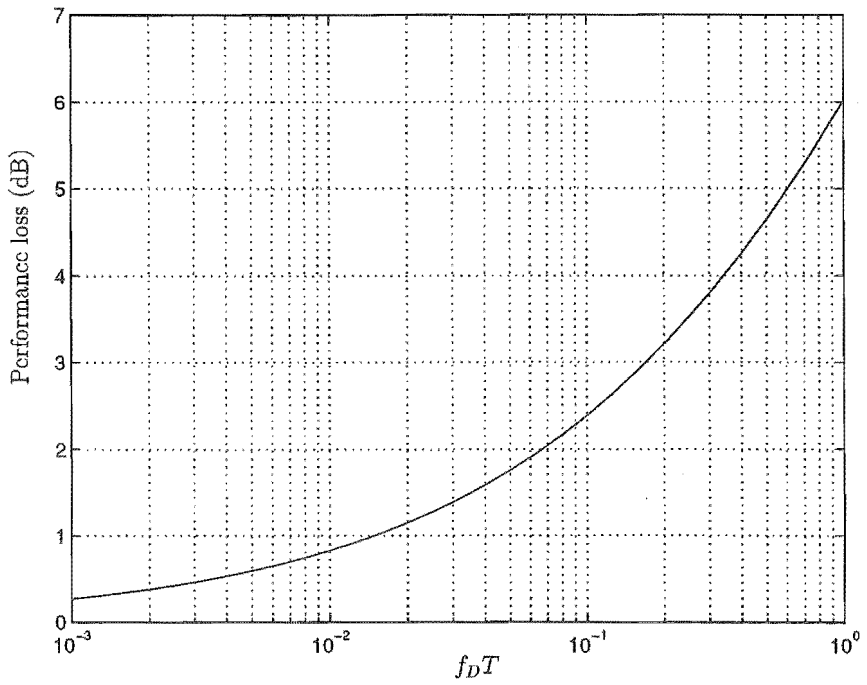


Figure 2.10: Loss in performance over ideal CSI with pilot tone aided detection as a function of $f_D T$.

in performance with the aid of a pilot tone is only about 2.4 dB relative to ideal channel state information and therefore it is a good method of obtaining a channel state estimate.

2.8 Summary.

We have presented a new generalised derivation of a maximum likelihood decoder for signal sequences transmitted on the Rayleigh fading channel. The analysis takes into account a non-ideal channel state estimate, consisting of the ideal channel state and a Gaussian distributed error term, and the correlation of the fading process. The final decoder metric is a maximum likelihood decoding metric of a general form. We derived an exact analytic expression for the pairwise probability of error for this decoding metric and discovered a new tight upperbound on the probability of error for the case of ideal CSI with non-interleaved transmission. The analysis of the probability of error shows that the correlation of the fading process is detrimental to the performance of a coded system and that interleaving is essential to optimise code performance fully. With interleaving the pairwise probability of error is inversely proportional to the signal-to-noise ratio to the power of the Hamming distance and inverse proportional to the squared product distance as shown by Divsalar and Simon [22]. This means that the optimal codes are those with the greatest Hamming distance and squared product distance in contrast to maximum Euclidean distance as is the case for the Gaussian channel. The need for interleaving means there is a fixed transmission delay proportional to the rate of fading and the diversity of the code.

Chapter 3

Introduction to Coding.

This chapter introduces the technique of coding in the context of chapters 4, 6 and 7 on geometrically uniform codes, multilevel codes and codes designed for multiple symbol differential detection respectively. We review the definition of signal space codes, the maximum likelihood decoding thereof, the performance evaluation methods and the consequent design criteria. In section 3.5 we introduce trellis codes as a special case of signal space coding and the discussion includes the performance evaluation, the decoding by the Viterbi algorithm and a bound on the maximum attainable Hamming distance (the desired maximised code parameter for the Rayleigh fading channel) of a trellis encoder over a multi-dimensional signal constellation.

3.1 Introduction.

In chapter 2 we analysed in detail the pairwise performance of maximum likelihood (ML) decoded symbol sequences transmitted across the Rayleigh fading channel. It has been established that the asymptotic behaviour of the probability of confusing sequence $\mathbf{c}_i$ for the sequence $\mathbf{c}_j$ (the pairwise probability of error) is bounded by

$$p(\mathbf{c}_i \to \mathbf{c}_j) \leq K \prod_{k \in \eta} \frac{1}{\frac{E_s}{N_0}|c_{ik} - c_{jk}|^2} \tag{3.1}$$

where the constant K is a function of the channel correlation and the method of obtaining channel state information. The set η contains the indices for which $c_{ik} \neq c_{jk}$ and has order l, the Hamming distance between the sequences $\mathbf{c}_i$ and $\mathbf{c}_j$. The probability of error decreases inversely with the signal-to-noise ratio (SNR) E_s/N_0,

raised to the power of l and inversely with the product distance

$$d_p^2 = \prod_{k \in \eta} |c_{ik} - c_{jk}|^2. \tag{3.2}$$

For sufficiently high SNR, an increase in l reduces the probability of error faster than an increase in d_p^2. To minimise the probability of error, we want to maximise the minimum Hamming distance l of the sequences first, and the squared product distance d_p^2 second.

A first order approach to increase the Hamming distance is simply to lengthen the sequences by symbol repetition. For example for uncoded 4-PSK, instead of transmitting one of the symbols 0, 1, 2, or 3, corresponding to the binary inputs 00, 01, 11 or 10, we transmit one of the repetition sequences (0,0), (1,1), (2,2), or (3,3). The Hamming distance between these sequences is two and the error probability is reduced by a factor of the SNR. Clearly the drawback of this method is that the data rate of the system is halved, and is reduced even further for symbol repetitions of a higher order. To restore or maintain the original data rate, the signalling rate would have to be increased, leading to an increase in the bandwidth of the system and this may not be desirable. An alternative which does not require this tradeoff is the technique of coding. Coding aims to reduce the probability of error by constructing sets of symbol sequences with increased distance (relative to uncoded signalling) without sacrificing the data rate, or increasing the bandwidth of the system. The cost is an increase in complexity.

3.2 Hard and Soft decision Coding Systems.

The literature on coding and decoding techniques is vast and only be touched upon in any one body of work. We will distinguish two types of coding on the manner by which they are decoded, namely *hard decision* decoding and *soft decision* decoding. In a hard decoding system, the processes of demodulation and decoding are performed independently as illustrated in figure 3.1. A vector $\mathbf{a}$ of k bits to be transmitted is input to a binary encoder C. The encoder C is such that it outputs a unique codeword $\mathbf{c}$ of length n bits corresponding to each $\mathbf{a}$. The set of all codewords $\mathbf{c}$ is denoted $\mathcal{C}$, and the order of $\mathcal{C}$ is 2^k, the number of possible input vectors. The code $\mathcal{C}$ has a Hamming distance d and as such can correct up to $t = d/2 - 1$ random

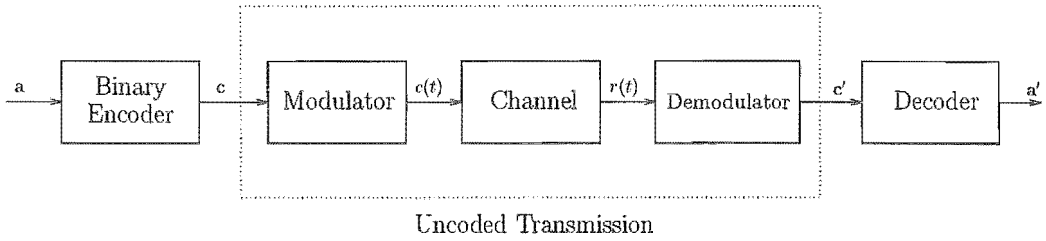


Figure 3.1: The block diagram of a hard-decision decoding system.

bit errors [30]. Such a code is an (n,k,d) block or convolutional code. The stream of codewords $\mathbf{c}$ is mapped to a signal constellation S, such as 4-PSK, and transmitted across the channel. The receiver demodulates the signal, corrupted by the channel, and makes a decision on the received symbols on a symbol by symbol basis. These symbols are mapped back to binary and assembled into output codewords $\mathbf{c}'$. The binary codewords have effectively passed through an uncoded system and contain random bit errors introduced by the channel. The decoder exploits the redundancy inherent in the code to correct up to t random bit errors caused by the channel and outputs the binary k-tuple most likely to have been transmitted. If more than t bit errors occurred then the decoder will make an incorrect decision and the output will not correctly reflect the transmitted data. The advantage of a hard decision technique is that the problems of coding and modulation are separated and the systems may be designed individually. The fact that block codes may be defined algebraically over finite fields means that many very efficient algebraic decoding algorithms and techniques may be used for decoding [7],[16],[47],[48]. Convolutional codes are another hard decision scheme, similar to block codes, except that the codewords are of infinite length. Much work on convolutional encoding and decoding techniques is in the literature [1],[14],[40],[38],[36],[37],[55].

The disadvantage of a hard-decision technique is that a loss of performance is incurred due to decision information being lost in the hard decision process in the demodulator.

In contrast to a hard decision system, a *soft decision* decoding system is one which *combines* the processes of demodulation and decoding as illustrated in figure 3.2. The binary input data $\mathbf{a}$ is mapped directly onto a set of channel symbols S, such as a PSK constellation, and the resultant code C is treated as a set of points in a real Euclidean space in contrast to a set of vectors over a binary field as is the

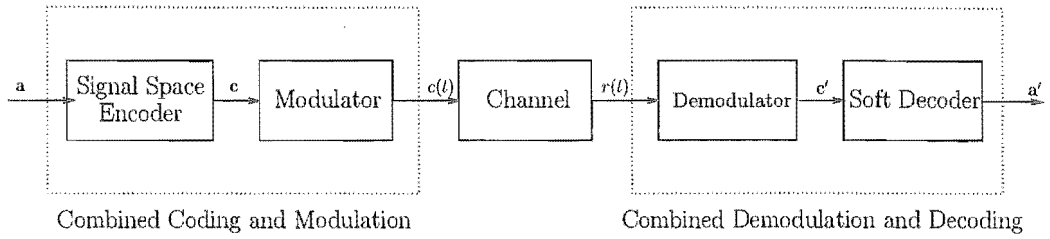


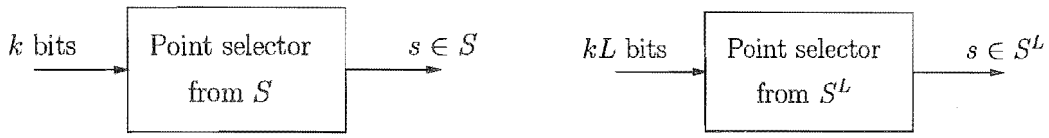
Figure 3.2: The block diagram of a soft-decision decoding system.

case in a hard decision system. Such a code $\mathcal{C}$ is called a *signal space code*. The stream of signal space codewords is modulated onto the carrier to give $c(t)$ and is transmitted across the channel. The receiver demodulates the received signal $r(t)$, corrupted by the channel, and outputs a stream of points in Euclidean space. This stream of points is decoded, not by making a symbol by symbol decision, but by making a decision on a *sequence* of symbols, as introduced in chapter 2. Such a decision is made by computing a metric (which is a function of the channel) between the received point and the set of all possible points that may have been transmitted. The codeword $\mathbf{c} \in \mathcal{C}$ which minimises the metric to the received point $\mathbf{r}$ is the output of the decoder. The soft-decision process does not discard any decision information as occurs in the demodulator of a hard decision system. By computing a metric, individual symbols in a codeword are weighted by a "reliability" or "confidence" factor which increases the decision accuracy. Our continuing discussion will focus on signal space codes only. Section 3.3 is a discussion on signal space constellations leading up to the definition of signal space codes in section 3.4. Sections 3.4.1, 3.4.2, 3.4.3 discuss decoding, error evaluation and design criteria respectively. Trellis codes are a special case of signal space codes and are the topic of section 3.5.

3.3 Signal space constellations.

In a conventional uncoded digital communications system we transmit points (symbols) from a two-dimensional constellation S across the channel on a symbol by symbol basis. Two dimensions are transmitted per symbol because the carrier frequency has two degrees of freedom, namely phase and amplitude, or alternatively in-phase and quadrature components corresponding to a polar and rectangular representation respectively. The two-dimensional points to be transmitted are drawn from a two-

dimensional constellation such as M-PSK, or QAM. We may relabel such a system by taking the L-fold product of a two-dimensional constellations S and calling it a multi-dimensional constellation S^L, an example of which is $L \times M$-PSK. Clearly S^L is a set of $|S|^L$ points in $2L$ dimensional space. The communication system may alternatively be represented as one which transmits a sequence of $2L$-dimensional points in contrast to transmitting a sequence of two-dimensional points. The contrast is shown in figure 3.3. The advantage of considering constellations of higher



(a) Selector from S. (b) Equivalent selector from S^L.

Figure 3.3: A system transmitting points from a two-dimensional constellation S is represented equivalently by a system transmitting points from a multi-dimensional constellation S^L.

dimensionality is that it is possible to design constellations in higher dimensions with better distance properties than those in lower dimensions. An informative example [29] illustrating the concept of higher dimensional constructions is based on the S =16-QAM constellation shown in figure 3.4. With the coordinate system labelled

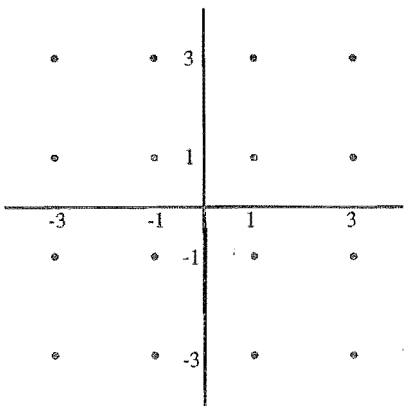


Figure 3.4: 16-QAM constellation.

on the axes, the minimum Euclidean distance between points is 2. The two-fold product S^2 of S is a four dimensional constellation consisting of 256 points with the

same distance properties as the underlying 16-QAM constellation. However, unlike two-dimensional 16-QAM, it is possible to add points to this four-dimensional constellation without changing the minimum distance. For example if a point is added to the origin, i.e. at $(0,0,0,0)$ then the distance to one of its nearest neighbours, say $(1,1,1,1)$ is $\sqrt{1^2+1^2+1^2+1^2} = 2$ which is the same the minimum distance of 16-QAM. Similarly points may be added at all positions with even coordinates in the range of $-2 \ldots 2$ to give a 625 point constellation. By selecting the 256 point subset of this constellation shown in table 3.1 we have a four dimensional constellation S' with the same number of points as the square of 16-QAM, but with better distance properties after the normalisation of the constellations to unit energy. The

	Generator	Number of permutations
1	(1,1,1,1)	16
2	(2,0,0,0)	8
3	(2,2,0,0)	24
4	(2,2,2,0)	32
5	(2,2,2,2)	16
6	(3,1,1,1)	64
7	(3,3,1,1)	96
	total	256

Table 3.1: 256 signal points generated by taking all sign and coordinate permutation of each of the seven generators.

constellation S' is a subset of the densest known lattice in four dimensions, namely the Schäfli or D_4 lattice [17]. Notice that S' cannot be written as a product of two lower dimensional constellations. This is true in general for a higher dimensional constellation, however they can always be represented as a *subset* of an L-fold product of a lower dimensional constellation. To show this consider a constellation S in $L \times N$ dimensions. We wish to write S as a subset of an L-fold product of an N dimensional constellation S_0. To do so, we may write S as a set of L-tuples over N

dimensions.

$$\begin{aligned} S \;=\; & \{(s_{11}, s_{12}, \ldots, s_{1L}), \\ & (s_{21}, s_{22}, \ldots, s_{2L}), \\ & \qquad \vdots \\ & (s_{n1}, s_{n2}, \ldots, s_{nL}), \quad s_{ij} \in R^N\} \end{aligned} \tag{3.3}$$

where $n = |S|$, the number of points in S. A projection $P_j(S)$ of S is defined as the set

$$P_j(S) = \{s_{ij}, i = 1 \ldots n\} \tag{3.4}$$

The underlying N-dimensional constellation S_0, such that $S \subset S_0^L$ is

$$S_0 = \bigcup_{j=1}^{L} P_j(S), \tag{3.5}$$

the union of all projections of S. For example, the projections $P_1(S')$, and $P_2(S')$ of S', where S' is defined in the previous example, are shown in figure 3.5. We have $P_1(S') = P_2(S')$, $S'_0 = P_1(S') \cup P_2(S')$ and $S' \subset S'_0 \times S'_0$, hence S' has been written as a subset of a power of a lower dimensional constellation S'_0.

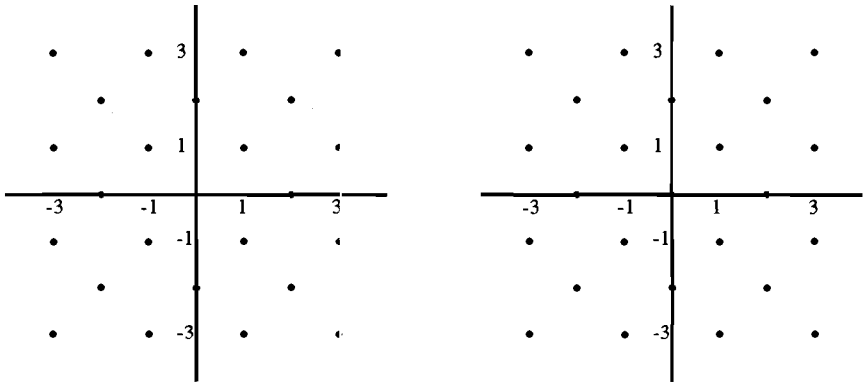


Figure 3.5: Projections $P_1(S')$ and $P_2(S')$ of S' respectively.

3.4 Signal space Codes.

In the previous section we have shown that all multi-dimensional signal space constellations may be written as a subset of a power of lower dimensional constellations.

From a simple example it was shown that gains can be made by carefully selecting such a subset. The technique of systematically generating good subsets of products of lower dimensional constellations is called signal space coding. Depending upon the encoding technique used to generate the signal space code it may be convenient to define the underlying constellation as multi-dimensional (see chapter 4, however such a constellation may again be reduced to a subset of a power of lower dimensional constellations. Traditionally the lowest number of dimensions considered is two, the degrees of freedom of the carrier. Definition 1 formally defines a signal space code.

Definition 1 *Consider a signal constellation S in the real L dimensional space R^L. The N-fold Cartesian product of S, written as S^N, is the set of all possible length N sequences over S:*

$$S^N = \{(s_1, s_2, \ldots, s_N), s_k \in S\} \tag{3.6}$$

S^N may be viewed as a set of $|S|^N$ points in NL dimensional space. Any subset $\mathcal{C}$ of S^N is a signal space code.

Definition 1 is general and does not make any assertions as to how to construct good signal space codes or how to decode them. The block diagram of a signal space encoder is drawn in figure 3.6. A vector $\mathbf{a}$ of k bits is input to the encoder. The

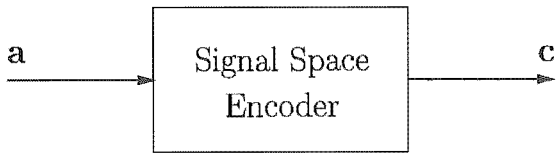


Figure 3.6: A signal-space encoder.

output is a unique sequence $\mathbf{c} \in \mathcal{C}$ of points in Euclidean space. The order of $\mathcal{C}$ is

$$|\mathcal{C}| = 2^k \tag{3.7}$$

We may write the one-to-one mapping between $\mathbf{a}$ and $\mathbf{c}$ as

$$\mathbf{c} = \mathcal{C}(\mathbf{a}) \tag{3.8}$$

And the inverse mapping as $\mathbf{a} = \mathcal{C}^{-1}(\mathbf{c})$. The rate r of the signal space code $\mathcal{C}$ in bits per two-dimensional symbol is

$$r = \frac{2k}{LN} \tag{3.9}$$

3.4.1 Decoding of Signal Space Codes.

A signal space code $\mathcal{C}$ is decoded using a decoding metric $m(\mathbf{r}, \mathbf{c})$. The metric $m(\mathbf{r}, \mathbf{c})$ measures the "distance" between the received vector $\mathbf{r}$ and the postulated codeword $\mathbf{c} \in \mathcal{C}$. The decoder selects the codeword $\mathbf{c} \in \mathcal{C}$ which minimises the metric $m(\mathbf{r}, \mathbf{c})$, i.e.

$$\mathbf{c} = \min_{\mathbf{c} \in \mathcal{C}} m(\mathbf{r}, \mathbf{c}) \tag{3.10}$$

Typically $m(\mathbf{r}, \mathbf{c})$ is derived for a particular channel such that the output of the decoder is the maximum likelihood estimate of the transmitted codeword. It has been shown by Divsalar and Simon [22] and in chapter 2 that the ML decoding metric for the Rayleigh fading channel, assuming ideal interleaving and ideal CSI, is the squared Euclidean distance metric, i.e. $m(\mathbf{r}, \mathbf{c}) = ||\mathbf{r} - \mathbf{c}||^2$. The Euclidean distance is also the maximum likelihood decoding metric for the additive white Gaussian noise (AWGN) channel [38]. Associated with each codeword in the code $\mathcal{C}$ is a region of space called the *Voronoi* region. The Voronoi region $V(\mathbf{c})$ of a codeword $\mathbf{c}$ is the region of space containing all points closer to $\mathbf{c}$ than to any other codeword in $\mathcal{C}$, i.e.

$$V(\mathbf{c}) = \{\mathbf{r}, \min_{\mathbf{c} \in \mathcal{C}} m(\mathbf{r}, \mathbf{c}) = m(\mathbf{r}, \mathbf{c})\} \tag{3.11}$$

The Voronoi region of a codeword $\mathbf{c}$ is also known as the *decision* region of $\mathbf{c}$, that is, if the received vector $\mathbf{r}$ is in $V(\mathbf{c})$ then the decoder decides on $\mathbf{c}$ as its output. Figure 3.7 shows the Voronoi region of a code point $\mathbf{c}$ in two-dimensional space, with $m(\mathbf{r}, \mathbf{c}) = ||\mathbf{r} - \mathbf{c}||^2$. For arbitrary positioning of the code points in space, the Voronoi regions are not easily computed and the complexity increases quickly with increased dimensionality of the problem. For an arbitrary signal space code the decoding algorithm is to compute the decoding metric from the received point $\mathbf{r}$ to every codeword $\mathbf{c} \in \mathcal{C}$ and to output that codeword which produces a minimum. Clearly such an exhaustive method is very computationally intensive and the complexity increases exponentially with the dimensionality of the code. Much more efficient

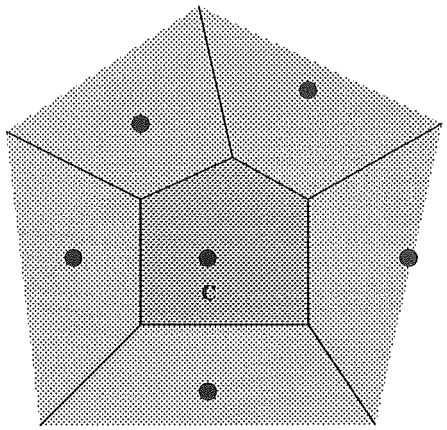


Figure 3.7: Voronoi region of a code point $\mathbf{c}$ in two dimensions. The decoding metric is the Euclidean distance.

methods are available for specific encoding techniques, such as trellis codes. These are discussed in section 3.5.

3.4.2 Error evaluation of Signal Space Codes.

If the codeword $\mathbf{c}_i$ was transmitted, and the decoder selects the codeword $\mathbf{c}_j \neq \mathbf{c}_i$ as its output then an *error event* is said to have occurred. Associated with an error event are a number of *bit errors* which are the number of bits different between the corresponding binary input words of $\mathbf{c}_i$ and $\mathbf{c}_j$. Clearly it is of interest to determine the probability of an error event occurring, and if possible the average probability of a bit error. To do this we consider the probability density function (pdf) $p(\mathbf{r}|\mathbf{c}_i)$ of the received vector $\mathbf{r}$ conditioned on the transmitted codeword $\mathbf{c}_i$. Assuming the *a priori* probability of transmission $p(\mathbf{c}_i)$ to be equiprobable, then the probability of making a *correct* decision, is the probability of the received vector $\mathbf{r}$ occurring in the Voronoi region $V(\mathbf{c}_i)$ of $\mathbf{c}_i$. This probability may be found by integrating the pdf of $\mathbf{r}$ conditioned on $\mathbf{c}_i$ over $V(\mathbf{c}_i)$ and is written as

$$p_{\text{correct}}(\mathbf{c}_i) = \int_{V(\mathbf{c}_i)} p(\mathbf{r}|\mathbf{c}_i) dV \tag{3.12}$$

The probability of making an error when $\mathbf{c}_i$ is transmitted is simply $p_{\text{err}}(\mathbf{c}_i) = 1 - p_{\text{correct}}(\mathbf{c}_i)$ and this probability may be averaged over all codewords to give the *average* probability of error $p_{\text{err}}(\mathcal{C})$ of the code

$$p_{\text{err}}(\mathcal{C}) = \frac{1}{|\mathcal{C}|} \sum_{\mathbf{c}_i \in \mathcal{C}} p_{\text{err}}(\mathbf{c}_i) \tag{3.13}$$

To compute the bit error rate (BER) of the system, we must take into account the number of bits different between the codeword actually transmitted and an error word. The expression for the BER is

$$\text{BER} = \frac{1}{|\mathcal{C}|} \sum_{\mathbf{c}_i \in \mathcal{C}} \sum_{\substack{\mathbf{c}_j \in \mathcal{C} \\ \mathbf{c}_j \neq \mathbf{c}_i}} w(\mathbf{c}_i, \mathbf{c}_j) \int_{V(\mathbf{c}_j)} p(\mathbf{r}|\mathbf{c}_j) dV \tag{3.14}$$

where $w(\mathbf{c}_i, \mathbf{c}_j)$ is the weight or the Hamming distance between the corresponding binary inputs of codewords $\mathbf{c}_i$ and $\mathbf{c}_j$. Although equations (3.13) and (3.14) are exact, they are, with the exception of trivial examples, very difficult to compute in practice. In general, for good codes, the Voronoi regions are multi-dimensional polytopes making the integration over the probability density function difficult. A useful upperbound on the probability of error $p_{\text{err}}(\mathbf{c}_i)$ of a codeword $\mathbf{c}_i$ is the *union bound* given by

$$p_{\text{err}}(\mathbf{c}_i) \leq \sum_{\substack{\mathbf{c}_j \in \mathcal{C} \\ \mathbf{c}_j \neq \mathbf{c}_i}} p(\mathbf{c}_i \rightarrow \mathbf{c}_j) \tag{3.15}$$

where $p(\mathbf{c}_i \rightarrow \mathbf{c}_j)$ is the pairwise probability of confusing the codeword $\mathbf{c}_i$ for $\mathbf{c}_j$. For the Rayleigh fading channel, $p(\mathbf{c}_i \rightarrow \mathbf{c}_j)$ has been evaluated exactly for different system configurations in chapter 2. Figure 3.8 shows the application of the union bound to calculate the probability of error of a two-dimensional codeword. The areas on the other side of the lines bounding $\mathbf{c}_i$ have been integrated. The darker areas show the overlap regions and generally the union bound is a loose bound, particularly at low signal to noise ratios. As the SNR increases, the bulk of the area of the conditional pdf of the received vector is within the Voronoi region and the union bound tightens. The union bound on the overall probability of error of the system is

$$p_{\text{err}}(\mathbf{c}_i) \leq \frac{1}{|\mathcal{C}|} \sum_{\mathbf{c}_i \in \mathcal{C}} \sum_{\substack{\mathbf{c}_j \in \mathcal{C} \\ \mathbf{c}_j \neq \mathbf{c}_i}} p(\mathbf{c}_i \rightarrow \mathbf{c}_j) \tag{3.16}$$

and similarly the union bound on the overall BER is

$$\text{BER} \leq \frac{1}{|\mathcal{C}|} \sum_{\mathbf{c}_i \in \mathcal{C}} \sum_{\substack{\mathbf{c}_j \in \mathcal{C} \\ \mathbf{c}_j \neq \mathbf{c}_i}} w(\mathbf{c}_i, \mathbf{c}_j) p(\mathbf{c}_i \rightarrow \mathbf{c}_j) \tag{3.17}$$

A special case on the probability of error of the system exists if the code has what is known as a *uniform error property* (UEP). This property means that the probability

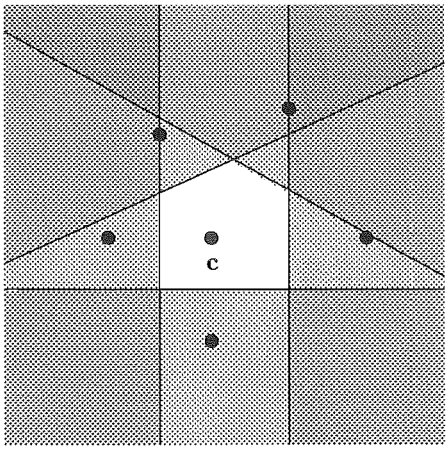


Figure 3.8: Example of application of the union bounding technique.

of error $p_{\text{err}}(\mathbf{c}_i)$, when $\mathbf{c}_i$ is transmitted, is *constant* independently of $\mathbf{c}_i$ and the evaluation of the probability of error of the code greatly simplified because only the error probability for *one* codeword needs to be considered. Typically the codeword considered is the *all zeroes* codeword written as $\mathbf{c}_0$, that is the codeword generated when a stream of all zeroes is transmitted. With this condition and assuming an equiprobable *a priori* transmission, the exact probability of error of a code $\mathcal{C}$ is given by

$$p_{\text{err}}(\mathcal{C}) = 1 - \int_{V(\mathbf{c}_0)} p(\mathbf{r}|\mathbf{c}_0)dV \tag{3.18}$$

and the union bound is

$$p_{\text{err}}(\mathcal{C}) \leq \sum_{\substack{\mathbf{c}_j \in \mathcal{C} \\ \mathbf{c}_j \neq \mathbf{c}_0}} p(\mathbf{c}_0 \rightarrow \mathbf{c}_j) \tag{3.19}$$

It is desirable to construct codes with a uniform error property as it greatly simplifies the code search effort. The geometrically uniform trellis codes constructed in chapter 4 by design have the UEP property.

3.4.3 Signal space code design criteria.

In the previous section we examined the probability of error of signal space codes. This probability of error calculation should be used as a guide to designing good signal space codes for the channel of interest by designing the code such that the probability of error is minimised for a specified level of system complexity. The

union bound of equation (3.19) on the probability of error of the system is in terms of the pairwise probability of error $p(\mathbf{c}_i \rightarrow \mathbf{c}_j)$. By substituting the appropriate pairwise probability of error expression for the channel we can determine the code properties necessary to minimise the probability of error. We briefly examine the design criteria for the classical additive white Gaussian noise (AWGN) channel and compare it to the channel of interest to us, the Rayleigh fading channel. We will see that the design criteria for these channels are quite different.

AWGN channel.

The pairwise probability of error for the AWGN channel is bounded by

$$p(\mathbf{c}_i \rightarrow \mathbf{c}_j) \leq Q\left(\sqrt{\frac{E_s}{N_0}} d(\mathbf{c}_i, \mathbf{c}_j)\right) \tag{3.20}$$

where $d(\mathbf{c}_i, \mathbf{c}_j)$ is the Euclidean distance between the codewords $\mathbf{c}_i$ and $\mathbf{c}_j$, and $Q(\cdot)$ is the Gaussian error-probability function defined by

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-\frac{y^2}{2}} dy \tag{3.21}$$

Substituting equation (3.20) into the union bound of equation 3.19 gives

$$p_{\text{err}}(\mathcal{C}) \leq \sum_{\substack{\mathbf{c}_j \in \mathcal{C} \\ \mathbf{c}_j \neq \mathbf{c}_0}} Q\left(\sqrt{\frac{E_s}{N_0}} d(\mathbf{c}_0, \mathbf{c}_j)\right) \tag{3.22}$$

which is dominated by the codewords $\mathbf{c}_j$ with the smallest Euclidean distance to the codeword $\mathbf{c}_0$. Clearly the code performance is optimised when the minimum Euclidean distance of the code d_{free} is maximised.

Rayleigh fading channel.

The channel of interest to us is the Rayleigh fading channel. The union bound of equation (3.19) is in terms of the pairwise probability of error which has been found exactly in chapter 2 and a tight upperbound at high SNR is given by [22]

$$p(\mathbf{c}_i \rightarrow \mathbf{c}_j) \leq \prod_{k \in \eta} \frac{4}{\frac{E_s}{N_0} |c_{ik} - c_{jk}|^2} \tag{3.23}$$

The product in equation (3.23) is taken over the set η of k for which $c_{ik} \neq c_{jk}$. The average probability, $P_{\text{err}}(\mathcal{C})$, of a ML decoded GU code $\mathcal{C}$, assuming the transmission

of each code word to be equally probable, is bounded by the union bound

$$p_{\text{err}}(\mathcal{C}) \leq \sum_{c_j \in \mathcal{C}} p(\mathbf{c}_0 \rightarrow \mathbf{c}_j) \tag{3.24}$$

The *minimum* Hamming distance l_0, between any two codewords, is referred to as the *diversity* of the code. By combining the contribution of the codewords with equal Hamming distance, equation (3.24) can be written to give

$$p_{err}(\mathcal{C}) \leq \sum_{l=l_0}^{\infty} \rho_l \left(\frac{4}{E_s/N_0} \right)^l \tag{3.25}$$

where

$$\rho_l = \sum_{c_j \in \mathcal{C}_l} \frac{1}{\prod_{k \in \eta} |c_{0k} - c_{jk}|^2}. \tag{3.26}$$

The set $\mathcal{C}_l$ is the set of codewords at a Hamming distance of l from the all zero codeword C_0. Effectively ρ_l is a weighted "multiplicity" [75] term of the dominant error event. We denote the term $(\rho_{l_0})^{-1}$ as the *weighted product distance* d^2_{pw} and it is defined by

$$d^2_{pw} = \left(\sum_{c_j \in \mathcal{C}_{l_0}} \frac{1}{\prod_{k \in \eta} |c_{0k} - c_{jk}|^2} \right)^{-1}. \tag{3.27}$$

At high SNR, the pairwise error performance of the code is dominated by the terms of equation (3.25) that have a Hamming distance equal to the diversity l_0. Therefore the code design criteria for the Rayleigh fading channel are

1. To find the set of codes with maximum diversity l_0.
2. To select from this set, the code that maximises the weighted product distance d^2_{pw}.

Note that the design criteria for the Gaussian channel is to maximise the minimum Euclidean distance while for the Rayleigh fading channel we want to maximise the minimum Hamming distance.

3.5 Trellis codes.

Trellis coding is a method for generating a signal space code with good distance properties for which an efficient decoding algorithm exists. Trellis codes were first introduced by Ungerboeck [75] for the AWGN channel and many papers have since been

published on the topic [4],[6],[11],[12],[19],[26],[28],[39],[49],[53],[54],[59],[73],[74],[83]. The general form of a trellis encoder is shown in figure 3.9. The encoder consists

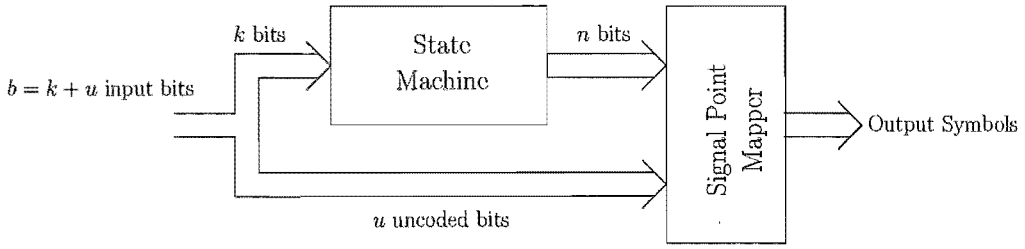


Figure 3.9: General form of a trellis encoder.

of a binary finite state machine with v memory elements and a mapper onto an L dimensional signal constellation S. At time step j, b bits to be transmitted enter the encoder. Of the b bits, k are passed to the state machine and the remaining $u = b - k$ bits, referred to as the *uncoded* bits are passed directly to the signal point mapper. The rate of the state machine is k/n to give a total of $u + n$ bits entering the signal point mapper. The signal mapper maps the $u + n$ bits one-to-one onto the constellation S. Clearly the order of S is $|S| = 2^{u+n}$. The rate of the overall signal space code in bits per two-dimensional channel symbol is

$$r = \frac{2b}{L}. \tag{3.28}$$

Figure 3.10 shows an example of a trellis encoder from the paper by Ungerboeck [75]. The encoder consists of a two delay element encoder, giving a total of $2^2 = 4$ states.

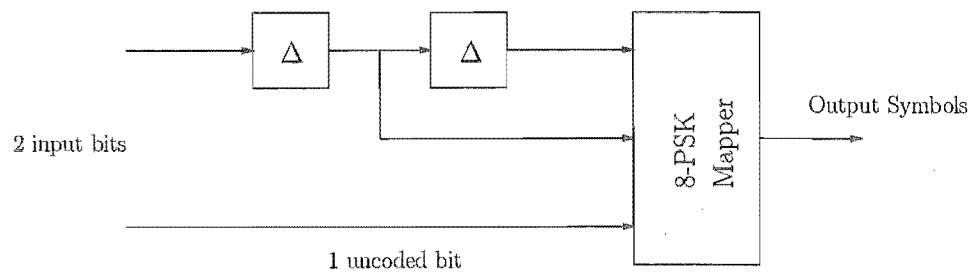


Figure 3.10: A 4-state rate 2/3 trellis encoder.

Of the two input bits, one is uncoded and one is encoded to give two output bits. The three bits are combined and mapped onto an 8-PSK constellation using the mapping shown in table 3.2. This mapping has been obtained by using the method

of *set partitioning* [75]. The structure of a trellis code may be conveniently drawn on

$a_0a_1a_2$	PSK symbol
000	0
001	4
010	2
011	6
100	1
101	5
110	3
111	7

Table 3.2: The mapping from the encoder output bits to an 8-PSK constellation.

a *trellis diagram.* A trellis diagram consists of the states of the state machine drawn as points vertically and repeated horizontally to represent transitions with time. An input to the encoder causes a state transition and an output symbol drawn from the constellation S . The state transition is represented by a line connecting the initial state at time j and the state at time $j+1$. Drawn on the line is the corresponding output symbol. Figure 3.11 shows the state transition diagram of the trellis encoder of figure 3.10.

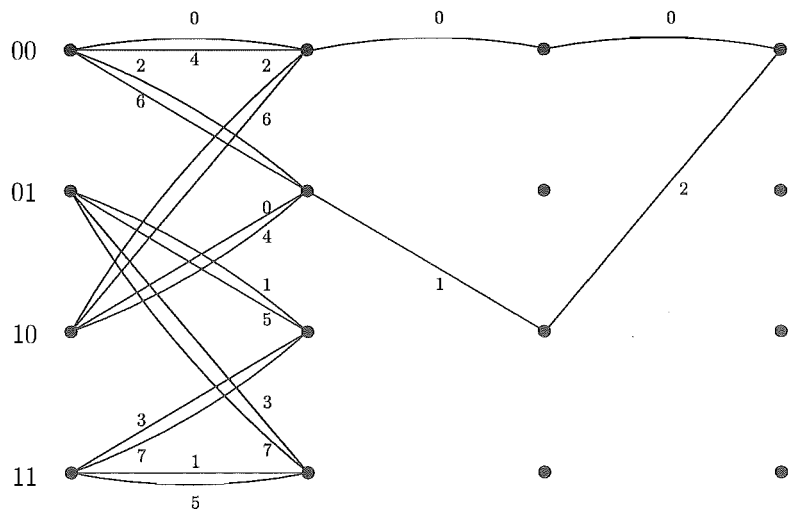


Figure 3.11: State diagram of the trellis encoder of figure 3.10.

3.5.1 Decoding of trellis codes.

Trellis codes are a special case of signal space codes and as such are decoded by computing a metric $m(\mathbf{r}, \mathbf{c})$ between the received sequence $\mathbf{r}$ and every sequence in the code $\mathcal{C}$. Consider a code sequence $\mathbf{c}$ and a received sequence $\mathbf{r}$. Let the notation $\mathbf{r}[i]$ denote the i^{th} element of the vector $\mathbf{r}$ and the notation $\mathbf{r}[i, j]$ the subsequence of $\mathbf{r}$ from elements i to j inclusive. If the metric $m(\mathbf{r}, \mathbf{c})$ can be written in the form

$$m(\mathbf{r}, \mathbf{c}) = \sum_i m(\mathbf{r}[i], \mathbf{c}[i]) \tag{3.29}$$

then a trellis code can be efficiently decoded using the *Viterbi* algorithm [38]. Examining equation (3.29) we see that the Euclidean distance metric $m(\mathbf{r}, \mathbf{c}) = ||\mathbf{r} - \mathbf{c}||^2$, used for decoding codes on the AWGN and Rayleigh fading channels may be written in this form. The Viterbi algorithm is described by the algorithm in table 3.3. Suboptimal techniques for the decoding of trellis codes are also exist but are not

1. Starting at time unit $j = 0$, compute the metric for the single path entering each state of the encoder. Store the survivor path and its metric for each state.

2. Increment the time j by one. Compute the metric for all paths entering each state by adding the metric of the incoming branches to the metric of the connecting survivor from the previous time unit. For each state identify the path with the lowest metric as the survivor of step 2. Store the survivor and its metric.

3. If all survivor paths have a common starting path to a state at time k then that is the maximum likelihood decision and is the decoder output. Repeat starting at step 2.

Table 3.3: The Viterbi algorithm for the maximum likelihood (ML) decoding of trellis codes.

discussed here [56].

3.5.2 Evaluation of Trellis codes.

If we assume the trellis code has a uniform error property, then the performance of a code is dominated by the shortest distance codeword to the all-zero codeword, where shortest distance for the Rayleigh fading channel is measured by the Hamming distance and product distance. To compute the shortest distance codeword we need to find a path through the trellis starting and ending in the zero state which has the minimum distance properties. In contrast to the Gaussian channel where the only distance measure of interest is the Euclidean distance, for the Rayleigh fading channel there are *two* distances of interest, namely the Hamming distance and the squared product distance and minimising the Hamming distance has priority over minimising the product distance. We therefore define a combined distance measure between sequences two sequences $\mathbf{c}_i$ and $\mathbf{c}_j$ consisting of the Hamming distance and product distance combined in a two element vector $\mathbf{d}$,

$$\mathbf{d}(\mathbf{c}_i, \mathbf{c}_j) = (d_H(\mathbf{c}_i, \mathbf{c}_j), d_p^2(\mathbf{c}_i, \mathbf{c}_j)) \tag{3.30}$$

where $d_H(\mathbf{c}_i, \mathbf{c}_j)$ and $d_p^2(\mathbf{c}_i, \mathbf{c}_j)$ are the Hamming distance and squared product distance between $\mathbf{c}_i$ and $\mathbf{c}_j$ respectively. We define the operation $\oplus$ of adding two distance metrics $\mathbf{d}_1 = (d_{H_1}, d_{p_1}^2)$ and $\mathbf{d}_2 = (d_{H_2}, d_{p_2}^2)$ by

$$\mathbf{d}_1 \oplus \mathbf{d}_2 = (d_{H_1} + d_{H_2}, d_{p_1}^2 \times d_{p_2}^2) \tag{3.31}$$

and the logical "less-than" comparison operator $<$ by

$$\mathbf{d}_1 < \mathbf{d}_2 \text{ if } d_{H_1} < d_{H_2} \text{ or } d_{H_1} = d_{H_2} \text{ and } d_{p_1}^2 < d_{p_2}^2 \tag{3.32}$$

The algorithm to compute the minimum distance measure is shown in table 3.4. The explanation of the algorithm to compute the minimum distance of a trellis code is as follows. Step 1 initialises the time variable t. Step 2 initialises the metric variables $\mathbf{m}(s,t)$ at time $t = 0$ by computing the minimum distance from the zero state to each state $s = 0 \ldots N-1$. Steps 3 and 4 update the distance metrics into each state at time $t+1$. At step 5, if the distance measure for the path remerging with the zero state is less than the minimum distance measure to any other state then the algorithm has found the smallest measure path and terminates.

Definitions

$b(s_1, s_2)$: The set of branch symbols connecting states s_1 and s_2.

$m(s, t)$: The minimum distance metric entering state s at time t.

c_0 : The output symbol on the zero state branch.

Input : The number of states N.

The $b(s_1, s_2)$ for $0 \leq s_1 < N$, $0 \leq s_2 < N$ defining the trellis.

Output : The minimum distance measure $\mathbf{d}_{\min}$.

1. $t = 0$
2. $\mathbf{m}(s, t) = \min_{\substack{c \in b(0,s) \\ c \neq c_0}} \mathbf{d}(c_0, c)$, $0 \leq s < N$.
3. t=t+1
4. $\mathbf{m}(s, t) = \min_{\substack{0 \leq s' < N \\ c \in b(s', s)}} \{\mathbf{d}(c_0, c) \oplus \mathbf{m}(s', t-1)\}$, $0 \leq s < N$
5. if $\min_{0<s<N} \mathbf{m}(s, t) > \mathbf{m}(0, t)$ then goto 3
6. output $\mathbf{m}(0, t)$

Table 3.4: The algorithm to compute the minimum Hamming distance and associated minimum product distance of a trellis code.

3.5.3 Bounds on the Hamming distance of trellis codes.

Before any code searches are performed it is useful to establish an upper bound on the diversity that can be achieved by a multiple symbol per trellis branch encoder for a given number of states and spectral efficiency (measured in bits transmitted per symbol).

Theorem 1 *The diversity l, of an $N = 2^v$ state trellis code transmitting L symbols per branch and r bits per symbol is upperbounded by*

$$l \leq L \times \lfloor \frac{v}{rL} \rfloor + L \tag{3.33}$$

where the notation $\lfloor \bullet \rfloor$ rounds its argument to the nearest lowest integer.

Proof:
The number of bits transmitted per state transition in the trellis is rL and hence the number of branches, b, leaving each state is

$$b = 2^{rL} \tag{3.34}$$

Consider the branches starting from a common state s. There must exist two remerging paths after at most k steps through the trellis when $b^{k-1} > N$. Since L symbols are transmitted per branch the greatest diversity of such a path is kL, leading to equation (3.33). In figure 3.12 we illustrate the argument behind the bound. We have a 4 state trellis transmitting $L = 2$ symbols per branch at a rate of $\frac{1}{2}$ bit per

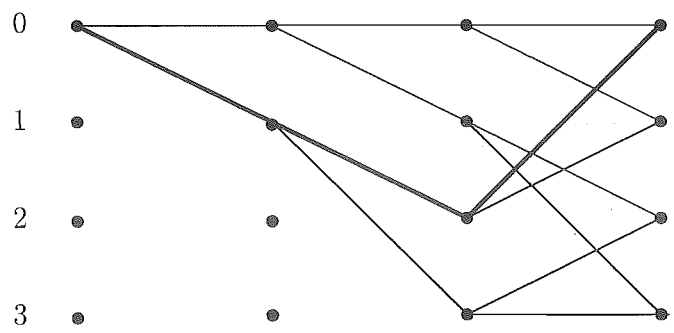


Figure 3.12: An example trellis illustrating the bound on the maximum Hamming distance of a trellis code. $L = 2$ and $r = \frac{1}{2}$.

symbol. The number of branches leaving a state is therefore $b = 2^{rL} = 2$. Starting

from the zero state, two states can be reached after one step, then after two steps four states can be reached, and after three steps, eight states can be reached. However the trellis only has four states and therefore there must be a path remerging with the zero state. This path is in bold face. Therefore $k = 3$ and the maximum diversity l is bounded to be less than or equal to 6.

A necessary condition for meeting equation (3.33) with equality is that the number of points in the underlying constellation is at least

$$M \geq 2^{rL}. \tag{3.35}$$

This is due to 2^{rL} branches leaving any state and to achieve the bound there must be at least that many symbols in our alphabet. Table 3.5 lists the results of the application of equation (3.33) for bit rates of $r = 1, 2, 3$ and values of L from 1 to 4. There are a number of trends to be noted from table 3.5 and equation (3.33).

v	$L=1$ bits/symbol			$L=2$ bits/symbol			$L=3$ bits/symbol			$L=4$ bits/symbol		
	1	2	3	1	2	3	1	2	3	1	2	3
1	2	1	1	2	2	2	3	3	3	4	4	4
2	3	2	1	4	2	2	3	3	3	4	4	4
3	4	2	2	4	2	2	6	3	3	4	4	4
4	5	3	2	6	4	2	6	3	3	8	4	4
5	6	3	2	6	4	2	6	3	3	8	4	4
6	7	4	3	8	4	4	9	6	3	8	4	4
7	8	4	3	8	4	4	9	6	3	8	4	4
8	9	5	3	10	6	4	9	6	3	12	8	4

Table 3.5: Maximum attainable diversity of multiple symbol per branch trellis codes.

For larger L, there is a potential for greater diversity for a low number of states. To take advantage of the greater diversity offered by codes using a higher L requires a move to a far greater number of states. For example for an $L = 4$ code transmitting 2 bits/symbol can achieve a diversity of 4 easily for a low number of states but no improvement can be made until the number of states is increased to 256. Careful study of table 3.5 shows that there are potentially a number of particularly good

codes. For example, $L = 2$, $r = 2$ and $v = 4$; $L = 3$, $r = 2$ and $v = 6$; or $L = 4$, $r = 1$ and $v = 4$.

The attainable squared product distance for PSK based codes is bounded by

$$d^2_{p_{max}} \leq 4^l, \tag{3.36}$$

where l is the diversity of the code. The upper bound of equation (3.36) is a result of the greatest squared Euclidean distance between two signal points in a PSK constellation equalling 4.

3.6 Summary.

In this chapter we have reviewed the technique of signal space coding with maximum likelihood decoding in context of the chapter 4, 6 and 7. Signal space codes have the advantage over hard decision coding schemes of making use of the reliability information of the symbols available at the receiver. We have examined the signal space code design criteria for the Rayleigh fading channel and compared them to those of the AWGN channel. An upperbound on the maximum Hamming distance of trellis codes has been derived.

Chapter 4

Geometrically Uniform Codes

The paper by Forney [41] describes the concept of *geometric uniformity* of a set points in Euclidean space. Geometric uniformity means the constellation (or code) looks the same when viewed from any signal point (or codeword), relative to an associated distance measure and implies that the error performance of the code may be completely characterised by considering only one codeword. The work by Benedetto *et al.* [2],[3], applies the theory of GU codes to construct good codes for the additive white Gaussian noise (AWGN) channel. Forney's definition of geometric uniformity is based on the squared Euclidean distance measure between points in signal space, which is the most important coding parameter for codes operating on the Gaussian channel, because the error performance of the code is an exponentially decreasing function with increasing minimum Euclidean distance [75], and under ideal conditions, Euclidean distance is also the metric for ML decoding. However, channels, such as the Rayleigh fading channel or Rician fading channel, have other distance measures associated with them. For the Rayleigh fading channel the error performance is inversely proportional to the signal-to-noise ratio (SNR) raised to the power of the Hamming distance of the code and inversely proportional to the squared product distance of the code [22]. However, the maximum likelihood soft decoding metric for coherent detection and ideal CSI is Euclidean distance. Therefore, for a code to be considered GU for the Rayleigh fading channel it must not only look the same from every codeword with respect to Euclidean distance, but also with respect to Hamming and squared product distance. For this reason we extend the definition of the geometric uniformity of a signal constellation such that the constellation is

geometrically uniform with respect to *sets* of distance measures, which allows the error performance of the code to be evaluated in a similar manner to the AWGN channel, i.e. by considering only the all-zero codeword. We begin by discussing distance measures between points in R^N.

4.1 Distance measures.

Definition 2 *Consider two points* $\mathbf{x}$ *and* $\mathbf{y}$ *in the space* R^N. *A* distance measure $D(\mathbf{x}, \mathbf{y})$ *is defined as a function of* $\mathbf{x}$ *and* $\mathbf{y}$ *such that*

$$D(\mathbf{x}, \mathbf{y}) = D(\mathbf{y}, \mathbf{x}) \tag{4.1}$$

and

$$D(\mathbf{x}, \mathbf{y}) \geq 0 \tag{4.2}$$

Distance measures are very useful in the context of communication systems. Typically they are used as a parameter or set of parameters characterising the expected performance of the system. For example the minimum squared Euclidean distance determines the performance of a coded system operating on the Gaussian channel, while squared product distance and Hamming distance characterise the error performance of the Rayleigh fading channel. Distance measures are also useful in the context of decoding, where with some additional properties to those of equations (4.1) and (4.2) they are referred to as *metrics*. A decoder computes the metric, appropriate to the channel and method of detection, between the received vector and each postulated codeword, and selects that codeword which minimises or maximises the metric. The form of a distance measure or metric $D(\mathbf{x}, \mathbf{y})$ is generally dependent on the problem for which it is derived. In the context of communication systems, and particularly coded systems, a useful general form for distance measures is as follows. First we write the vectors $\mathbf{x}$ and $\mathbf{y}$ as L-tuples of subvectors of length l:

$$\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_L), \ \mathbf{x}_i \in R^l \tag{4.3}$$

$$\mathbf{y} = (\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_L), \ \mathbf{y}_i \in R^l \tag{4.4}$$

and define a distance measure $D_l(\mathbf{x}_i, \mathbf{y}_i)$ over the subspace R^l. We then write the general form of a distance measure $D(\mathbf{x}, \mathbf{y})$ as

$$D(\mathbf{x}, \mathbf{y}) = f(d_1, d_2, \ldots, d_l), \tag{4.5}$$

i.e. $D(\mathbf{x}, \mathbf{y})$ is a function of the distance measures d_i given by:

$$d_i = D_l(\mathbf{x}_i, \mathbf{y}_i). \tag{4.6}$$

Equation (4.5) is general. Four special cases of interest are

1. **Squared Euclidean distance**: Denoted by $d^2(\mathbf{x}, \mathbf{y})$. Here $l = 1$, $D_l(x, y) = |x - y|^2$ and the function $f(d_1, d_2, \ldots, d_L)$ is given by

$$f(d_1, d_2, \ldots, d_L) = \sum_{i=1}^{L} d_i. \tag{4.7}$$

   Hence $d^2(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{L} |x_i - y_i|^2$.

2. **Generalised product distance**: Denoted by $d_{gp}(\mathbf{x}, \mathbf{y})$. Here l and D_l are arbitrary and $f(d_1, d_2, \ldots, d_L)$ is defined by:

$$f(d_1, d_2, \ldots, d_L) = \prod_{\substack{i=1 \\ \mathbf{x}_i \neq \mathbf{y}_i}}^{L} D_l(\mathbf{x}_i, \mathbf{y}_i). \tag{4.8}$$

   Note that the product is only over the i such that $\mathbf{x}_i \neq \mathbf{y}_i$.

3. **Squared product distance**: Denoted by $d_p^2(\mathbf{x}, \mathbf{y})$. This is a special case of the generalised product distance where $l = 2$ and $D_l = |\mathbf{x}_i - \mathbf{y}_i|^2 = d^2(\mathbf{x}_i, \mathbf{y}_i)$. Hence

$$d_p^2(\mathbf{x}, \mathbf{y}) = \prod_{\substack{i=1 \\ \mathbf{x}_i \neq \mathbf{y}_i}}^{L} |\mathbf{x}_i - \mathbf{y}_i|^2. \tag{4.9}$$

4. **Hamming distance**: Denoted by $d_H(\mathbf{x}, \mathbf{y})$. Here l is arbitrary but in our context usually equal to two. The distance measure $D_l(\mathbf{x}_i, \mathbf{y}_i)$ is

$$D_l(\mathbf{x}_i, \mathbf{y}_i) = \begin{cases} 0 & \mathbf{x}_i = \mathbf{y}_i \\ 1 & \mathbf{x}_i \neq \mathbf{y}_i \end{cases} \tag{4.10}$$

   and $f(d_1, d_2, \ldots, d_L) = \sum_{i=1}^{L} d_i$. Hence Hamming distance is defined by

$$d_H(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{L} \begin{cases} 0 & \mathbf{x}_i = \mathbf{y}_i \\ 1 & \mathbf{x}_i \neq \mathbf{y}_i \end{cases}. \tag{4.11}$$

Squared Euclidean distance is the most common distance measure occurring in communication systems. It characterises the performance of a coded system operating on the Gaussian channel and is also the ML decoding metric for both the coherently detected Gaussian channel, and as shown in chapter 2, the metric for decoding sequences transmitted over the Rayleigh fading channel assuming ideal CSI.

Special cases of the generalised product distance occur in the characterisation of the error performance of codes operating on the Rayleigh fading channel. For example the pairwise probability of error is inversely proportional to the squared product distance between code sequences. In chapter 7 on multiple symbol differential detection we present a special case of the generalised product distance important for differential detection.

The Hamming distance as defined by equation (4.11) is a measure of the number of subvectors different between two vectors. It is generally useful only when measured between vectors of elements drawn from a common finite alphabet. Hamming distance determines the asymptotic slope of the probability of error characteristic of codes operating on the Rayleigh fading channel. Stemming from this discussion we make the following definitions:

Definition 3 *We define the set $\mathcal{D}$ to denote a set of distance measures over R^N, i.e.*

$$\mathcal{D} = \{D_1, D_2, \ldots, D_n\} \tag{4.12}$$

Note that each D_i is defined over the same number of dimensions N.

Definition 4 *We will denote the set of distance measures pertinent to the additive white Gaussian noise (AWGN) channel by*

$$\mathcal{D}_A = \{d^2\} \tag{4.13}$$

and the set of distance measures pertinent to the Rayleigh fading channel by

$$\mathcal{D}_R = \{d^2, d_p^2, d_H\}. \tag{4.14}$$

Now that distance measures are understood we continue the discussion with mappings that are invariant under a distance measure or set of distance measures.

4.2 Isometries.

Consider a real N-dimensional space R^N with an associated *set* of distance measures $\mathcal{D}(\mathbf{x},\mathbf{y}) = \{D_1(\mathbf{x},\mathbf{y}), D_2(\mathbf{x},\mathbf{y}), \ldots, D_n(\mathbf{x},\mathbf{y})\}$ for $\mathbf{x} \in R^N$, $\mathbf{y} \in R^N$, where each distance measure $D_i(\mathbf{x},\mathbf{y})$ is as defined by definition 2.

Definition 5 *An* isometry *relative to the set of distance measures*

$$\mathcal{D} = \{D_1(\mathbf{x},\mathbf{y}), D_2(\mathbf{x},\mathbf{y}), \ldots, D_n(\mathbf{x},\mathbf{y})\} \tag{4.15}$$

is a mapping $u : R^N \rightarrow R^N$ *such that, given any two points* $\mathbf{x} \in R^N$, $\mathbf{y} \in R^N$, $D_i(u(\mathbf{x}), u(\mathbf{y})) = D_i(\mathbf{x},\mathbf{y})$ *for all* $D_i \in \mathcal{D}$, *i.e. the mapping is distance invariant under each distance measure* D_i *in the set* $\mathcal{D}$.

This definition is a generalisation of the definition of an isometry used by Forney [41], which is defined as an invariant mapping under Euclidean distance. The two are equivalent for $\mathcal{D} = \{d^2(\mathbf{x},\mathbf{y})\}$.

Definition 6 *We denote the set* $U(D)$ *to be the set of isometries in the space* R^N *relative to the distance measure* D, *and* $U(\mathcal{D})$ *to be the set of isometries relative to* $\mathcal{D}$.

Note that the size of $U(D)$ is a function of the number of dimensions N over which D is defined.

Theorem 2 *Let* $\mathcal{D}$ *be a set of* n *distance measures,* $\mathcal{D} = \{D_1, D_2, \ldots D_n\}$. *The set of isometries relative to the set of distance measures* $\mathcal{D}$ *is*

$$U(\mathcal{D}) = \bigcap_{i=1}^{n} U(D_i) \tag{4.16}$$

Proof: From set theory and definition 5 it is clear that $U(\mathcal{D})$ must be the intersections of $U(D_i)$.

4.2.1 Isometries under Euclidean distance.

First we will discuss isometry mappings under the distance measure of Euclidean distance, i.e. $\mathcal{D}(\mathbf{x},\mathbf{y}) = \{d^2(\mathbf{x},\mathbf{y})\}$. From the study of geometry, two classes of isometries for squared Euclidean distance are:

1. *Translations:* $t(\mathbf{x}) = \mathbf{x} + \tau$, for $\tau \in R^N$.

2. *Orthogonal transformations:* $r(\mathbf{x}) = A\mathbf{x}$, where A is an *orthogonal matrix*; i.e., A is an $N \times N$ matrix such that $A^T A = I$, the identity matrix. It follows from the orthogonality of A that $detA = \pm 1$. If $detA = +1$, then r is a pure *rotation*; if $detA = -1$, then r is a rotation with reflection and if $detA = -1$ and $A^2 = I$ then r is a pure *reflection.*

It can be shown that any isometry u of R^N can be uniquely expressed as the composition of an orthogonal transformation A with a translation t:

$$u(\mathbf{x}) = A\mathbf{x} + \tau, \tag{4.17}$$

where A is an orthogonal matrix and τ is an element of R^N. Figure 4.1 illustrates a translation, a pure rotation, and a pure reflection. For example, in the space R^2,

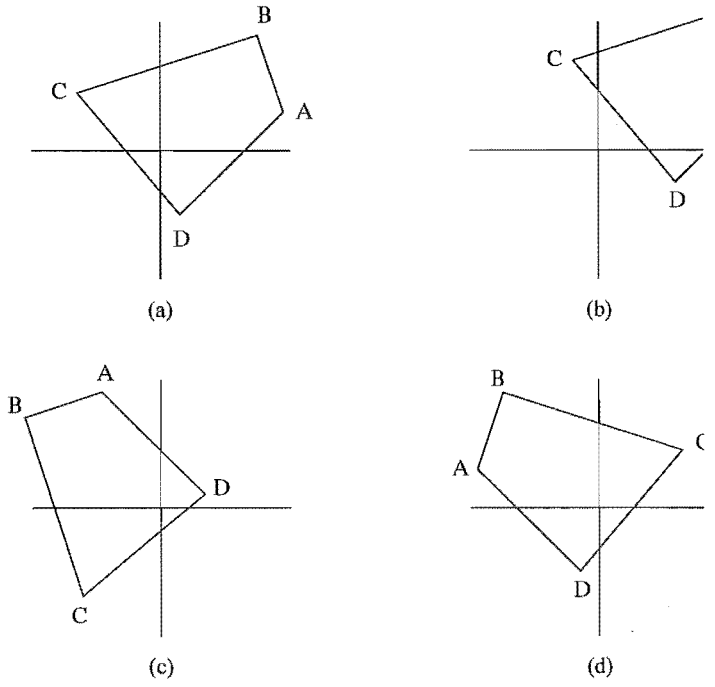


Figure 4.1: Isometry mappings of a figure ABCD in R^2. (a) The original. (b) A translation. (c) A pure anti-clockwise rotation of 90° about the origin. (d) A pure reflection about the vertical axis.

the orthonormal transformation matrices can be shown to be one of the two forms:

1. **Pure rotation**: Denoted by the matrix R_ϕ and given by

$$R_\phi = \begin{pmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{pmatrix} \tag{4.18}$$

and represents an anti-clockwise rotation about the origin by the angle ϕ. It is a pure rotation because the determinant of R_ϕ is $\det R_\phi = \cos^2\phi + \sin^2\phi = 1$. For example the 90° anti-clockwise rotation about the origin of figure 4.1(c) is represented by the matrix

$$R_{90^\circ} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}. \tag{4.19}$$

2. **Pure reflection**: Denoted by the matrix V_ϕ

$$V_\phi = \begin{pmatrix} \cos 2\phi & \sin 2\phi \\ \sin 2\phi & -\cos 2\phi \end{pmatrix} \tag{4.20}$$

and represents a reflection about the line passing through the origin and meeting the horizontal at angle ϕ. It is a pure reflection because $\det V_\phi = -\cos^2\phi - \sin^2\phi = -1$ and $(V_\phi)^2 = I$. For example the reflection about the vertical axis of figure 4.1(d) is represented by the matrix

$$V_{90^\circ} = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}. \tag{4.21}$$

4.2.2 Isometries under generalised product distance.

Given two vectors, written as a concatenation of length l subvectors, $\mathbf{x} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_L\}$, $\mathbf{y} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_L\}$ and a distance measure D_l over R^l, the generalised product distance from equation (4.8) is defined by

$$d_{gp}(\mathbf{x}, \mathbf{y}) = \prod_{\substack{i=1 \\ \mathbf{x}_i \neq \mathbf{y}_i}}^{L} D_l(\mathbf{x}_i, \mathbf{y}_i) \tag{4.22}$$

From this definition we may readily define three classes of isometries relative to d_{gp}:

1. Permutation mappings p over the subvectors of $\mathbf{x}_i$. Clearly if the subvectors of $\mathbf{x}$ and $\mathbf{y}$ are permuted in the same manner, then the distance measure d_{gp}

is invariant by the commutative property of the multiplications in equation (4.22). A permutation mapping may be written as

$$p(\mathbf{x}) = A_p \mathbf{x} \tag{4.23}$$

where A_p is a permutation matrix. From combinatorics, the number of distinct permutation mappings is $L!$. An example of a permutation matrix A_p for $L = 3$ and $l = 2$ is

$$A_p = \begin{pmatrix} 0 & I & 0 \\ I & 0 & 0 \\ 0 & 0 & I \end{pmatrix} \tag{4.24}$$

where $I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$. This instance of A_p swaps the first two elements of the vector it operates on.

2. The mapping u of $\mathbf{x} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_L\}$ of the form

$$u(\mathbf{x}) = \{u_1(\mathbf{x}_1), u_2(\mathbf{x}_2), \ldots, u_L(\mathbf{x}_L)\} \tag{4.25}$$

where each $u_i(\mathbf{x}_i)$ is an isometry relative to the distance measure D_l, i.e. $D_l(u_i(\mathbf{x}_i)) = D_l(\mathbf{x}_i)$. The mapping u is an isometry because

$$\begin{aligned} d_{gp}(u(\mathbf{x}), u(\mathbf{y})) &= \prod_{\substack{i=1 \\ u_i(\mathbf{x}_i) \neq u_i(\mathbf{y}_i)}}^{L} D_l(u_i(\mathbf{x}_i), u_i(\mathbf{y}_i)) \\ &= \prod_{\substack{i=1 \\ \mathbf{x}_i \neq \mathbf{y}_i}}^{L} D_l(\mathbf{x}_i, \mathbf{y}_i) \\ &= d_{gp}(\mathbf{x}, \mathbf{y}). \end{aligned} \tag{4.26}$$

3. Compositions of permutation mapping p and the mappings u defined by equation (4.25). If both p and u are distance invariant relative to d_{gp} then the maps $p(u(\mathbf{x}))$ and $u(p(\mathbf{x}))$ are invariant also.

4.2.3 Isometries under squared product distance.

Squared product distance is a special case of the generalised product distance and is defined by:

$$d_p^2(\mathbf{x}, \mathbf{y}) = \prod_{\substack{i=1 \\ \mathbf{x}_i \neq \mathbf{y}_i}}^{L} |\mathbf{x}_i - \mathbf{y}_i|^2, \tag{4.27}$$

which implies $D_l(\mathbf{x}_i, \mathbf{y}_i) = d^2(\mathbf{x}_i, \mathbf{y}_i)$, the squared Euclidean distance measure, and l is typically equal to two. We may combine the results of sections 4.2.1 and 4.2.2 to define the three classes of isometries of the distance measure $d_p^2(\mathbf{x}, \mathbf{y})$.

1. Permutations of the length l subvectors of $\mathbf{x}_i$. We write this isometry as

$$p(\mathbf{x}) = A_p\mathbf{x} \tag{4.28}$$

where A_p is a permutation matrix.

2. From section 4.2.1 we know that the isometries of squared Euclidean distance are of the form $u = A\mathbf{x} + \tau$, where A is an orthonormal matrix, hence from equation (4.25) a second class of isometries of $d_p^2(\mathbf{x}, \mathbf{y})$ may be written as

$$u(\mathbf{x}) = A_M\mathbf{x} + \tau \tag{4.29}$$

where

$$A_M = \begin{pmatrix} A_1 & & & \\ & A_2 & & \\ & & \ddots & \\ & & & A_L \end{pmatrix} \tag{4.30}$$

and

$$\tau = \begin{pmatrix} \tau_1 \\ \tau_2 \\ \vdots \\ \tau_L \end{pmatrix}. \tag{4.31}$$

Each A_i is an $l \times l$ orthonormal matrix, and each τ_i is a length l vector.

3. The composition of permutations and u:

$$\begin{aligned} u(\mathbf{x}) &= A_p(A_M\mathbf{x} + \tau) \\ &= A_pA_M\mathbf{x} + A_p\tau \\ &= A\mathbf{x} + \tau' \end{aligned} \tag{4.32}$$

which is of the form of equation (4.17), the isometries under Euclidean distance. The isometries relative to squared product distance are a subset of the isometries

relative to squared Euclidean distance because $\det A = \det A_p \det A_M = \pm 1$, and A is orthonormal because $A^T A = (A_p A_M)^T A_p A_M = I$. Hence we have the relationship:

$$U(d_p^2) \subseteq U(d^2), \tag{4.33}$$

which states that the isometries relative to squared product distance are a subset of the isometries relative to squared Euclidean distance. Equality of equation (4.33) is met for $l = L$, i.e. then we have $d_p^2 \equiv d^2$.

4.2.4 Isometries under Hamming distance.

Hamming distance is defined by

$$d_H(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{L} \begin{cases} 0 & \mathbf{x}_i = \mathbf{y}_i \\ 1 & \mathbf{x}_i \neq \mathbf{y}_i \end{cases} . \tag{4.34}$$

And similarly to the generalised product distance of section 4.2.2 there are three classes of isometries given by

1. Permutations over the subvectors $\mathbf{x}_i$.

2. The mapping u of $\mathbf{x} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_L\}$ of the form

$$u(\mathbf{x}) = \{u_1(\mathbf{x}_1), u_2(\mathbf{x}_2), \ldots, u_L(\mathbf{x}_L)\} \tag{4.35}$$

   where each $u_i(\mathbf{x}_i)$ is an isometry relative to the distance measure D_H, i.e. $D_H(u_i(\mathbf{x}_i)) = D_H(\mathbf{x}_i)$, which implies u_i is such that

$$\text{if } \mathbf{x}_i \neq \mathbf{y}_i \text{ then } u_i(\mathbf{x}_i) \neq u_i(\mathbf{y}_i). \tag{4.36}$$

3. The compositions $p(u(\mathbf{x}))$ and $u(p(\mathbf{x}))$ of permutation mapping p and the mappings u.

An example of an isometry mapping of the form of equation (4.35) and satisfying equation (4.36) is $u_i = d^2$. In general we have

$$U(d_p^2) \subseteq U(d_H). \tag{4.37}$$

i.e. the isometries relative to squared product distance are a subset of the isometries relative to Hamming distance.

4.2.5 Isometries under $\mathcal{D}_R$.

The set of distance measures denoted by the symbol $\mathcal{D}_R$ is defined in section 4.1 and contains the elements

$$\mathcal{D}_R = \{d^2, d_p^2, d_H\} \tag{4.38}$$

and is important for the analysis of the performance of a coherent communication system operating on the Rayleigh fading channel. From theorem 2 of section 4.2 the set of isometries relative to $\mathcal{D}_R$ are

$$\begin{aligned} U(\mathcal{D}_R) &= U(d^2) \cap U(d_p^2) \cap U(d_H) \\ &= U(d_p^2) \end{aligned} \tag{4.39}$$

where the result has been obtained from the application of equations (4.33) and (4.37), i.e. the isometries relative to $\mathcal{D}_R$ are the isometries relative to the distance measure d_p^2.

4.3 Symmetries of constellations.

So far we have discussed mappings over a continuous space that are invariant over a distance measure or a set of distance measures. Now we apply these mappings to a signal constellation S, a set of discrete points in R^N, and define a symmetry of S:

Definition 7 *Let S be a signal constellation in a real N-dimensional space R^N. An isometry under the set of distance measures $\mathcal{D}(\mathbf{x}, \mathbf{y})$ which maps a constellation S to itself is called a* symmetry *of S relative to the set of distance measures $\mathcal{D}(\mathbf{x}, \mathbf{y})$.*

In general, the set of symmetries of S forms a group under the composition of mappings, called the *symmetry group* $\Gamma(S, \mathcal{D})$ of S, relative to the distance measures $\mathcal{D}$. When the set $\mathcal{D}$ is clear, or not relevant from the context of the discussion, we simply write $\Gamma(S)$ to denote the group of symmetries of S. Clearly $\Gamma(S, \mathcal{D}) \subseteq U(S, \mathcal{D})$.

The class of conventional balanced M-PSK constellations defined in two dimensions, hence $l = 2$ and $L = 1$, is an example of a set of highly symmetrical signal constellations. A member of this class is $S = 8$-PSK shown in figure 4.2. The symmetries of S relative to the set $\mathcal{D}_R$ are discrete rotations and reflections about

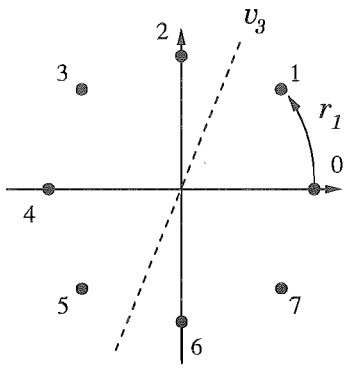


Figure 4.2: Two example symmetries of an 8-PSK constellation.

the origin. Equation (4.18) gives the general form of a rotation in R^2 and the set of rotation symmetries of 8-PSK are the matrices R_{ϕ_i}, where $\phi_i = \frac{\pi i}{4}$ radians, for $i = 0, 1, \ldots, 7$. Similarly equation (4.20) gives the general form of a reflection in R^2 and the set of reflection symmetries of 8-PSK are the matrices V_{ϕ_i}, where $\phi_i = \frac{\pi i}{2}$ radians, for $i = 0, 1, \ldots, 7$. A shorthand notation to denote these symmetries are the symbols $r_0, r_1, \ldots, r_7$ to represent the corresponding eight rotation symmetries and $v_0, v_1, \ldots, v_7$ to represent the corresponding eight reflection symmetries. These sixteen symmetries form the symmetry group of $S = 8\text{-PSK}$:

$$\Gamma(S_{8\text{-PSK}}) = \{r_0, r_1, r_2, r_3, r_4, r_5, r_6, r_7, v_0, v_1, v_2, v_3, v_4, v_5, v_6, v_7\} \tag{4.40}$$

which is isomorphic to the dihedral group D_8 (see appendix D reviewing the fundamentals of group theory).

4.3.1 Symmetries of M-PSK.

In general the symmetries of M-PSK relative to the set of distance measures $\mathcal{D}_R$ is the set of orthogonal transformations

$$\Gamma(S_{M-\text{PSK}}, \mathcal{D}_R) = \{R_{\phi_i}, V_{\phi_i}\} \tag{4.41}$$

where $i = 0, 1, \ldots, M-1$, the matrices R_{ϕ_i} and V_{ϕ_i} are defined by equations (4.18) and (4.20) respectively, and $\phi_i = \frac{2\pi i}{M}$. The symmetry group is isomorphic to D_M.

4.3.2 Symmetries of L × M-PSK constellations.

The L-fold Cartesian product of an M-PSK constellation S is a multi-dimensional constellation existing in R^{2L} and is written as S^L. If $\Gamma(S)$ is the symmetry group of S, then $[\Gamma(S)]^L$ is certainly a group of symmetries of S^L but it is not the *complete* symmetry group of S^L. Benedetto *et al.* [3] have shown that the symmetry group of an $L \times M$-PSK constellation relative to the distance measure set $\mathcal{D}_A$ is isomorphic to:

$$\Gamma(S^L) \simeq \begin{cases} P_{2L} \bullet (Z_2)^{2L} & M = 4 \\ P_L \bullet (D_M)^L & M \text{ even and } M > 4 \end{cases} \tag{4.42}$$

where P_n is the symmetric permutation group on a set of n elements, and the operator $\bullet$ represents a direct product. The case for $M = 4$ is special due to the additional symmetry resulting from the ability to write a 4-PSK constellation as the square of the 1-dimensional GU constellation $\{-1, 1\}$. The number of elements in $\Gamma(S^L)$ is given by

$$|\Gamma(S^L)| = \begin{cases} (2L)!2^{2L} & M = 4 \\ L!(2M)^L & M \text{ even and } M > 4 \end{cases} \tag{4.43}$$

Theorem 3 *The symmetries of $L \times M$-PSK relative to the set $\mathcal{D}_R$ are the* same *as the symmetries of $L \times M$-PSK relative to $\mathcal{D}_A$ for $M > 4$, while for $M = 4$, they are a subset of $\mathcal{D}_A$. i.e.*

$$\Gamma(S^L, \mathcal{D}_R) \subset \Gamma(S^L, \mathcal{D}_A) \quad M = 4 \tag{4.44}$$

$$\Gamma(S^L, \mathcal{D}_R) = \Gamma(S^L, \mathcal{D}_A) \quad M > 4 \tag{4.45}$$

Proof: Since $U(\mathcal{D}_R) \subseteq U(\mathcal{D}_A)$ we must have

$$\Gamma(S, \mathcal{D}_R) \subseteq \Gamma(S, \mathcal{D}_A). \tag{4.46}$$

We have shown that the isometries of $\mathcal{D}_R$ are of the form

$$u(\mathbf{x}) = A_p(A_M\mathbf{x} + \tau) \tag{4.47}$$

Clearly translations need not be considered as symmetries of $L \times M$-PSK and $\tau = 0$. Equation (4.47) then reduces to the linear transformation

$$u(\mathbf{x}) = A_p A_M \mathbf{x} \tag{4.48}$$

where A_p is a permutation matrix over L elements of dimension l, and A_M is of the form of equation (4.30). For u to be an element of $\Gamma(S^L, \mathcal{D}_R)$, each 2×2 submatrix A_i of A_M must represent a symmetry of M-PSK given by equation (4.41). Hence the number of elements of $\Gamma(S^L, \mathcal{D})$ is $|A_p||A_M| = L! \times (2M)^L$. For $M > 4$ we have $|\Gamma(S^L, \mathcal{D}_R)| = |\Gamma(S^L, \mathcal{D}_A)|$, and $\Gamma(S^L, \mathcal{D}_R) \subseteq \Gamma(S, \mathcal{D}_A)$, and therefore

$$\Gamma(S^L, \mathcal{D}_R) = \Gamma(S^L, \mathcal{D}_A), \quad M > 4 \tag{4.49}$$

For $M = 4$ we have $|\Gamma(S^L, \mathcal{D}_R)| = L! \times 4^L \neq |\Gamma(S^L, \mathcal{D}_A)|$ except for $L = 1$, and

$$\Gamma(S^L, \mathcal{D}_R) = \Gamma(S^L, \mathcal{D}_A), \quad M = 4, \quad L = 1 \tag{4.50}$$

$$\Gamma(S^L, \mathcal{D}_R) \subset \Gamma(S^L, \mathcal{D}_A), \quad M = 4, \quad L > 1, \tag{4.51}$$

completing the proof.

The composition of two symmetries $u_1(\mathbf{x}) = A_{p_1} A_{M_1}$ and $u_2(\mathbf{x}) = A_{p_2} A_{M_2}$ may be written as

$$\begin{aligned} u_1(u_2(\mathbf{x})) &= A_{p_1} A_{M_1} A_{p_2} A_{M_2} \mathbf{x} \\ &= (A_{p_1} A_{p_2}^{-1})(A_{p_2} A_{M_1} A_{p_2} A_{M_2}) \mathbf{x} \\ &= A_p A_M \mathbf{x} \end{aligned} \tag{4.52}$$

which is a matrix in the form of equation (4.48) and equation (4.52) may be used to define the operation table of the symmetry group of $L \times M$-PSK.

4.3.3 Notation for symmetries of L × M-PSK.

The codes presented in this chapter are based on $L \times M$-PSK constellations and the description of the code generators are in terms of vectors of elements of the symmetry group of the underlying constellation. We have shown that the symmetries of $L \times M$-PSK relative to $\mathcal{D}_R$ are of the form

$$u(\mathbf{x}) = A_p A_M \mathbf{x} \tag{4.53}$$

where A_p is a permutation matrix over L elements and A_M is a matrix of the form of equation (4.30), where each submatrix A_i corresponds to a symmetry of M-PSK. A shorthand notation is to write a symmetry u as a concatenation of two shorthand notations corresponding to A_p and A_M respectively. The matrix

A_p represents a permutation over L elements and can be written from the group algebra of permutation groups as the resultant vector of the permutation operating on the vector $(1, 2, \ldots, L)$. For example the permutation represented by the matrix of equation (4.24) may be written as

$$p = (2, 1, 3) \tag{4.54}$$

or using a cycle notation [60] as

$$p = (1, 2) \tag{4.55}$$

The matrix A_M can be described by a vector $\mathbf{m}$ of L elements, where each element represents a matrix A_i. From equation (4.41) we can represent each A_i by an element from the set $\{r_0, r_1, \ldots, r_{M-1}, v_0, v_1, \ldots, v_{M-1}\}$. Hence we write u as the concatenation of p and $\mathbf{m}$.

$$u = p\mathbf{m}. \tag{4.56}$$

An example is

$$u = (1, 2)(r_2, r_1, v_3). \tag{4.57}$$

If permutation symmetries are not considered then u is written simply as a vector over elements of $\Gamma(S_{M\text{-PSK}})$.

4.4 Geometrically uniform signal sets.

The comprehensive work by Forney [41] defines the conditions for geometric uniformity of a signal set S in terms of the symmetry mappings of S. Now we extend the Forney's definition of the geometrical uniformity of a signal set to the geometrical uniformity of a signal set relative to a set of distance measures $\mathcal{D}$:

Definition 8 *Let S be a signal set and $\mathcal{D}(\mathbf{x}, \mathbf{y})$ a set of distance measures. The signal set S is geometrically uniform relative to $\mathcal{D}(\mathbf{x}, \mathbf{y})$, if for any two points s and s' in S, there exists a symmetry u relative to $\mathcal{D}(\mathbf{x}, \mathbf{y})$ of S that maps s to s', while leaving S invariant, i.e.*

$$u(s) = s', \tag{4.58}$$

$$u(S) = S. \tag{4.59}$$

For example consider the constellation S = 8-PSK shown in figure 4.2, and the set of distance measures $\mathcal{D}_R$. The symmetry group relative to $\mathcal{D}_R$ is given by equation (4.40). By inspection there exists a symmetry that maps any point in S to any other point in S and therefore 8-PSK is GU with relative to $\mathcal{D}_R$.

4.4.1 Symmetry properties of GU signal sets.

Here we reproduce, in the context of our definitions, two of the main results by Forney [41] on the symmetry properties of GU signals sets. Essentially geometric uniformity of S implies that S looks the same when viewed from any of its points.

Decision regions.

Given a decoding metric $D(\mathbf{x}, \mathbf{y})$, the decision region $R_V(s)$ associated with a point s of a signal constellation S is

$$R_V(s) = \{\mathbf{x} \in R^N : D(\mathbf{x}, s) = \min_{s' \in S} D(\mathbf{x}, s')\}, \tag{4.60}$$

i.e. the set of points $\mathbf{x}$ in R^N where the metric $D(\mathbf{x}, s)$ is less than or equal to the metric $D(\mathbf{x}, s')$ measured between $\mathbf{x}$ and any other point $s' \in S$. If $D(\mathbf{x}, \mathbf{y}) \equiv d^2$ then the decision region of s is referred to as the Voronoi region of s.

Theorem 4 *If S is a geometrically uniform signal set relative to the distance measure set $\mathcal{D}$, and s and s' are points in S, then*

$$u(R_V(s)) = R_V(s') \tag{4.61}$$

where u is a symmetry mapping s to s'.

Proof: By definition, $\mathbf{x} \in R^N$ is in $R_V(s)$ if and only if

$$D(\mathbf{x}, s) = \min_{s'' \in S} D(\mathbf{x}, s'') \tag{4.62}$$

If s' is another point in S, then there exists a symmetry u such that $u(s) = s'$, and $u(S) = S$. If u is such a symmetry and $\mathbf{x}$ is in $R_V(s)$ then $u(\mathbf{x})$ is in $R_V(s')$ because

$$\begin{aligned} D(u(\mathbf{x}), s') &= D(u(\mathbf{x}), u(s)) \\ &= D(\mathbf{x}, s) \\ &= \min_{s'' \in S} D(\mathbf{x}, s'') \\ &= \min_{s'' \in S} D(u(\mathbf{x}), u(s'')) \\ &= \min_{s''' \in S} D(u(\mathbf{x}), s''') \end{aligned} \tag{4.63}$$

where as s'' ranges through S, so does $s''' = u(s'')$.

Distance profiles.

The distances seen by a point in a constellation S to the other points in S characterises the error probability of that point.

Definition 9 *The* global distance profile *associated with any point $s \in S$ and a distance measure D is the set of distances to all points in S measured by D:*

$$DP(s, D) = \{D(s, s'), s' \in S\}. \tag{4.64}$$

Theorem 5 *Let S be a geometrically uniform constellation relative to the set of distance measures $\mathcal{D}$. If $D \in \mathcal{D}$ then all points $s \in S$ have the same global distance profile $DP(s, D)$.*

Proof: Let s and s' be two points in S and u a symmetry of S mapping s to s'. The distance profile of s equals

$$\begin{aligned} DP(s, D) &= \{D(s, s''), s'' \in S\} \\ &= \{D(u(s), u(s'')), s'' \in S\} \\ &= \{D(s', u(s''), s'' \in S\} \\ &= \{D(s', y), y \in S\} \\ &= DP(s', D) \end{aligned} \tag{4.65}$$

where as s'' ranges through S so does $y = u(s'')$.

In the context of communications, the importance of theorem 4 is that all decoder decision regions have the same shape (assuming equiprobable signal point transmission). Theorem 5 implies the probability of error is identical for each transmitted signal point, and hence the probability of error of the system can be calculated by considering only one (arbitrary) signal point.

4.5 Generating groups.

The symmetry properties of geometrically uniform constellations have been characterised, and we may induce group properties on the constellation by defining a mapping between S and a group. Such groups are *generating groups* [41].

Definition 10 *Let S be a geometrically uniform constellation with symmetry group $\Gamma(S)$. A* generating group *$G(S)$ of S is a subgroup of $\Gamma(S)$ sufficient to generate S from any single point in S, i.e.*

$$\forall s_0 \in S \quad \{g(s_0), g \in G(S)\} = S \tag{4.66}$$

The map from $G(S)$ to S induces a group structure on S that is isomorphic to $G(S)$. In general a GU constellation has multiple generating groups. For example the constellation $S = 8$-PSK has exactly two generating groups given by

$$G_1(8\text{-PSK}) = \{r_0, r_1, r_2, r_3, r_4, r_5, r_6, r_7\} \tag{4.67}$$

and

$$G_2(8\text{-PSK}) = \{r_0, r_2, r_4, r_6, v_1, v_3, v_5, v_7\} \tag{4.68}$$

These two groups have a different group structure. The group G_1 is isomorphic to the abelian group Z_8, while G_2 is isomorphic to the non-abelian group D_4.

Benedetto *et al.* [3] have shown that a necessary and sufficient condition for a subgroup G of $\Gamma(S, \mathcal{D})$, such that $|G| = |S|$, to be a generating group of S is:

$$\begin{gathered} \forall g \in G, g \neq e \\ \nexists s_i \in S \text{ such that } g(s_i) = s_i, \end{gathered} \tag{4.69}$$

i.e. no element of G can leave unchanged an element of the constellation S. This definition may be used to help construct the generating groups of S as will be shown in section 4.8.3.

In general there are exactly two generating groups of an M-PSK constellation. One isomorphic to Z_M and one isomorphic to $D_{M/2}$.

4.6 Geometrically uniform constructions.

A very useful property of geometrically uniform constellations is the ability to construct more geometrically uniform structures starting from a GU constellation [41]. The two main constructions are:

1. **Powers of GU constellations**: If S is a GU constellation then S^L is also a GU constellation. If $G(S)$ is a generating group of S then $G^L(S)$ is a generating

group of S^L. If $\Gamma(S)$ is the symmetry group of S then in general we have

$$[\Gamma(S)]^L \subseteq P_L \bullet [\Gamma(S)]^L \subseteq \Gamma(S^L) \tag{4.70}$$

where P_L is the symmetric permutation group of L elements. Equation (4.70) implies that the complete symmetry group of S^L is not simply the L-fold direct product of the symmetry group of S and also includes other symmetries.

2. **Subgroups of generating groups:** If S is a GU constellation and $G(S)$ is a generating group of S then a normal subgroup G' of G induces a partitioning of S into S' and its cosets. The set S' is geometrically uniform and has G' as a generating group.

The proofs of these constructions are in Forney's paper [41]. The power of constructions 1 and 2 is that they can be applied recursively. Starting from a simple GU constellation S with a generating group G, we may apply construction 1 to create a GU constellation with L times the dimensionality. From construction 2, the subgroups of G^L define a large set of new geometrically uniform constellations. The creation of geometrically uniform trellis codes is based on the application of these constructions.

4.7 Geometrically uniform trellis codes.

Now that the properties of geometrically uniform signal sets have been established we are interested in systematically constructing signal space codes with the same properties as a GU constellation. From section 3.3 of chapter 3 we have shown that any signal space code may be written as a subset of a power of an underlying signal constellation, i.e.

$$\mathcal{C} \subset S^N. \tag{4.71}$$

A connection may readily be made with geometrically uniform constellations. We have seen from section 4.5 that a geometrically uniform constellation has associated with it a generating group $G(S)$. The direct powers $G^N(S)$ of $G(S)$ are generating groups for the geometrically uniform constellation S^N. Any subgroup H of $G^N(S)$ therefore defines a subset of S^N and forms a signal space code. From construction

2 of section 4.6 this set of points is geometrically uniform and forms a group code [69]. The comprehensive work by Forney and Trott [42] shows how to construct a minimal encoder for any group code based on the decomposition into elementary constituents, called *granules*. This decomposition is very general, and includes time-varying codes. We will reiterate some of the results by Forney and Trott in context of our discussion to explain the construction of the codes presented.

4.7.1 Linear codes over groups.

Definition 11 *A code $\mathcal{C}$ is a set of sequences $\mathcal{C} = \{c_k, k \in I\}$, defined on a discrete index I. The symbols c_k are drawn from groups G_k, $k \in I$.*

The code $\mathcal{C}$ is *linear* if the code sequences form a group under component-wise group operation. In the context of the codes defined in this chapter the code symbols are drawn from a generating group G, i.e. $G_k = G$ and $\mathcal{C}$ is called a linear code over G.

Definition 12 *Given a subset $J \subseteq I$, the* projection $P_J : W \to W$ *is the map that sends a sequence $g \in W$ to the sequence $h \in W$ defined by*

$$h_k = \begin{cases} g_k, & \text{if } k \in J \\ 0, & \text{if } k \notin J \end{cases} \tag{4.72}$$

It can be shown that the image of $\mathcal{C}$ under the projection P_J is a subgroup of $\mathcal{C}$ denoted $P_J(\mathcal{C})$. The subset J is typically an interval. For example, $P_{[m,n)}$ denotes the projection onto the interval $J = [m, n)$. In the notation for intervals, a square bracket is inclusive and a round bracket is exclusive. Therefore for the example interval $J = [m, n)$ we have $m \in J$, while $n \notin J$.

Definition 13 *If a linear code $\mathcal{C}$ is generated by all code sequences $\mathbf{c} \in \mathcal{C}$ of length $v + 1$ or less then $\mathcal{C}$ is referred to as v-controllable.*

The sequences that generate a code are called the *code generators.*

Definition 14 *The* j-controllable subcode *of a linear code $\mathcal{C}$ is defined as the code C_j generated by the sequences in $\mathcal{C}$ of length $j + 1$ or less:*

$$C_j = \prod C_{[k,k+j]}. \tag{4.73}$$

In general we have for a v-controllable code $\mathcal{C}$ that

$$C_0 \subseteq C_1 \subseteq \cdots \subseteq C_v = \mathcal{C} \tag{4.74}$$

which forms a normal series. There is a one-to-one correspondence with the chain coset decomposition

$$\mathcal{C} \leftrightarrow C_0 \times (C_1/C_0) \times \cdots \times (C_v/C_{v-1}). \tag{4.75}$$

Given that a linear group code may be decomposed in the manner described by equations (4.73) and (4.75) we may construct a group code by appropriately defining the subcodes C_j. We use the form of a general feed-forward convolutional encoder to define the codes C_j as described in the next section.

4.7.2 Encoder structure.

The general structure of a trellis encoder for the codes of our interest is shown in figure 4.3. The encoder consists of b binary tapped delay lines. The number of delay elements in the i^{th} line is l_i and ranges from 0 to v. The delay lines are drawn in order of increasing length from the top of the diagram downwards. Each tap of each delay line is weighted by an element g_{ij} from the group G. If a segment of the delay line is active (i.e. contains a 1) the corresponding weight contributes to the current output by addition using the group operator. We may observe that if the encoder is in the zero-state and a one is entered into the i^{th} input at time k then the contribution to the output starting at time k is the sequence

$$\{g_{i0}, g_{i1}, \ldots, g_{il_i}\} \tag{4.76}$$

These sequences are called the code *generators*. We define the set S_i to contain the generators of the encoder of length i, and denote the elements of S_i by

$$S_i = \{S_{i1}, S_{i2}, \ldots\} \tag{4.77}$$

The purpose of a code search is to select the weights $g \in G$ of the tapped delay line such that the encoder structure of figure 4.3 forms a linear group code and hence a geometrically uniform code over G. We may do so by applying the results of Forney and Trott given in section 4.7.1 to figure 4.3. We know from equation (4.74) that a group code can be decomposed into chain of subgroups. The i^{th} group in the chain

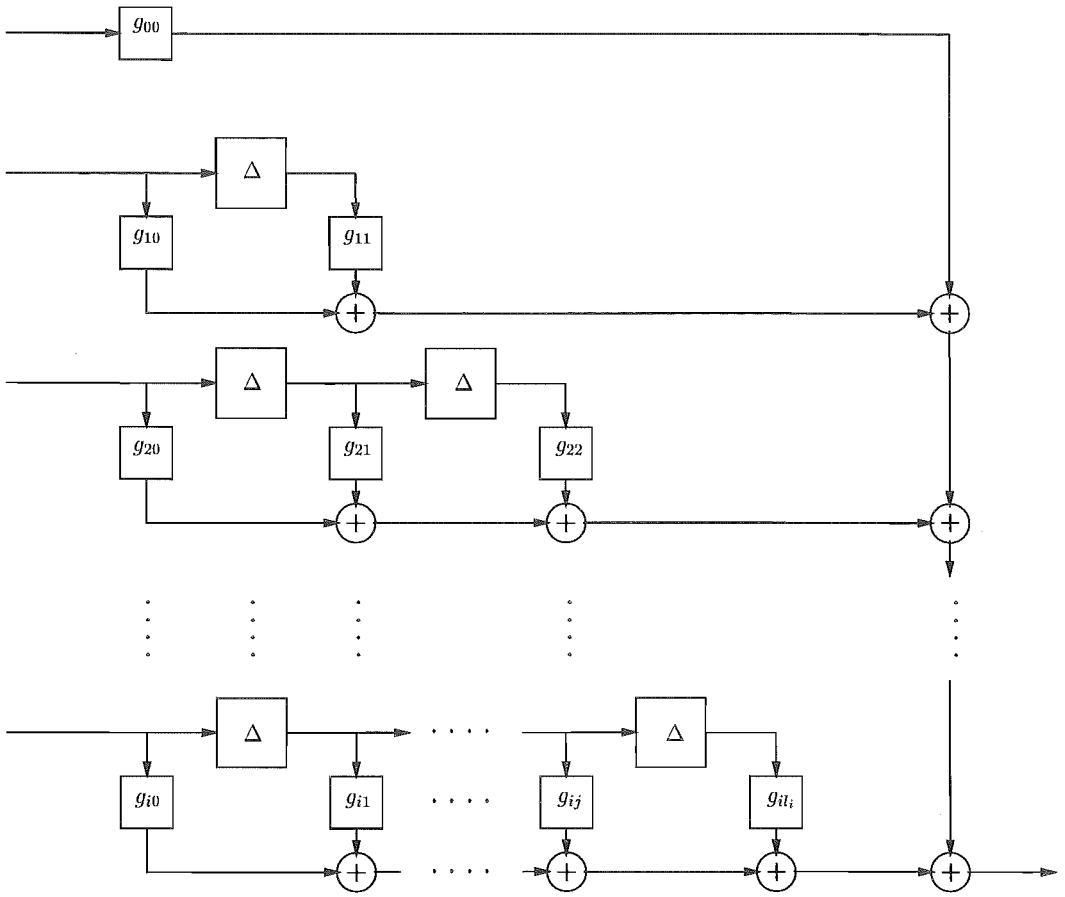


Figure 4.3: General structure of a convolutional encoder for the generation of trellis codes.

is composed of all the length i sequences in $\mathcal{C}$. From equation (4.73) it is clear that the group C_i cannot contain the elements of the set S_j for $j > i$. Therefore the subcode C_0 is generated by the elements from the set S_0 and we must select S_0 such that it forms a group. The group C_1 contains the group C_0 which is generated by the elements from S_0. Since the elements from S_2 and above cannot be in C_1 we know that C_1 must be formed from the union of C_0 and its cosets, where the coset representatives are the elements generated from S_1. Hence S_1 expands C_0 to C_1. Similarly S_2 expands C_1 to C_2 etc. up to S_v expanding C_{v-1} to $C_v = \mathcal{C}$. The code construction strategy is therefore to first select the elements of S_0 such that $\langle S_0 \rangle$ forms a subgroup of G. We then have $C_0 = \langle S_0 \rangle^L$. The next step is to select the elements of S_1 such that $C_1 = \langle C_0, S_1 \rangle$ forms a group etc. until all generators are defined.

An example trellis encoder is shown in figure 4.4. The figure is drawn using a shorthand notation of the general encoder of figure 4.3 where the generators have

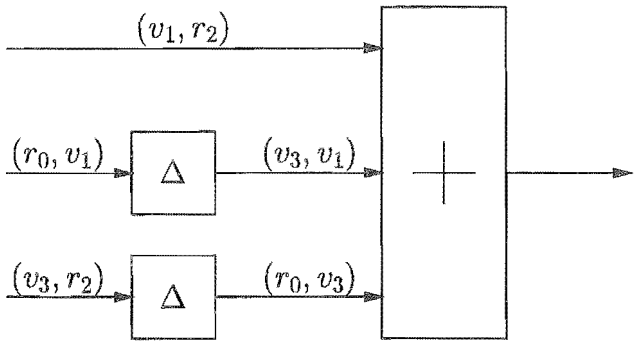


Figure 4.4: An example trellis encoder of a code transmitting 1.5 bits/symbol using $S = 2 \times 4$-PSK.

been written directly on the tapped delay lines instead of as multipliers of the lines. The generators are elements from the generating group G of 2×4-PSK isomorphic to $(D_2)^2$ and given by

$$G = \langle (r_2, r_0), (v_1, r_0), (r_0, r_2), (r_0, v_1) \rangle \tag{4.78}$$

A notation for the encoder is to describe it by the encoder matrix

$$G_c = \begin{pmatrix} (v_1, r_2) & (r_0, r_0) \\ (r_0, v_1) & (v_3, v_1) \\ (v_3, r_2) & (r_0, v_3) \end{pmatrix} \tag{4.79}$$

and a more compact notation is to write down the generators separated by semi-colons.

$$G_g = \{(v_1, r_2); (r_0, v_1), (v_3, v_1); (v_3, r_2), (r_0, v_3)\} \tag{4.80}$$

The encoder consists of three tapped delay lines, one of length zero, and two of length one. The sets S_0 and S_1 are

$$S_0 = \{\{(v_1, r_2)\}\} \tag{4.81}$$

and

$$S_1 = \{\{(r_0, v_1), (v_3, v_1)\}, \{(v_3, r_2), (r_0, v_3)\}\}. \tag{4.82}$$

From figure 4.3 we observe that the tapped delay lines of length zero do not encode the incoming bits, and these are referred to as the uncoded input bits. Of the b bits

entering the encoder, the number of uncoded bits is denoted b_u and the remaining b_c bits are the coded bits. The number of uncoded bits determines the number of parallel branches in the trellis diagram of the encoder. The set of parallel branches forms a group called the parallel transition group P of the code. The parallel group is related to S_0 by

$$P = \langle S_0 \rangle \tag{4.83}$$

The order of P is

$$|P| = 2^{b_u} \tag{4.84}$$

In general if the encoder of figure 4.3 is constructed over an L-dimensional constellation and b input bits are accepted per trellis interval then the spectral efficiency r of the code in bits per two-dimensional symbol is

$$r = \frac{b}{L}. \tag{4.85}$$

The number of delay elements v in the code is given by

$$v = \sum_{i=1}^{b} l_i \tag{4.86}$$

4.7.3 Decoding complexity.

The most efficient method of maximum likelihood decoding of trellis codes is the Viterbi algorithm [38] described in section 3.5.1 of chapter 3. The core of the algorithm consists of computing the minimum cumulative metric entering each state. This is done by extending the branch metrics of all branches entering a state and retaining the minimum. If the trellis has parallel branches then the branch metric is the minimum metric from the parallel set to the received vector. The *normalised decoding complexity* has been defined by Wei [77] and others [73] as the number of distinct trellis transitions to be dealt with by the decoder per two dimensional symbol received. If v is the constraint length of the encoder, b_c is the number of coded bits and L is the number of two dimensional symbols per branch then the normalised trellis complexity of a code $\mathcal{C}$ is defined by

$$\xi_T(\mathcal{C}) = \frac{2^{v+b_c}}{L}. \tag{4.87}$$

We will refer to $\xi_T(\mathcal{C})$ as the normalised *trellis complexity* or simply the trellis complexity of the code $\mathcal{C}$. This definition ignores parallel branches in the trellis which are not very significant when their numbers are small (e.g. two or four). However geometrically uniform trellis codes over multidimensional constellations can often have large parallel groups and these need to be accounted for when considering the decoding complexity. The number of cosets of the parallel group in the code is

$$N_p = \min(|G|/|P|, 2^{v+b_c}) \tag{4.88}$$

where G is the generating group, and we define the normalised parallel group decoding complexity by

$$\xi_P(\mathcal{C}) = \frac{N_p \xi(P)}{L} \tag{4.89}$$

where N_p is the number of cosets of P in the code, and $\xi(P)$ is the decoding complexity of P. We define the modified normalised decoding complexity of a trellis code by

$$\xi(\mathcal{C}) = \xi_T(\mathcal{C}) + \xi_P(\mathcal{C}) \tag{4.90}$$

i.e. the sum of the trellis complexity and the parallel group complexity.

4.7.4 Computation of $\xi(\mathbb{P})$.

To decode the parallel group P, we need to compute the metric between each element in P and the received vector $\mathbf{r}$. However since P is a subgroup of G it forms a linear subcode and is written as a trellis. In general this trellis is time-varying and starts and ends in the zero state. If at time k the trellis has T_k states and B_k branches then the decoding complexity of the trellis is

$$\xi(P) = \sum_k B_k T_k \tag{4.91}$$

For example if P is the subset of points from a 6×4-PSK constellation generated by the nine generators

$$\begin{aligned} P = \; & \langle (r_2, r_2, r_2, r_2, r_0, r_0), \\ & (r_0, r_2, r_0, r_2, r_0, r_0), \\ & (r_1, r_1, r_1, r_1, r_0, r_0), \\ & (r_2, r_1, r_2, r_1, r_0, r_0), \\ & (r_0, r_0, r_0, r_0, r_2, r_2), \\ & (r_0, r_0, r_0, r_0, r_0, r_2), \\ & (r_0, r_0, r_0, r_0, r_1, r_1), \\ & (r_0, r_0, r_0, r_0, r_2, r_1), \\ & (r_0, r_0, r_2, r_2, r_2, r_2) \rangle \end{aligned} \tag{4.92}$$

then there are a total of 512 points. Clearly an exhaustive comparison to find the minimum distance element is computationally expensive. Alternatively we may construct an equivalent trellis representation of P. To do so we need to find the minimal representation as described in the paper by Forney and Trott [42]. For the example set of generators of equation (4.92), the minimal representation is described by the set of generators given by

$$\begin{aligned} P = \; & \langle (r_2, r_2, r_2, r_2, r_0, r_0), \\ & (r_2, r_2, r_0, r_0, r_0, r_0), \\ & (r_1, r_0, r_1, r_0, r_0, r_0), \\ & (r_0, r_1, r_2, r_0, r_0, r_0), \\ & (r_0, r_2, r_1, r_2, r_0, r_0), \\ & (r_0, r_0, r_1, r_1, r_0, r_0), \\ & (r_0, r_0, r_0, r_0, r_2, r_0), \\ & (r_0, r_0, r_0, r_0, r_1, r_0), \\ & (r_0, r_0, r_0, r_0, r_0, r_2), \\ & (r_0, r_0, r_0, r_0, r_0, r_1) \rangle. \end{aligned} \tag{4.93}$$

The skeleton of the corresponding trellis diagram of the group P is shown in figure 4.5 and the minimum squared Euclidean distance to the set P may now be found by applying the VA to this trellis. The number of metric computations and comparisons now required is reduced to $\xi(P) = 4 + 16 + 16 + 4 + 4 + 4 = 48$, which is a factor of about 11 less complex than the direct implementation. A coset of P may be written

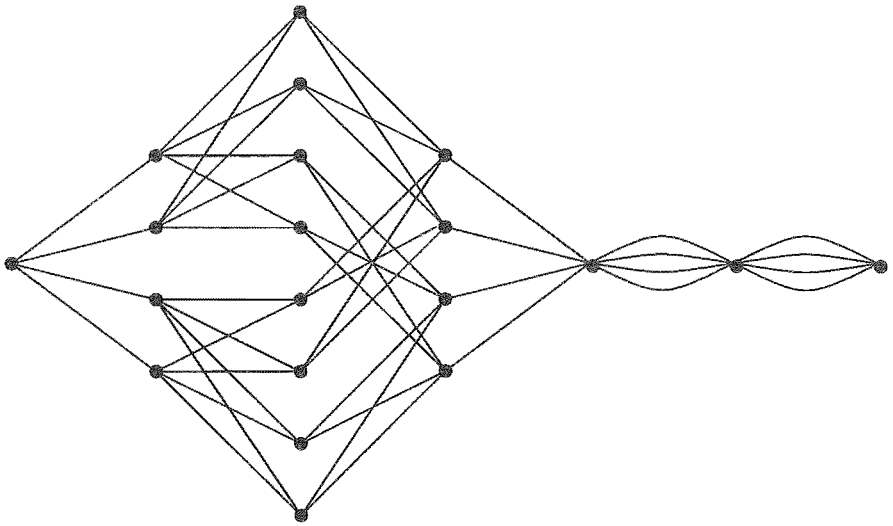


Figure 4.5: The trellis structure for the set P.

as $P' = gP$ and its trellis is identical to that of P, with each branch modified by the coset representative g, hence $\xi(P') = \xi(P)$. In general a reduction in complexity is possible only if $|P|$ is large.

The next section describes the algorithms used to construct generating groups and to search for good geometrically uniform codes for the Rayleigh fading channel.

4.8 Code construction algorithms.

We have presented how the code generators must be chosen such that the convolutional encoder of figure 4.3 generates a geometrically uniform trellis code. In this section we present algorithms that efficiently implement the code search. There are three main steps:

1. The construction of the generating groups of the underlying geometrically uniform constellation. The trellis code search is based on the generating groups.

2. The expansion of each generating group $G(S)$ into a subgroup tree. This step is a precomputation for the code search algorithm.

3. The trellis code search over the group $G(S)$.

Each of the the three construction steps relies on the construction of subgroups of a group. We first present a method of constructing subgroups of a group by the

technique of *cyclic extension*, and then discuss each of the above three steps in detail in sections 4.8.3, 4.8.2, 4.8.4 respectively.

4.8.1 Subgroup construction by cyclic extension.

A computationally efficient technique for constructing subgroups of a group is by the method of *cyclic extension* [9]. This is a general method that constructs a new subgroup by starting with a known subgroup and adding a well-chosen generator. In the code constructions we are concerned only with subgroups with orders that are powers of two, and only use expansions that double the size of the starting group. This construction is as follows:

Definition 15 Doubling construction: *If G is a group and H is a subgroup of G, then the set $H' = \{H, hH\}$ forms a group of order $2|H|$ provided h satisfies the three conditions:*

$$h \notin H \tag{4.94}$$

$$h^2 \in H \tag{4.95}$$

and

$$hH = Hh \tag{4.96}$$

From the last condition we need only consider those elements $h \in G$ that *normalise* H. The set of elements that normalises H forms a group [60].

Definition 16 *The normalising group $N(H, G)$ of a subgroup H of G is defined as*

$$N(H, G) = \{g \in G : gH = Hg\} \tag{4.97}$$

If $N(H, G) = G$ then H is a normal subgroup of G.

The doubling construction can be used recursively to construct all subgroups of a group that are an order of a power of 2. A starting subgroup always exists, namely the trivial subgroup $H = \{e\}$. The recursive construction forms a *subgroup tree* of a group G.

4.8.2 Subgroup trees.

The application of the doubling construction to the trivial group $H = \{e\}$ forms subgroups of order two. The application of the doubling construction to these groups produces subgroups of order four etc. The resulting groups may conveniently be drawn in a tree structure, where the i^{th} level of the tree contains those subgroups of order 2^i and the branches connecting the levels represent the generators expanding the group from level i to $i+1$. The algorithm of table 4.1 which is a specialisation of the cyclic extension method constructs the subgroup tree of a group G of order 2^n [9]. The algorithm works top down by expanding each subgroup at level i using the doubling construction of definition 15 and placing the resulting subgroups at level $i+1$. The set Λ contains the candidate generators for the group at level i under consideration and elements of Λ are eliminated under the following conditions:

1. The elements not in the normalising group $N(U_i, G)$.

2. If U_{ij} is a subgroup of a group $U_{(i+1)k}$ on level $i+1$ then the elements of $U_{(i+1)k}$ are eliminated from Λ thus avoiding the construction of identical groups.

The algorithm begins with $U_0 = \{\{e\}\}$ and works down to level n, ending with $U_n = \{G\}$.

An example of a binary subgroup tree of G is shown in figure 4.6, where the group $G = \{r_0, r_1, r_2, r_3, v_0, v_1, v_2, v_3\}$, the symmetries of 4-PSK. Note that the i^{th} level contains those subgroups of order 2^i and that a path from level 0 to level n defines a subgroup chain of G.

4.8.3 Construction of generating groups.

The generating groups of a constellation S are those subgroups of the symmetry group $\gamma(S)$ with the properties given by equation (4.66). To find these generating groups we may simply scan the appropriate level of the binary subgroup tree of the symmetry group. However this is inefficient because the size of a subgroup tree grows exponentially with the size of the symmetry group of the constellation. A more efficient technique is to use a modification of the binary subgroup tree generation algorithm of table 4.1 that only generates those subgroups satisfying the criteria of a generating group. Another reduction in complexity is possible because we are

Input : G, a group of order 2^n

Output : The binary subgroup tree of G, described by the set $\mathcal{U}$ of sets of subgroups U_i at level i, where 2^i is the order of the subgroups in U_i.

begin
  Set $U_0 = \{\{e\}\}$
  for $i = 0$ to n **do**
    for each subgroup U_{ij} in U_i **do**
      $\Lambda := N(U_{ij}, G) - U_{ij}$
      for each subgroup $U_{(i+1)k}$ in U_{i+1} **do**
        if $U_{ij} \subseteq U_{(i+1)k}$ **then** $\Lambda := \Lambda - U_{(i+1)k}$ **end if**
      end for
      while $|\Lambda| > 0$ **do**
        choose $g \in \Lambda$
        if $g^2 \in U_{ij}$ **then**
          $U_{i+1} := U_{i+1} + \langle U_{ij}, g \rangle$
          $\Lambda := \Lambda - \langle U_{ij}, g \rangle$
        else
          $\Lambda := \Lambda - g$
      end while
    end for
  end for
  Output $\mathcal{U} = \{U_i : i = 0, 1, \ldots, n\}$
end.

Table 4.1: Algorithm for constructing a binary subgroup tree.

interested in the generating groups only for the construction of codes. Clearly if two generating groups are such that they have the same group structure and distance properties then they will produce codes with the same distance properties and hence

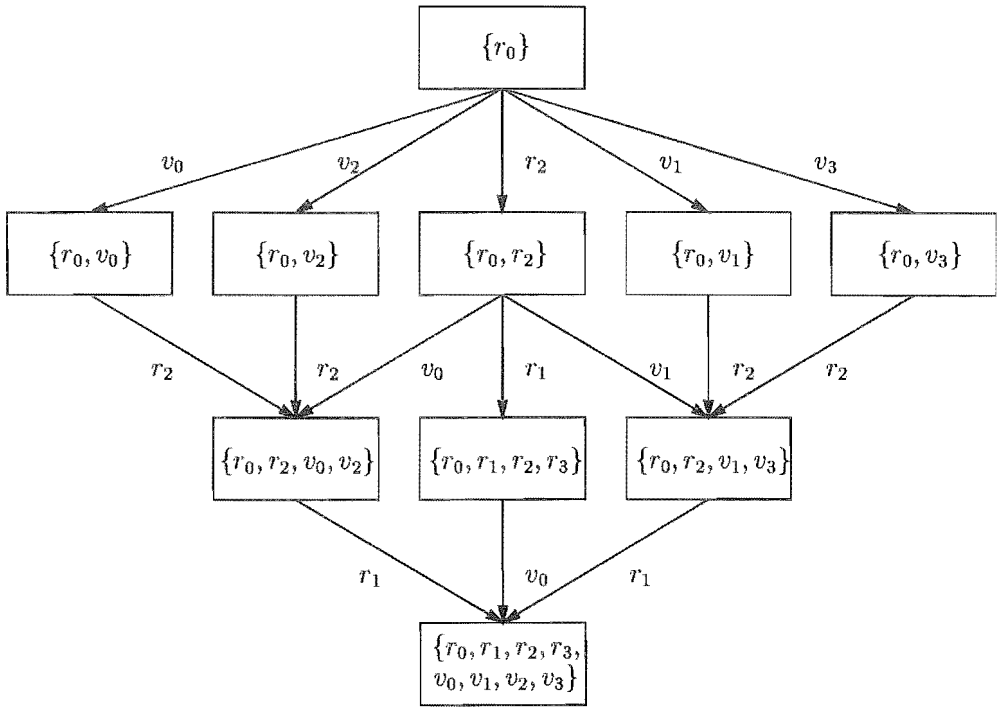


Figure 4.6: An example binary subgroup tree of the group $G = \{r_0, r_1, r_2, r_3, v_0, v_1, v_2, v_3\}$.

only one of the two generating groups needs to be considered. The following theorem identifies such groups.

Theorem 6 *Two generating groups $G_1(S)$ and $G_2(S)$ of a signal set S have 1) the same group structure and 2) the same distance structure if for some element $\gamma \in \Gamma(S)$:*

$$G_1(S) = \gamma^{-1} G_2(S) \gamma \tag{4.98}$$

Proof: 1) The mapping $\alpha(x) = \gamma^{-1} x \gamma$ is an isomorphism hence G_1 and G_2 are isomorphic and have the same group structure. 2) The element γ is a symmetry of S and hence the distance structure of G_1 is the same as that of G_2. G_1 and G_2 are said to be *congruent*.

For example consider two generating groups of the 2×4-PSK signal constellation,

$$G_1 = \langle (r_2, r_0), (r_1, r_0), (r_0, r_2), (r_0, v_1) \rangle \simeq Z_4 \times D_2 \tag{4.99}$$

and

$$G_2 = \langle (r_2, r_0), (v_1, r_0), (r_0, r_2), (r_0, r_1) \rangle \simeq D_2 \times Z_4. \tag{4.100}$$

Clearly G_1 and G_2 have the same structure and distance properties and can be considered equivalent for code generation purposes. The isomorphism is $G_1 = \gamma^{-1} G_2 \gamma$, for $\gamma = (1, 2)$, i.e. the swapping of coordinates. The algorithm we have used to construct the generating groups of a constellation is based on the subgroup tree algorithm and is a modification of that used by Benedetto *et al.* [3], to include theorem 6 such that no congruent generating groups are constructed. The algorithm is shown in table 4.2 In general the number of generating groups constructed by the algorithm of table 4.2 is too large to be practical for an exhaustive code search and the number considered in practice needs to be reduced. Typically this is done by considering only specific classes of generating groups. The classes considered for the code search results we present are discussed on a case by case basis in section 4.9.

Input : $\Gamma(S)$, the symmetry group of the signal constellation S.
Output : A set of generating groups $\mathcal{G} = \{G_1, G_2, \dots\}$.

begin
Eliminate from $\Gamma(S)$ all the symmetries different from the identity which leave unchanged at least one symbol of S. Call the resulting subset $\Gamma'(S)$.
Set $U_0 = \{\{e\}\}$
for $i = 0$ to n **do**
for each subgroup U_{ij} in U_i **do**
$\Lambda := [N(U_{ij}, \Gamma(S)) - U_{ij}] \cap \Gamma'(S)$
for each subgroup $U_{(i+1)k}$ in U_{i+1} **do**
if $U_{ij} \subseteq U_{(i+1)k}$ **then** $\Lambda := \Lambda - U_{(i+1)k}$
if $\exists\gamma\, U_{ij} \subseteq \gamma^{-1}U_{(i+1)k}\gamma$ **then** $\Lambda := \Lambda - \gamma^{-1}U_{(i+1)k}\gamma$
end for;
while $|\Lambda| > 0$ **do**
choose $g \in \Lambda$
if $g^2 \in U_{ij}$ **then**
$U_{i+1} := U_{i+1} + \langle U_{ij}, g\rangle$
$\Lambda := \Lambda - \langle U_{ij}, g\rangle$
else
$\Lambda := \Lambda - \{g\}$
end if
end while
end for
end for
Output $\mathcal{G} = U_n$
end.

Table 4.2: An algorithm to construct the generating groups of a constellation S from the symmetry $\Gamma(S)$.

4.8.4 Code Search Algorithm.

We have described the necessary properties of the code generators such that the encoder forms a group code. Before these can be applied we need to decide on the exact structure of the encoder, i.e. the number of input bits b and the lengths of the tapped delay lines l_i. The best configuration of the encoder to give codes with a high Hamming distance is to have the lengths of the delay lines as nearly equal as possible. To satisfy this constraint we have in general for a 2^v state code and b input bits, a set of tapped delay lines with the following lengths:

$$\begin{array}{ll} \text{number} & \text{length} \\ & \\ b - v \bmod b & \lfloor v/b \rfloor \\ v \bmod b & \lfloor v/b \rfloor + 1 \end{array} \tag{4.101}$$

e.g. for $b = 2$ and $v = 5$ we have one line with two delay elements and one line with three delay elements. The parameter b is selected such as to give the desired spectral efficiency, as given by equation (4.85). For example if we desire a spectral efficiency of $r = 2$ bits/symbol and we have $L = 2$, then a value of $b = 4$ is needed. The parameter v is selected based on two opposing trade-offs of coding gain versus decoding complexity. In general the decoding complexity doubles with each increase in constraint length v, while coding gain has a diminishing return. Table 4.3 shows the outline of the code search algorithm used for the generation of the codes presented in section 4.9.

4.8.5 Code evaluation criteria.

The performance of a code on the Rayleigh fading channel is a function of several parameters. The slope of the probability of error curve is determined by the Hamming distance, while the product distance d_p^2 varies the vertical positioning of the curve. In chapter 3 we showed that the probability of error of a code may be bounded by the union bound given by

$$p_{\text{err}}(C) = \sum_{\mathbf{c}_j \in C} p(\mathbf{c}_0 \rightarrow \mathbf{c}_j) \tag{4.102}$$

where $p(\mathbf{c}_0 \rightarrow \mathbf{c}_j)$ is the probability of confusing codeword $\mathbf{c}_j$ for codeword $\mathbf{c}_0$ and that this equation at high SNR is dominated by the term with the lowest Hamming

Input :$G(S)$, a generating group of S,
a list of values for $|S_j|$, $1 \leq j \leq \nu$.
Output : A set of code generators S_j, $1 \leq j \leq \nu$.

begin
$A = \{e\}$;
for $j = 1$ to ν **do**
$A = A_{[0,j-1]}A_{[1,j]}$
for $i = 1$ to $\log_2 |S_j|$ **do**
• choose an element $a \in G^j(S), a \notin A, a^2 \in A$;
$A = A + aA$
if the distance properties of subcode A are worse than the best found **goto** •
$S_j = S_j + a$
end for
end for
end.

Table 4.3: Trellis code search algorithm.

distance. To compare codes we compute the minimum Hamming distance, and the associated weighted product distance defined by equation (3.27) of chapter 3. If there is a tie we use equation (4.102) to select the better code. Equation (4.102) effectively makes a decision on the higher diversity terms, i.e. on the sequences that have a Hamming distance greater than the minimum Hamming distance from the all-zero codeword.

4.9 Results.

In this section we present codes discovered by the application of the algorithm of the section 4.8.4. We have investigated codes with spectral efficiencies of 1 bit/symbol and 2 bits/symbol. For each case we describe the generating groups searched. The code tables list the properties of the best code discovered for each encoding scheme.

The generators corresponding to the codes listed have been placed in appendix E.

4.9.1 Codes transmitting 1 bit/symbol.

We first investigate codes transmitting 1 bit/symbol. These codes are of interest as they may be mapped onto 2 bit/symbol 16-QAM codes with distance properties better than that of equivalent complexity 2 bit/symbol encoders. Table 4.4 presents the codes over the constellation $S = 1 \times$ 4-PSK with the best distance properties, for $v = 1$ to $v = 8$. These codes do not have parallel transitions and are constructed over the generating group isomorphic to D_2. Note that the Hamming distance of the codes meet the bound of equation (3.33) of chapter 3 on the maximum attainable Hamming distance for a given value of L and constraint length v, for up to $v = 7$. The minimum squared product distance is high and typically doubles with each increase in constraint length. The weighted squared product distance of the codes is on average less than the minimum squared product distance, except for $v = 1$ and $v = 2$, and implies a degree of multiplicity. As an interesting aside, in table

1 bit/symbol $1 \times$ 4-PSK							
v	d_H	d_p^2	d_{pw}^2	$\xi_P(\mathcal{C})$	$\xi_T(\mathcal{C})$	$\xi(\mathcal{C})$	plot
1	2	8	8.00	0	4	4	-
2	3	32	32.00	0	8	8	-
3	4	64	42.67	0	16	16	-
4	5	128	51.2	0	32	32	-
5	6	256	64.0	0	64	64	-
6	7	512	157.54	0	128	128	-
7	8	1024	273	0	256	256	-
8	8	8192	4096	0	512	512	-

Table 4.4: Properties of $1 \times$ 4-PSK codes transmitting 1 bit/symbol.

4.5 we have listed the properties of codes also transmitting 1 bit/symbol but using the constellation $S = 1 \times$ 8-PSK, i.e. a double of size, relative to 4-PSK. Although we may intuitively expect that an increase in size of the underlying alphabet of the code allows for a greater Hamming distance for a given constraint length, this is not

true. The maximum Hamming distance is bounded by equation (3.33) in terms of v and L, and not M. As table 4.5 shows, all of the Hamming distances achieved using 8-PSK are the same as for 4-PSK, however a significant improvement in the minimum squared product distance of the codes is attained. It is interesting to note that all of the 1×8-PSK codes have a multiplicity of one. The decoding complexity does not change from 1×4-PSK to 1×8-PSK. The next set of results are also codes

1 bit/symbol 1×8-PSK							
v	d_H	d_p^2	d_{pw}^2	$\xi_P(\mathcal{C})$	$\xi_T(\mathcal{C})$	$\xi(\mathcal{C})$	plot
1	2	13.66	13.66	0	4	4	-
2	3	54.6	54.6	0	8	8	-
3	4	186.5	186.5	0	16	16	-
4	5	637	637	0	32	32	-
5	6	2174	2174	0	64	64	fig. 4.7, curve 5
6	7	8696	8696	0	128	128	-
7	8	5094	5094	0	256	256	fig. 4.7, curve 7
8	9	20380	20380	0	512	512	fig. 4.7, curve 8

Table 4.5: Properties of 1×8-PSK codes transmitting 1 bit/symbol.

transmitting 1 bit/symbol but over the constellation $S = 2 \times 4$-PSK, i.e. now we have $L = 2$. With a change in L we can expect a change in the maximum attainable Hamming distance of the code. From table 3.5 of chapter 3 we observe that for even values of v, the maximum attainable Hamming distance improves over $L = 1$ codes by one, while remaining the same for odd values of v. The codes discovered and listed in table 4.6 indeed reflect this fact. We observe that the 2×4-PSK codes improve over 1×4-PSK codes in two ways. For even values of v the codes over 2×4-PSK have a Hamming distance 1 higher than codes over 1×4-PSK, while for odd v the Hamming distances are the same, but the squared product is improved by a factor of two. Note that the decoding complexities are the same for 1×4-PSK as they are are for 2×4-PSK for the same values of constraint length v. The codes searches were performed over all generating groups of 2×4-PSK constructed by the algorithm of table 4.2, and also over $(Z_4)^2$ and $(D_2)^2$ individually. It was found that all of the best codes are over the generating group isomorphic to $(D_2)^2$.

Although we have not listed the codes, the case of transmitting 1 bit/symbol using the constellation $S = 2 \times 8$-PSK, similar to going from 1×4-PSK to 1×8-PSK improves the squared product distance of the codes further, but not the Hamming distance.

The last set of codes are over the constellation $S = 3 \times 4$-PSK. Again an increase in L changes the maximum attainable Hamming distance of the codes as listed in table 3.5. However unlike moving from $L = 1$ to $L = 2$, moving from $L = 2$ to $L = 3$ is not necessarily an advantage for all values of v. For $v = 1$, the expected Hamming distance is three, which is better than $L = 1$ and $L = 2$. For $v = 2$ the expected Hamming distance is still three, while for $L = 2$ it is four and therefore is not an improvement. However for $v = 3$, the expected Hamming distance is six which does improve over $L = 2$. Table 4.7 lists the codes discovered over $S = 3 \times 4$-PSK. We observe that not all of the codes over 3×4-PSK meet the bound of equation (3.33). For values of $v = 1$, $v = 2$, $v = 4$ and $v = 5$ the bound is met, while for values of $v = 3$, $v = 6$ and $v = 7$ the codes are one short of the bound. Consequently the only improvement over the codes discussed so far is for the case of $v = 1$, where a Hamming distance of three is attained. Note also that the average decoding complexity for a given number of states is now greater than for $L = 1$ and $L = 2$.

In figure 4.7 we have plotted the upperbounds on the pairwise probability of error of the best codes at each constraint length. The curves have been computed using the union bound equation (3.25) of chapter 3 combined with the exact pairwise

1 bit/symbol 2×4-PSK							
v	d_H	d_p^2	d_{pw}^2	$\xi_P(\mathcal{C})$	$\xi_T(\mathcal{C})$	$\xi(\mathcal{C})$	plot
1	2	16	16.00	4	2	6	-
2	4	32	12.80	0	8	8	fig. 4.7, curve 2
3	4	128	128.00	0	16	16	-
4	6	256	36.57	0	32	32	fig. 4.7, curve 4
5	6	1024	512.00	0	64	64	-
6	8	512	64.5	0	128	128	fig. 4.7, curve 6
7	8	4096	819.2	0	256	256	-
8	9	1024	334	0	256	256	-

Table 4.6: Properties of 2×4-PSK codes transmitting 1 bit/symbol.

1 bit/symbol 3×4-PSK							
v	d_H	d_p^2	d_{pw}^2	$\xi_P(\mathcal{C})$	$\xi_T(\mathcal{C})$	$\xi(\mathcal{C})$	plot
1	3	16	5.33	5.33	1.33	6.66	fig. 4.7, curve 1
2	3	64	64.00	10.67	5.33	16	-
3	5	64	9.06	0	21.33	21.33	fig. 4.7, curve 3
4	6	256	22.8	0	42.66	42.66	-
5	6	2048	2048	0	85.3	85.3	-
6	7	256	46	0	170.7	170.7	-
7	8	1024	146	0	341.3	341.3	-
8	9	2048	188	0	682.6	682.6	-

Table 4.7: Properties of 3×4-PSK codes transmitting 1 bit/symbol.

probability of error term derived in chapter 2 assuming ideal interleaving and ideal CSI. As we have shown for optimal performance, sufficient interleaving is required such that the fading affecting the code symbols is independent from symbol to symbol. Ideal CSI is justified for the comparison of codes, as any non-ideal estimate of the channel state information leads only to a loss of a few dB in performance (except for differential detection, which is discussed in chapter 7). Clearly each increase in constraint length of the code leads to a diminishing increase in coding gain. The amount of coding gain is a function of the the BER considered.

4.9.2 Codes transmitting 2 bits/symbol.

Now we examine the codes found transmitting at the most conventional rate of 2 bits/symbol. The increase in spectral efficiency implies a lower attainable Hamming distance of the codes. First we examine the conventional form of trellis codes, transmitting one symbol per branch. Table 4.8 lists the properties of the codes discovered. All of the codes meet the bound of equation (3.33) on the maximum attainable Hamming distance for $L = 2$ and a rate of 2 bits/symbol. Similar code searches over 1×8-PSK have been performed by Schlegel and Costello [62] and Dingman [18], which attain similar distance properties as those of table 4.8. By expanding the signal constellation to $S = 1 \times 16$-PSK we may improve upon the squared product distance similarly to the case of transmitting 1 bit per symbol using

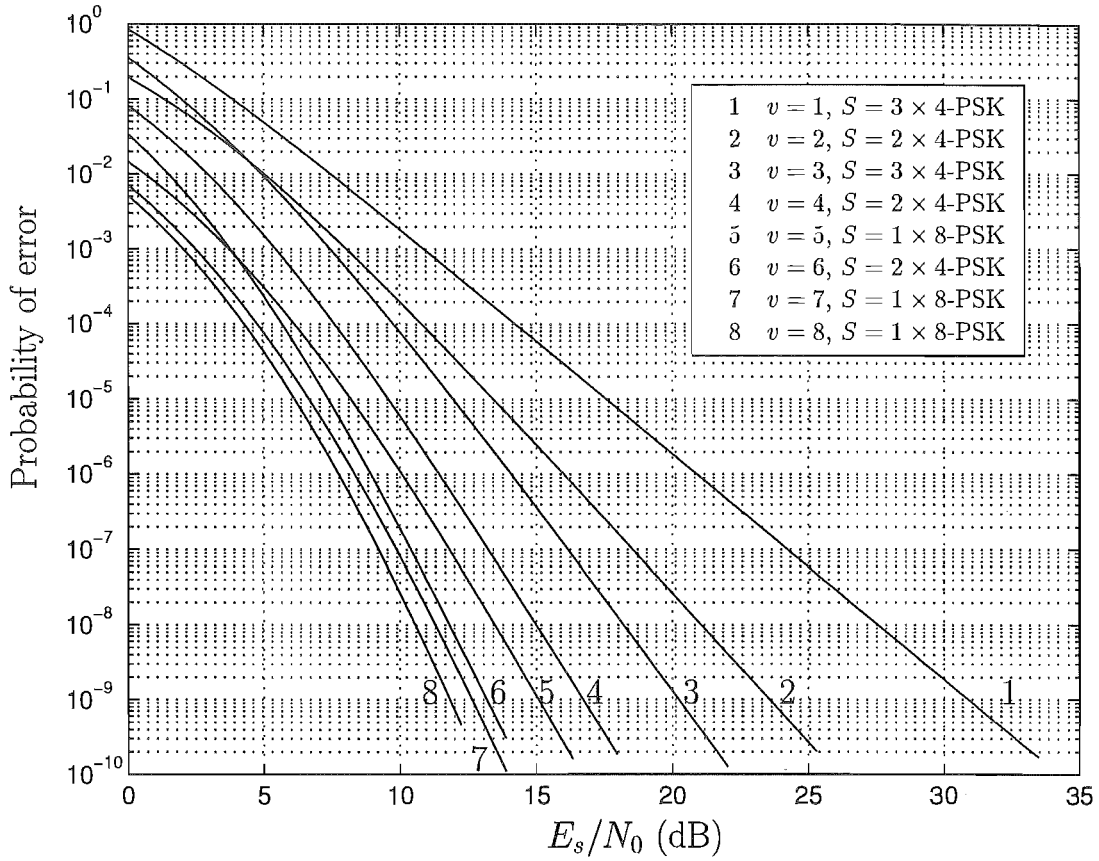


Figure 4.7: Upperbounds on geometrically uniform trellis codes transmitting 1 bit/symbol over the Rayleigh fading channel.

2 bits/symbol 1×8-PSK							
v	d_H	d_p^2	d_{pw}^2	$\xi_P(\mathcal{C})$	$\xi_T(\mathcal{C})$	$\xi(\mathcal{C})$	plot
1	1	4	4	8	4	12	-
2	2	2	0.67	0	16	16	-
3	2	13.66	13.66	0	32	32	-
4	3	4.686	1.12	0	0	64	-
5	3	9.373	5.73	0	128	128	-
6	4	5.49	1.39	0	256	256	-
7	4	37.5	23.4	0	512	512	-
8	5	13.67	3.43	0	1024	1024	-

Table 4.8: 1×8-PSK codes transmitting 2 bit/symbol.

8-PSK. These codes are listed in table 4.9. Typically the improvement in product distance is about at least a factor of two. By transmitting more than one symbol

2 bits/symbol 1×16-PSK							
v	d_H	d_p^2	d_{pw}^2	$\xi_P(\mathcal{C})$	$\xi_T(\mathcal{C})$	$\xi(\mathcal{C})$	plot
1	1	4	4	8	4	12	-
2	2	3.41	1.14	0	16	16	-
3	2	15.4	15.4	0	32	32	-
4	3	9.442	2.98	0	64	64	-
5	3	61.6	61.6	0	128	128	-
6	4	16.9	6.73	0	256	256	-
7	4	170.2	170.2	0	512	512	-
8	5	38.97	10.13	0	512	512	-

Table 4.9: 1×16-PSK codes transmitting 2 bit/symbol.

per trellis branch we can take advantage of the higher expected Hamming distance predicted by equation (3.33). Table 4.10 lists the code properties of codes over 2×8-PSK. The generating group considered is $(D_4)^2$. By using 2×8-PSK we have improved upon the diversities of codes over 1×8-PSK for the constraint lengths of $v = 1$, and $v = 5$. The $v = 1$ code of table 4.10 is equivalent to the code described by Divsalar and Simon [23]. For all cases where the 2×8-PSK codes have the same Hamming distance as the 1×8-PSK codes, they improve upon the squared product distance. It must be noted that the decoding complexity of the 2×8-PSK codes for the same constraint length is twice (for $v \geq 3$) as high as for codes over 1×8 or 1×16, unlike the case of 1×4 and 2×4 transmitting one bit per symbol which have equal complexity. This means that for a given decoding complexity the codes over 1×8 are in fact better than the codes over 2×8-PSK. Table 4.11 lists the properties of codes over 2×16-PSK. These codes all improve upon the squared product distance compared to the codes over 2×8-PSK. Also the code with $v = 4$ now reaches the maximum possible Hamming distance for $L = 2$, and $v = 4$, unlike the codes over 2×8-PSK. The decoding complexity is the same as for codes over 2×8-PSK, and therefore an expansion to 16-PSK represents a good improvement. In figure 4.8 we have plotted the upperbounds on the probability of error of the

2 bits/symbol 2 × 8-PSK							
v	d_H	d_p^2	d_{pw}^2	$\xi_P(\mathcal{C})$	$\xi_T(\mathcal{C})$	$\xi(\mathcal{C})$	plot
1	2	2	0.28	16	2	18	-
2	2	4	1.78	32	8	40	
3	2	16	16.00	32	32	64	fig. 4.8, curve 3
4	3	8	1.59	0	128	128	-
5	4	0.687	0.197	0	256	256	-
6	4	8.00	1.43	0	512	512	-
7	4	159.2	61.7	0	1024	1024	-
8	5	16.0	3.5	0	2048	2048	-

Table 4.10: 2 × 8-PSK codes transmitting 2 bit/symbol.

2 bits/symbol 2 × 16-PSK							
v	d_H	d_p^2	d_{pw}^2	$\xi_P(\mathcal{C})$	$\xi_T(\mathcal{C})$	$\xi(\mathcal{C})$	plot
1	2	2	0.39	16	2	18	fig. 4.8, curve 1
2	2	4	1.77	32	8	40	fig. 4.8, curve 2
3	2	16	16.00	64	32	96	-
4	4	0.893	0.17	0	128	128	fig. 4.8, curve 4
5	4	2	0.457	0	256	256	fig. 4.8, curve 5
6	4	24.4	10.6	0	512	512	fig. 4.8, curve 6
7	4	246.3	246.3	0	1024	1024	fig. 4.8, curve 7
8	5	37.5	13.8	0	2048	2048	fig. 4.8, curve 8

Table 4.11: 2 × 16-PSK codes transmitting 2 bits/symbol.

best codes transmitting 2 bits/symbol for each values of v. These curves have been computed in the same manner as for the codes transmitting 1 bit per symbol and assume ideal interleaving. Good coding gains are achieved. These codes improve upon the codes of [8] and [18].

4.10 Summary.

For the additive white Gaussian noise channel, the measure determining decoding decisions and error performance is Euclidean distance - the measure over which

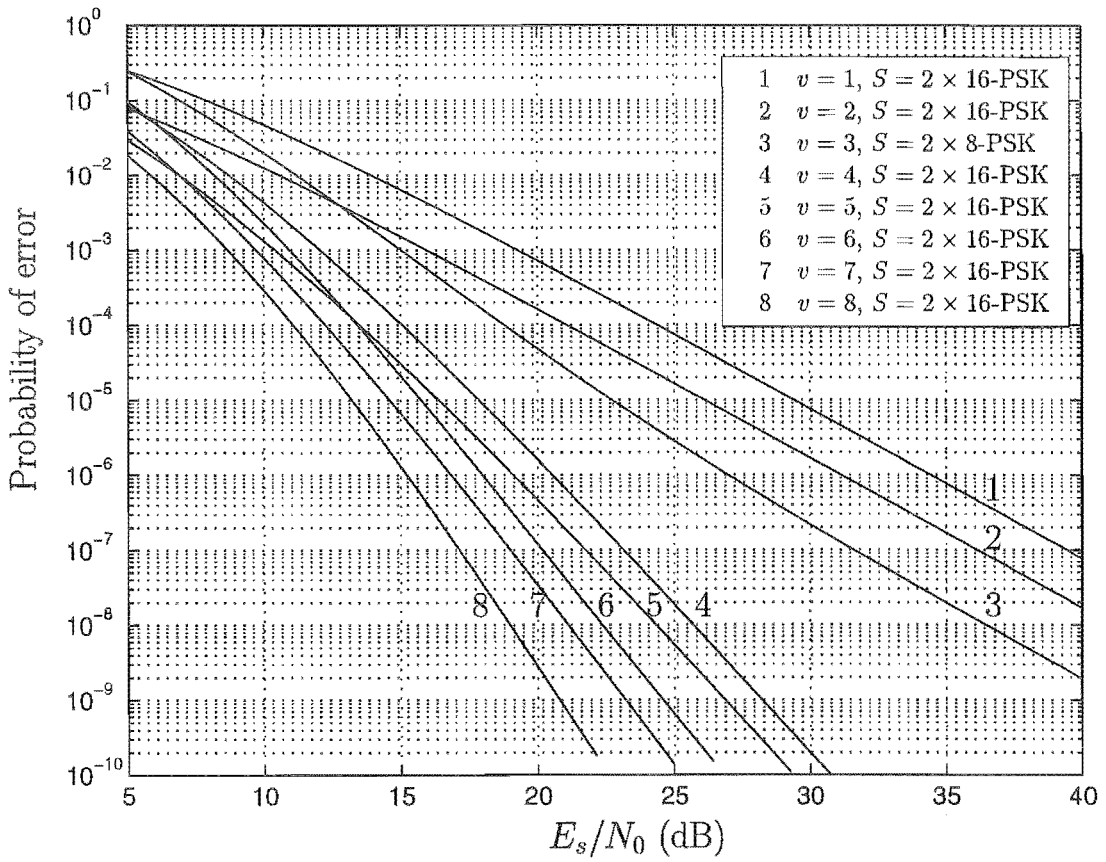


Figure 4.8: Upperbounds on geometrically uniform trellis codes transmitting 2 bits/symbol over the Rayleigh fading channel.

geometric uniformity has been defined by Forney. For other channels, such as the Rayleigh fading channel, the performance measure is no longer Euclidean distance, but rather Hamming distance and product distance. In this chapter we have generalised the definition of geometric uniformity such that the signal sets considered are geometrically uniform relative to a set of distance measures. With such a definition, codes can be constructed such that they have the desirable property of uniform error probability on the Rayleigh fading channel.

We have shown that relative to the set of distance measures $\mathcal{D}_{\mathcal{R}}$ important for the Rayleigh fading channel, the symmetries of $L \times M$-PSK constellations are the same as those for the Gaussian channel. With these symmetries we have constructed some good geometrically uniform codes over $L \times M$-PSK constellations for the Rayleigh fading channel which improve upon codes transmitting one symbol per branch published in the current literature [8],[52],[62],[82]. We have found that expanding the

signal set by more than a factor of two over the size required for uncoded modulation leads to codes that have significantly better fading channel distance properties.

Chapter 5

Geometrically uniform partitions.

In chapter 4 we applied the theory of geometrically uniform codes to design good GU trellis codes for the Rayleigh fading channel. The practical use of these codes is limited by the increase in complexity of maximum likelihood decoding with the increasing number of code states. One alternative is to use the combination of a suboptimal decoder that has less complexity than a maximum likelihood encoder, and a high complexity code. With a careful design it is possible to perform better (for the same level of complexity) with this combination than a ML decoded code. The technique of *multi-level coding* [10],[35],[44],[57],[65],[71] is such an alternative. In a multi-level coding scheme, the underlying signal constellation S_l is partitioned into an l-level hierarchy or tree of subsets. A set of component codes $C_1, C_2, \ldots, C_l$ is defined, one on each level such that the code at level l selects a subset of points S_{l-1} from the underlying constellation S_l. The encoder at level $l-1$ selects a subset from S_{l-1} etc. until the encoder at level one selects only a point to be transmitted. The overall code is not decoded as a whole, as would be done in a ML decoding scheme, but instead it is decoded in the same manner as it was constructed - in a hierarchical manner. The top level decoder D_l corresponding to code C_l decides from which subset the received point was transmitted, outputs the corresponding data bits, and passes the decision information to the next decoder D_{l-1}. The decoder D_{l-1} does likewise, and the process is repeated until the codeword is entirely decoded. Effectively each decoder is concerned only with selecting which sequence of *sets of points* was transmitted by the corresponding encoder. The overall error rate of the system will be dominated by the worst component code in the hierarchy.

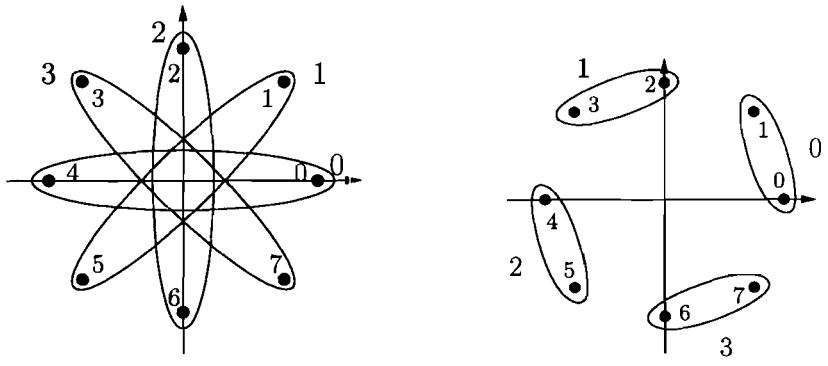
In this chapter we lay the theoretical groundwork for the design of good multi-level codes based on the extension of the definition of geometric uniformity. We extend the concept of geometric uniformity to a set of subsets of points of a constellation, such that the distance properties seen from any subset are the same as seen from any other set. With such a definition each level of the multi-level code has all of the desirable properties of a GU code and as such the search strategy and the evaluation of the error performance of each component code may be approached in the same manner as that of a GU code. Chapter 6 looks at specific multi-level codes based on GU partitions designed for the Rayleigh fading channel.

5.1 Signal Set Partitions.

Consider a constellation S, not necessarily GU, with the symmetry group $\Gamma(S)$ under the set of distance measures $\mathcal{D}$, where $\Gamma(S)$ and $\mathcal{D}$ are as defined in chapter 4.

Definition 17 *A uniform* partition S_P *of* S *is a set of disjoint subsets of* S *written as* $S_P = \{S_0, S_1, \ldots, S_{n-1}\}$ *such that for all* i, $|S_i| = |S_0|$. *The order of* S_P *is* $|S_P| = |S|/|S_0|$.

For example, figure 5.1 shows two different partitions of an 8-PSK constellation. The partition of figure 5.1(a) is written as $S_{P_1} = \{\{0,4\},\{1,5\},\{2,6\},\{3,7\}\}$ and



(a) Partition S_{P_1} (b) Partition S_{P_2}

Figure 5.1: Two four-way partitions of an 8-PSK constellation.

the partition of figure 5.1(b) is written as $S_{P_2} = \{\{0,1\},\{2,3\},\{4,5\},\{6,7\}\}$. Both

partitions are four-way and we have $|S_{P_1}| = |S_{P_2}| = 4$. By inspection, the partitions exhibit symmetry.

Definition 18 *We define a* symmetry u *of* S_P *as a symmetry of* S *that maps* S_P *onto itself, i.e.* $u(S_P) = S_P$. *We write the set of all symmetries of* S_P *as* $U(S_P)$.

For example the set of symmetries of S_{P_1} of figure 5.1(a) are

$$U(S_{P_1}) = \{r_0, r_1, r_2, r_3, r_4, r_5, r_6, r_7, v_0, v_1, v_2, v_3, v_4, v_5, v_6, v_7\} \tag{5.1}$$

where r_i and v_i represent rotation and reflection mappings as introduced in section 4.3 of chapter 4, and the symmetries of S_{P_2} are

$$U(S_{P_2}) = \{r_0, r_2, r_4, r_6, v_1, v_3, v_5, v_7\}. \tag{5.2}$$

Note that the mappings such as r_1, r_3, etc. are not symmetries of S_{P_2} because they do not map subsets onto subsets. In general, more than one symmetry may exist that generates the same mapping from S_P to S_P. For example the symmetries r_1 and r_5 of S_{P_1} both map subsets 0 to 1, 1 to 2, 2 to 3, and 3 to 0 and therefore they are equivalent.

Definition 19 *An equivalence relation* $\equiv$ *on* $U(S_P)$ *is*

$$\equiv \;=\; \{(u, u') |\, u(S_P) = u'(S_P)\} \tag{5.3}$$

i.e. two symmetries are equivalent if they both generate the same mapping on S_P.

From elementary group theory [60], each symmetry u has an associated equivalence class $[u]$, defined as the set of elements in $U(S_P)$ that are equivalent under the equivalence relation defined by equation (5.3). For example the set $[r_1, r_5]$ forms an equivalence class of $U(S_{P_1})$ since r_1 and r_5 are equivalent symmetries. The symmetry classes of $U(S_P)$ form a group [60].

Definition 20 *The symmetry group* $\Gamma(S_P)$ *of* S_P *is the group formed by the set of all equivalence classes of* $U(S_P)$.

For example the symmetry group of S_{P_1} of figure 5.1(a) is

$$\Gamma(S_{P_1}) = \{[r_0, r_4], [r_1, r_5], [r_2, r_6], [r_3, r_7], [v_0, v_4], [v_1, v_5], [v_2, v_6], [v_3, v_7]\} \tag{5.4}$$

The operation table of $\Gamma(S_{P_1})$ is shown in table 5.1. By inspection $\Gamma(S_{P_1})$ is isomorphic to the dihedral group D_4. The symmetry group of S_{P_2} is

$$\Gamma(S_{P_2}) = \{[r_0], [r_2], [r_4], [r_6], [v_1], [v_3], [v_5], [v_7]\} \tag{5.5}$$

and is also isomorphic to D_4.

$\bullet$	$[r_0, r_4]$	$[r_1, r_5]$	$[r_2, r_6]$	$[r_3, r_7]$	$[v_0, v_4]$	$[v_1, v_5]$	$[v_2, v_6]$	$[v_3, v_7]$
$[r_0, r_4]$	$[r_0, r_4]$	$[r_1, r_5]$	$[r_2, r_6]$	$[r_3, r_7]$	$[v_0, v_4]$	$[v_1, v_5]$	$[v_2, v_6]$	$[v_3, v_7]$
$[r_1, r_5]$	$[r_1, r_5]$	$[r_2, r_6]$	$[r_3, r_7]$	$[r_0, r_4]$	$[v_3, v_7]$	$[v_0, v_4]$	$[v_1, v_5]$	$[v_2, v_6]$
$[r_2, r_6]$	$[r_2, r_6]$	$[r_3, r_7]$	$[r_0, r_4]$	$[r_1, r_5]$	$[v_2, v_6]$	$[v_3, v_7]$	$[v_0, v_4]$	$[v_1, v_5]$
$[r_3, r_7]$	$[r_3, r_7]$	$[r_0, r_4]$	$[r_1, r_5]$	$[r_2, r_6]$	$[v_1, v_5]$	$[v_2, v_6]$	$[v_3, v_7]$	$[v_0, v_4]$
$[v_0, v_4]$	$[v_0, v_1]$	$[v_1, v_5]$	$[v_2, v_6]$	$[v_3, v_7]$	$[r_0, r_4]$	$[r_1, r_5]$	$[r_2, r_6]$	$[r_3, r_7]$
$[v_1, v_5]$	$[v_1, v_5]$	$[v_2, v_6]$	$[v_3, v_7]$	$[v_0, v_4]$	$[r_3, r_7]$	$[r_0, r_4]$	$[r_1, r_5]$	$[r_2, r_6]$
$[v_2, v_6]$	$[v_2, v_6]$	$[v_3, v_7]$	$[v_0, v_4]$	$[v_1, v_5]$	$[r_2, r_6]$	$[r_3, r_7]$	$[r_0, r_4]$	$[r_1, r_5]$
$[v_3, v_7]$	$[v_3, v_7]$	$[v_0, v_4]$	$[v_1, v_5]$	$[v_2, v_6]$	$[r_1, r_5]$	$[r_2, r_6]$	$[r_3, r_7]$	$[r_0, r_4]$

Table 5.1: Operation table of the group formed by the equivalence classes of the symmetries of S_{P_1}.

5.1.1 Distance measures to partitions.

Section 4.1 defined a general distance measure $D(\mathbf{x}, \mathbf{y})$ between two points in Euclidean space R^N. We generalise this definition further. Given a set of points S in R^N we define a distance measure from a point $\mathbf{x}$ to the set of points S.

Definition 21 *A distance measure $D(\mathbf{x}, S)$ between a point $\mathbf{x} \in R^N$ and a set of points S, where $|S| = n$, is defined by*

$$D(\mathbf{x}, S) = f(D_1(\mathbf{x}, s_1), D_2(\mathbf{x}, s_2), \ldots, D_n(\mathbf{x}, s_n)) \tag{5.6}$$

where each $D_i(\mathbf{x}, s_i)$ is a distance measure between the point $\mathbf{x}$ and the point $s_i \in S$, and $f(\cdot)$ is an arbitrary function.

For example the distance measure

$$f(\mathbf{x}, S) = \sum_{i=1}^{n} e^{-||\mathbf{x}-s_i||^2/2N_0} \tag{5.7}$$

is used for the soft-decision decoding of transmitted partitions as discussed in section 5.6.1. The distance measures $D_i(\mathbf{x}, s_i)$ for the example of equation (5.7) are therefore given by

$$D_i(\mathbf{x}, s_i) = ||\mathbf{x} - s_i||^2 \tag{5.8}$$

Theorem 7 *Consider a distance measure $D(\mathbf{x}, S)$ as defined by equation (5.6). If the mapping u is distance invariant under the set of distance measures $\mathcal{D} = \{D_1, D_2, \ldots, D_n\}$ then the measure $D(\mathbf{x}, S)$ is distance invariant under u also.*

Proof: For $D(\mathbf{x}, S)$ to be distance invariant under u we need to show that $D(u(\mathbf{x}), u(S))$ $= D(\mathbf{x}, S)$.

$$\begin{aligned} D(u(\mathbf{x}), u(S)) &= f(D_1(u(\mathbf{x}), u(s_1)), D_2(u(\mathbf{x}), u(s_2)), \ldots, D_n(u(\mathbf{x}), u(s_n))) \\ &= f(D_1(\mathbf{x}, s_1), D_2(\mathbf{x}, s_2), \ldots, D_n(\mathbf{x}, s_n)) \\ &= D(\mathbf{x}, S) \end{aligned} \tag{5.9}$$

With this theorem if the set of distance measures $\mathcal{D} = \{D_1, D_2, \ldots, D_n\}$ are invariant under u then all distance measures composed as a function of $\mathcal{D}$ are invariant under u also.

5.2 Geometric uniformity of partitions.

We now define the conditions for the geometric uniformity of a partition S_P, of S.

Definition 22 *A partition S_P of S is geometrically uniform relative to the set of distance measures $\mathcal{D}$ if, for any two sets $S_i \in S_P$ and $S_j \in S_P$, there exists a symmetry $u_{S_i,S_j} \in \Gamma(S_P)$ that maps S_i to S_j, while leaving S_P invariant.*

$$u_{S_i,S_j}(S_i) = S_j, \tag{5.10}$$

$$u_{S_i,S_j}(S_P) = S_P \tag{5.11}$$

The two example partitions of figure 5.1 are both geometrically uniform. Note that the definition of geometric uniformity of signal sets as defined by Forney [41] and again described in section 4.4 of chapter 4, is a special case of geometric uniformity of signal set partitions, i.e. the case where each subset S_i contains only one point of S. The definition of geometric uniformity of partitions is therefore a generalisation of geometric uniformity of signal sets.

5.2.1 Geometric properties of GU partitions.

Analogous to a point in a GU signal constellation S, a GU partition S_P has the property of looking the same when viewed from any of its elements. We expand the definitions by Forney [41] of the geometric properties of GU constellations to geometric properties of GU partitions.

Decision regions.

In a conventional communications system, a soft-decision decoder has associated with it a decoding measure $D(\mathbf{r}, s)$. Given a received vector $\mathbf{r}$, the decoder selects the point s from the transmission signal constellation S, such that $D(\mathbf{r}, s)$ is minimised. The *decision region* $R_V(s)$ of a point s is that region in R^N such that if $\mathbf{r}$ falls in $R_V(s)$, then s is selected as the output of the decoder. To define a decision region of a subset S_i of a partition S_P, we define the decoding measure for S_i to be $D(\mathbf{x}, S_i)$ as defined by equation (5.6). The decision region $R_V(S_i)$ of a subset $S_i \in S_P$ is the region in R^N such that if $\mathbf{r}$ falls in $R_V(S_i)$, then S_i is selected as the output of the decoder. Formally:

Definition 23 *The decision region, $R_V(S_i)$ associated with a subset $S_i \in S_P$ is the set of all points in R^N that such that $D(\mathbf{x}, S_i)$ is a minimum over $D(\mathbf{x}, S_j)$, $S_j \in S_P$:*

$$R_V(S_i) = \{\mathbf{x} \in R^N : D(\mathbf{x}, S_i) = \min_{S_j \in S_P} D(\mathbf{x}, S_j)\} \tag{5.12}$$

A decision region is also known as a Voronoi region for $D(\mathbf{x}, S_i) = d^2$.

Theorem 8 *If the partition S_P of S is geometrically uniform under a set of distance measures $\mathcal{D} = \{D_1, D_2, \ldots, D_n\}$ and the decision measure $D(\mathbf{x}, S_i)$ is a function of the set $\mathcal{D}$ only, then all Voronoi regions $R_V(S_i)$ have the same shape.*

Proof: By definition, $\mathbf{x} \in R^N$ is in $R_V(S_i)$ if and only if

$$D(\mathbf{x}, S_i) = \min_{S_j \in S} D(\mathbf{x}, S_j)$$

If $S_j \neq S_i$ is a subset in S_P, then by the definition of geometrical uniformity there exists a symmetry u such that $u(S_i) = S_j$. If u is such a symmetry then $u(\mathbf{x})$ is in

$R_V(S_j)$ because

$$\begin{aligned} D(u(\mathbf{x}), S_j) &= D(u(\mathbf{x}), u(S_i)) \\ &= D(\mathbf{x}, S_i) \\ &= \min_{S_k \in S_P} D(\mathbf{x}, S_k) \\ &= \min_{S_k \in S_P} D(u(\mathbf{x}), u(S_k)) \end{aligned} \tag{5.13}$$

where as S_k ranges through S_P, so does $u(S_k)$. The fact that the shape of every Voronoi region of S_P is the same implies the error performance can be computed by considering only *one* subset of the partition.

Distance profiles.

Given a partitioning S_P of a signal constellation S we define the following distance profiles for a distance measure $D \in \mathcal{D}$:

Definition 24 *The distance profile of a subset S_i to a point s is the set of distances between every point in S_i and s:*

$$DP(S_i, s) = \{D(s_i, s) : s_i \in S_i\} \tag{5.14}$$

Definition 25 *The distance profile between two subsets S_i and S_j is the set of distance profiles between S_i and every point in S_j:*

$$DP(S_i, S_j) = \{DP(S_i, s_j) : s_j \in S_j\} \tag{5.15}$$

Theorem 9 *If a symmetry u exists such that $u = u^{-1}$ and u maps S_i onto S_j then $DP(S_i, S_j) = DP(S_j, S_i)$.*

Proof: From definition 25:

$$\begin{aligned} DP(S_i, S_j) &= \{DP(S_i, s) : s \in S_j\} \\ &= \{\{D(s_i, s_j) : s_i \in S_i\} : s_j \in S_j\} \\ &= \{\{D(u(s_i), u^{-1}(s_j)) : s_i \in S_i\} : s_j \in S_j\} \\ &= \{\{D(s', s'') : s' \in S_j\} : s'' \in S_i\} \\ &= DP(S_j, S_i) \end{aligned} \tag{5.16}$$

Definition 26 *The global distance profile associated with any subset $S_i \in S_P$ is the set of all distance profiles of S_i to all subsets in S_P:*

$$DP(S_i, S_P) = \{DP(S_i, S_j) : S_j \in S_P\} \tag{5.17}$$

Definition 27 *The* minimum distance $D_{\min}(S_i, S_j)$ *from subset $S_i \in S_P$ to subset $S_j \in S_P$ is the minimum of the distances between any point in S_i and any point in S_j:*

$$D_{\min}(S_i, S_j) = \min_{s_i \in S_i, s_j \in S_j} D(s_i, s_j) \tag{5.18}$$

In general $D_{\min}(S_i, S_j) = D_{\min}(S_j, S_i)$, since $D(\mathbf{x}, \mathbf{y}) = D(\mathbf{y}, \mathbf{x})$.

Definition 28 *The* multiplicity $N(S_i, S_j)$ *between a subset S_i and subset S_j is the maximum number of points in S_j at the minimum distance from a point in S_i.*

In general $N(S_i, S_j) \neq N(S_j, S_i)$.

Theorem 10 *If the partitioning S_P of S is geometrically uniform under a distance measure D then the global distance profile $DP(S_i, S_P)$ is the same for all $S_i \in S_P$.*

Proof: From the definition of the global distance profile for the distance measure D:

$$\begin{aligned} DP(S_i, S_p) &= \{\{\{D(s', s) : s' \in S_i\} : s \in S_j\} : S_j \in S_p\} \\ &= \{\{\{D(u(s'), u(s) : s' \in S_i\} : s \in S_j\} : S_j \in S_p\} \\ &= \{\{\{D(s'', u(s)) : s'' \in s_i'\} : u(s) \in u(S_j)\} : u(S_j) \in S_p\} \\ &= DP(S_i', S_p) \end{aligned} \tag{5.19}$$

where $u(s)$ ranges through $u(S_j)$ as s ranges through S_j, and $u(S_j)$ ranges through S_P as S_j ranges through S_P.

Mappings of GU partitions.

Consider a geometrically uniform partition $S_P = \{S_0, S_1, \ldots, S_{n-1}\}$ with symmetry group $\Gamma(S_P)$, under the set of distance measures $\mathcal{D}$.

Theorem 11 *If v is a distance invariant mapping under $\mathcal{D}$ then $S_P' = vS_P$ is a geometrically uniform partition with symmetry group $\Gamma(S_P') = v\Gamma(S_P)v^{-1}$.*

Proof: For $S'_P = vS_P$ to be a geometrically uniform partition there must exist a symmetry w that maps S'_i to S'_j for all $S'_i \in S'_P$ and $S'_j \in S'_P$ and such that $w(S'_P) = S'_P$. If we write $S'_i = vS_i$ and $S'_j = vS_j$ then the mapping $w = vuv^{-1}$ is such a symmetry provided a u exists such that $u(S_i) = S_j$. Since S_P is GU such a u exists. To show $w(S'_P) = S'_P$, we have

$$\begin{aligned} w(S'_P) &= vuv^{-1}S'_P \\ &= vuS_P \\ &= vS_P \\ &= S'_P \end{aligned} \tag{5.20}$$

Hence S'_P is GU with the symmetry group $v\Gamma(S_P)v^{-1}$. The example of figure 5.2 shows the image of a GU partition $S_P = \{S_0, S_1, S_2, S_3\}$, mapped through the transformation

$$v(\mathbf{x}) = A_v\mathbf{x} + \tau_v \tag{5.21}$$

where $A_v = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$, a pure rotation of 90 degrees clockwise, and $\tau_v = \begin{pmatrix} 4 \\ 2 \end{pmatrix}$. The partition S_P has

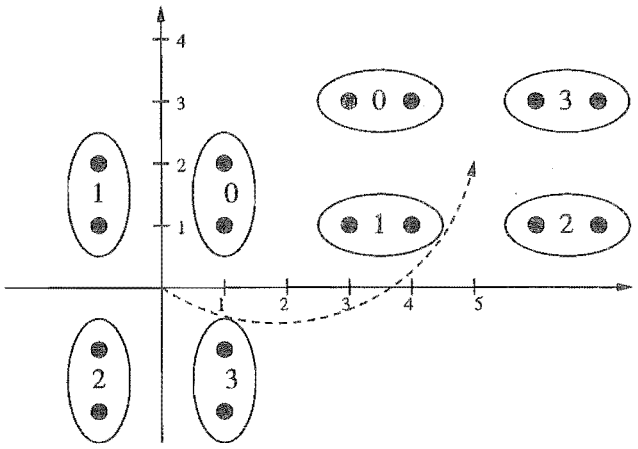


Figure 5.2: A GU partition mapped through an isometry is a GU partition.

$$
\begin{aligned}
S_0 &= \{(1,1),(1,2)\}, && (5.22)\\
S_1 &= \{(-1,1),(-1,2)\}, && (5.23)\\
S_2 &= \{(-1,-1),(-1,-2)\}, && (5.24)\\
S_3 &= \{(1,-1),(1,-2)\}, && (5.25)
\end{aligned}
$$

and four symmetries given by $u_i(\mathbf{x}) = A_i\mathbf{x}$ where

$$
\begin{aligned}
A_0 &= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad A_1 = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix} \\
A_2 &= \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \quad A_3 = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}
\end{aligned} \qquad (5.26)
$$

correspond to the identity, a 180 degree rotation about the origin, a reflection about the horizontal axis and a reflection about the vertical axis respectively. From theorem 11, the symmetries w_i of vS_P are given by

$$
\begin{aligned}
w_i(\mathbf{x}) &= vu_iv^{-1}\mathbf{x} \\
&= A_vA_i(A_v^{-1}\mathbf{x} - \tau_v) + \tau_v \\
&= A_vA_iA_v^{-1}\mathbf{x} + (I - A_vA_i^{-1}A_v)\tau_v \qquad (5.27)
\end{aligned}
$$

and form the symmetry group $\Gamma(vS_P)$.

Geometric uniformity of partitions.

Consider a geometrically uniform partition $S_P = \{S_0, S_1, \ldots, S_{n-1}\}$ with symmetry group $\Gamma(S_P)$.

Theorem 12 *If the subset S_0 itself can be written as a GU partition*

$$S_{0P} = \{S_{01}, S_{02}, \ldots\} \qquad (5.28)$$

then all subsets $S_i \in S_P$ are GU partitions.

Proof: From the definition of the geometric uniformity of the partition S_P, there must exist for any $S_i \in S_P$ a mapping $u \in \Gamma(S_P)$ such that $u(S_0) = S_i$. Theorem 11 shows that a GU partition under a distance invariant mapping is a GU partition, hence all $S_i \in S_P$ are GU partitions. For example for the partition S_{P_1} of figure 5.1(a), the subset $\{0,4\}$ is clearly GU, hence all subsets of S_{P_1} are GU.

5.3 GU partitions of non-GU constellations.

The definition of geometric uniformity over signal set partitions does not require the underlying signal constellation to be GU. For example, in figure 5.3 we have partitioned a 16-QAM constellation into four sets of four points. The set of symmetries of this partition are

$$U(S_P) = \{r_0, r_1, r_2, r_3, v_0, v_1, v_2, v_3\} \tag{5.29}$$

where r_i, is now defined as an anti-clockwise rotation about the origin of $\pi i/2$ radians, and v_i is defined as a reflection about the line passing through the origin and meeting at an angle of $\pi i/4$ to the horizontal. The equivalence classes contain one element only and as such the symmetry group of S_P is

$$\Gamma(S_P) = \{[r_0], [r_1], [r_2], [r_3], [v_0], [v_1], [v_2], [v_3]\} \tag{5.30}$$

Writing a non-GU constellation as a GU partition allows, as will be seen, the

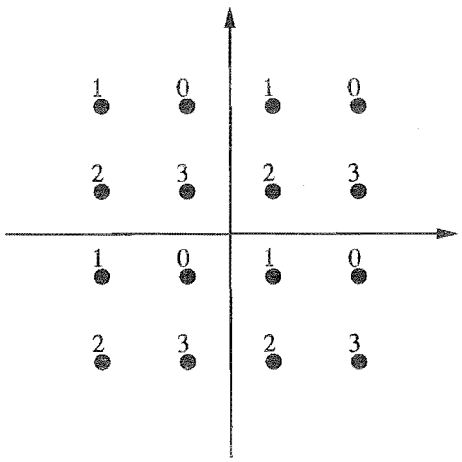


Figure 5.3: A four-way partition of a 16-QAM constellation.

construction of good multi-level codes such that each level of the code is GU.

5.4 Geometrically uniform constructions.

We have shown that if a partition S_P is GU then all of its distance properties can be completely characterised by considering only the distance properties of any one subset $S_i \in S_P$. Clearly this is a very strong property which vastly reduces the

complexity of the evaluation of the probability of error of a communications system. We now look at the techniques available for creating other GU constructions starting from a GU partition. A key tool is generating groups:

Definition 29 *Consider a GU partition S_P, with the symmetry group $\Gamma(S_P)$. A* generating group $G(S_P)$ *of S_P is a subgroup of $\Gamma(S_P)$ which is minimally sufficient to generate S_P starting from an arbitrary initial subset S_i of S_P. A necessary and sufficient condition for $G(S_P)$ to be a generating group of S_P is that:*

$$\begin{aligned} &\textit{For all } u \in G(S_P), \textit{ where } u \neq e \textit{ the identity element,} \\ &\textit{there does not exist a subset } S_i \in S_P \textit{ such that } u(S_i) = S_i, \end{aligned} \tag{5.31}$$

i.e. no symmetry of G can leave unchanged a subset of the partition S_P.

For example, the generating groups of the partition of figure 5.1(a) are

$$G_1(S_{P_1}) = \{[r_0, r_4], [r_1, r_5], [r_2, r_6], [r_3, r_7]\} \tag{5.32}$$

$$G_2(S_{P_1}) = \{[r_0, r_4], [r_2, r_6], [v_1, v_5], [v_3, v_7]\} \tag{5.33}$$

and the generating groups of the partition of figure 5.1(b) are

$$G_1(S_{P_2}) = \{[r_0], [r_2], [r_4], [r_6]\} \tag{5.34}$$

$$G_2(S_{P_2}) = \{[r_0], [r_4], [v_3], [v_7]\} \tag{5.35}$$

Notice how the operation of all the elements of either group on any subset $S_i \in S_{P_j}$ generates S_{P_j} completely and uniquely.

5.4.1 Subgroups of generating groups.

Consider a generating group $G(S_P)$ of a GU partition S_P. From group algebra, a normal subgroup G_n of G partitions G into G_n and its cosets. The orbits of any partition S_i in S_P, under G_n and its cosets, are disjoint subsets of S_P whose union is S_P. Following a similar proof to Forney's we can show that these subsets, which form a partitioning of S_P, are geometrically uniform, mutually congruent, and have G_n as a common generating group [41].

Hence any normal subgroup of a generating group G can be used to construct a partition which is geometrically uniform. For example the non-trivial normal

subgroups of the generating group $G_2(S_{P_1})$ of the partition of figure 5.1(a) are

$$G_{21}(S_{P_1}) = \{[r_0, r_4], [r_2, r_6]\} \quad (5.36)$$

$$G_{22}(S_{P_1}) = \{[r_0, r_4], [v_1, v_5]\} \quad (5.37)$$

$$G_{23}(S_{P_1}) = \{[r_0, r_4], [v_3, v_7]\} \quad (5.38)$$

which partition S_{P_1} into the three GU partitions, namely $S_{P_{11}} = \{S_0, S_2\}$, $S_{P_{12}} = \{S_0, S_1\}$, and $S_{P_{13}} = \{S_0, S_3\}$. These three partitions $S_{P_{11}}$, $S_{P_{12}}$ and $S_{P_{13}}$ of S_{P_1} are shown in figures 5.4(a), 5.4(b) and 5.4(c) respectively.

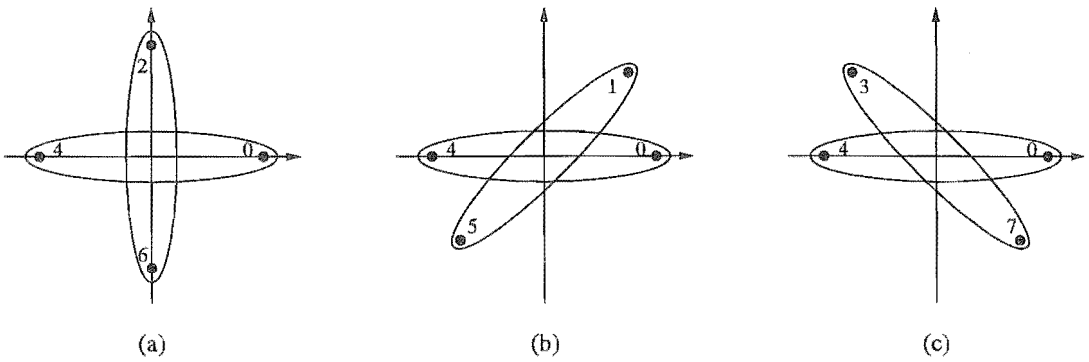


Figure 5.4: GU partitions generated by normal subgroups of a generating of S_{P_1}. (a) Partition $S_{P_{11}}$. (b) Partition $S_{P_{12}}$. (c) Partition $S_{P_{13}}$.

5.4.2 GU partition generation algorithm.

The algorithm of table 5.2 generates GU partitions S_P of a GU constellation S, given a generating group $G(S)$ of S and a normal subgroup G_i of $G(S)$. If $|G(S)| = 2^n$ and $|G_i| = 2^i$ then the algorithm constructs 2^k unique solutions, where $k = i(2^{n-i} - 1)$. The order of S_P is $|S_P| = |G_i|$. For example consider constellation $S = 8$-PSK with generating group $G(S) = \{r_0, r_1, r_2, r_3, r_4, r_5, r_6, r_7\}$ and subgroup $G_i = \{r_0, r_2, r_4, r_6\}$. We have $n = 3$, $i = 2$ and the number of solutions generated by the algorithm is $2^{2(2^{3-2}-1)} = 4$. These are shown in figure 5.5.

5.4.3 Quotient groups.

Consider a generating group $G(S_P)$ of a GU partition S_P and a normal subgroup G_n of G. The quotient group G/G_n is a set of elements such that each element is a

Input : $G(S)$, a generating group of the signal constellation S.
G_i a normal subgroup of G.
Output : A GU partition S_P of S.

begin
Set $R = G - G_i$, Set $S_i = \{e\}$
while $R \neq \emptyset$ **do**
Select any $r \in R$.
$S_i = S_i + r$
$R = R - rG_i$
end while
$S_P = \{gS_i : g \in G_i\}$
end.

Table 5.2: An algorithm to construct GU partitions from a GU constellation S and a normal subgroup G_i of a generating group $G(S)$ of S.

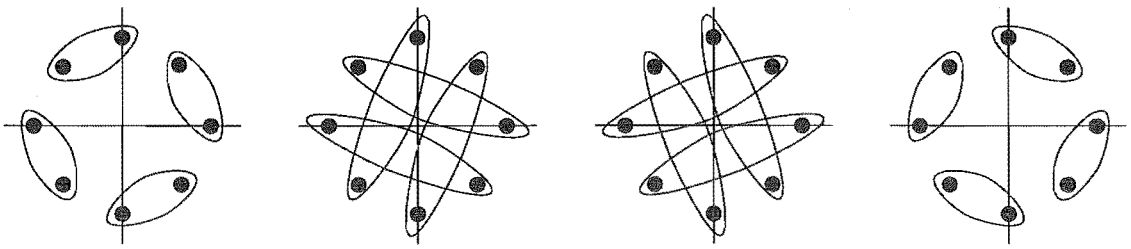


Figure 5.5: GU partitions generated by the algorithm of table 5.2.

union of subsets of S_P. Following a similar proof to that by Forney, it may be shown that the quotient group G/G_n forms a GU partition [41]. For example consider the partition S_{P_1} of an 8-PSK constellation shown in figure 5.1(a) and generating group $G_2(S_{P_1})$ of equation (5.33). The three normal subgroups of $G_2(S_{P_1})$ are listed in equations (5.37), (5.38), and (5.38) respectively and the corresponding quotient

groups are

$$G_2/G_{21} = \{\{[r_0, r_4], [r_2, r_6]\}, \{[v_1, v_5], [v_3, v_7]\}\} \quad (5.39)$$

$$G_2/G_{22} = \{\{[r_0, r_4], [v_1, v_5]\}, \{[r_2, r_6], [v_3, v_7]\}\} \quad (5.40)$$

$$G_2/G_{23} = \{\{[r_0, r_4], [v_3, v_7]\}, \{[r_2, r_6], [v_1, v_5]\}\}. \quad (5.41)$$

Figures 5.6(a), 5.6(b), and 5.6(c) show the partitions formed by G_2/G_{21}, G_2/G_{22}, and G_2/G_{23} respectively.

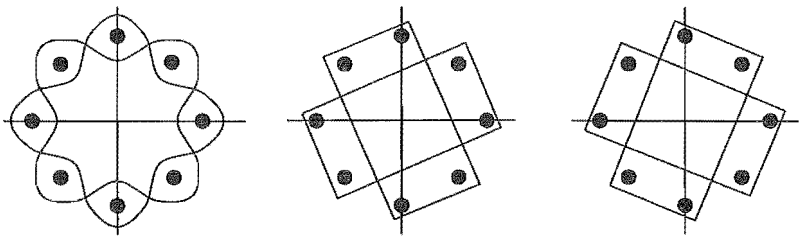


Figure 5.6: GU partitions generated by quotient groups. (a) G_2/G_{21}. (b) G_2/G_{22}. (c) G_2/G_{23}.

5.4.4 Cartesian products of GU partitions.

Consider two partitions S_{P_i} and S_{P_j} which are both GU under the set of distance measures $\mathcal{D}$. The Cartesian product of S_{P_i} and S_{P_j} is:

$$S_P = S_{P_i} \times S_{P_j} = \{S_i \times S_j \ : \ S_i \in S_{P_i} \text{ and } S_j \in S_{P_j}\} \quad (5.42)$$

where the set $S_i \times S_j$ equals:

$$S_i \times S_j = \{(s_i, s_j) : \ s_i \in S_i \text{ and } s_j \in S_j\} \quad (5.43)$$

The order of S_P is $|S_{P_i}| \times |S_{P_j}|$ and each subset of S_P contains $|S_i| \times |S_j|$ points. If $u_1 \in \Gamma(S_{P_1})$, the symmetry group of S_{P_1} and $u_2 \in \Gamma(S_{P_2})$, the symmetry group of S_{P_2}, then (u_1, u_2) is a symmetry of $\Gamma(S_P)$ only if all the distance measures $D_i \in \mathcal{D}$ are such that

$$D_i((\mathbf{x}_1, \mathbf{x}_2), (\mathbf{y}_1, \mathbf{y}_2)) = D_i((u_1(\mathbf{x}_1), u_2(\mathbf{x}_2)), (u_1(\mathbf{y}_1), u_2(\mathbf{y}_2))) \quad (5.44)$$

If equation (5.44) holds then a group of symmetries of S_P is

$$\begin{aligned} \Gamma(S_P) &= \Gamma(S_{P_i}) \times \Gamma(S_{P_j}) \\ &= \{(u_i, u_j) : u_i \in \Gamma(S_{P_i}) \text{ and } u_j \in \Gamma(S_{P_j})\} \end{aligned} \quad (5.45)$$

Note that $\Gamma(S_P)$ is not necessarily the *complete* symmetry group of S_P. The operation of a symmetry $u \in \Gamma(S_P)$ on an element $S \in S_P$ is defined by

$$u(S) = u_i(S_i) \times u_j(S_j) \tag{5.46}$$

where $S = S_i \times S_j$, and $u = (u_i, u_j)$.

Theorem 13 *If S_{P_i} and S_{P_j} are GU partitions under the set of distance measures $\mathcal{D}$ and every $D_i \in \mathcal{D}$ satisfies equation (5.44), then the partition S_P formed by the product of S_{P_i} and S_{P_j} as defined by equation* (5.42) *is geometrically uniform.*

Proof: To prove S_P is GU we need to show the following: 1) There exists a u that maps any subset $S_1 \in S_P$ to any other $S_2 \in S_P$. 2) The mapping u is a symmetry of S_P. Consider any two subsets of S_P, $S_1 = S_{i_1} \times S_{j_1}$ and $S_2 = S_{i_2} \times S_{j_2}$. From equation (5.46), the mapping $u = (u_{i_1,i_2}, u_{j_1,j_2})$ maps S_1 to S_2 if $u_{i_1,i_2}(S_{i_1}) = S_{i_2}$ and $u_{j_1,j_2}(S_{j_1}) = S_{j_2}$. Such u_{i_1,i_2} and u_{j_1,j_2} exist because S_{P_i} and S_{P_j} are GU, and hence 1) is shown. Clearly the mapping u is a symmetry of S_P.

From theorem 13 it follows that powers of GU partitions are GU, provided the set of distance measures satisfies equation (5.44).

5.4.5 Iterative construction.

Useful GU partitions may be constructed iteratively from a set of GU constellations combined with an arbitrary constellation. Consider a GU constellation S with the corresponding symmetry group $\Gamma(S)$ and a constellation S_0.

Theorem 14 *If S is a geometrically uniform constellation, s is a point in S, $G(S)$ is a generating group of S, and S_0 is an arbitrary constellation then the partition defined by:*

$$S_P = \{g(S_0 + s) : g \in G(S)\}, \tag{5.47}$$

is geometrically uniform, provided $g_i(S_0+s) \cap g_j(S_0+s) = \emptyset$, $g_i \in G, g_j \in G, g_i \neq g_j$, i.e. the subsets do not intersect.

Proof: For S_P to be a GU partition we require that there exists an isometry w for any $S_i \in S_P$ and $S_j \in S_P$ such that $w(S_i) = w(S_j)$ and $w(S_P) = S_P$. If we write $S_i = g_i(S_0 + s)$ and $S_j = g_j(S_0 + s)$ then the mapping $w = g_i^{-1} g_j$ maps S_i to S_j.

The operation of w on any element $S_k = g_k(S_0 + s)$ of S_P is $g_k g_i^{-1} g_j(S_0 + s)$ and is an element of S_P, hence $w(S_P) = S_P$ and S_P is GU.

Note that the result of this construction depends on to which point $s \in S$ it is applied. For example, figure 5.7(a) shows a GU 4-PSK constellation S with generating group $G(S) = \{r_0, r_1, r_2, r_3\}$, and figure 5.7(b) shows a non-GU constellation S_0. The construction $\{g(S_0 + s) : g \in G(S)\}$ applied to point 0 of S results in a partitioning of 16-QAM as shown in 5.7(c). Note that the application of this construction to point 1 of S results in the same partitioning but with a different labelling, while the application to points 2 and 3 causes the subsets to intersect and is therefore not valid.

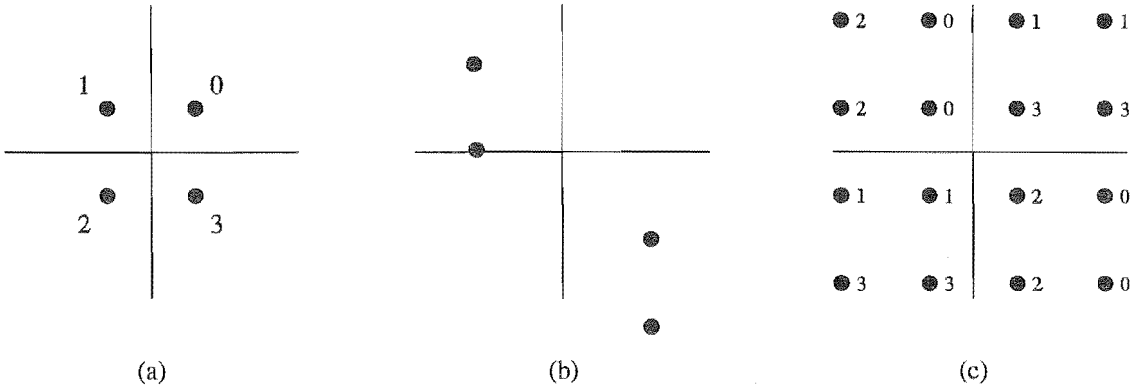


Figure 5.7: A geometrically uniform partition constructed from the constellations S and S_0. (a) The GU constellation S. (b) The constellation S_0. (c) Result of the iterative construction applied to point 0 of S.

Definition 30 *We will denote the iterative construction of theorem 14 by the notation* $I_c(\cdot)$*:*

$$\begin{aligned} S_P = I_c(S_0, S) &= \{g(S_0 + s_0) : g \in G(S)\} \\ &= \{hS : h \in H(S)\} \end{aligned} \tag{5.48}$$

The set $H(S)$ contains the mappings $h(\mathbf{x}) = g(\mathbf{x} + s_0)$, $g \in G(S)$ and is used to simplify the notation. The s_0 is the point in S with the label 0. No generality is lost by selecting the point s_0 because S may be relabelled such that s_0 corresponds to any point in S.

Definition 31 *Given a partition S_P we use the operator $\Lambda(S_P)$ to convert S_P to a constellation S,*

$$\Lambda(S_P) = \{s : \ s \in S_i \in S_P\}. \tag{5.49}$$

For example $\Lambda(\{\{0,1\},\{2,3\}\}) = \{0,1,2,3\}$. The construction described by theorem 14 can be applied iteratively to a constellation S_0 and a set of GU constellations $\{S_1, S_2, \ldots, S_n\}$ with associated generating groups $\{G_1(S_1), G_2(S_2), \ldots, G_n(S_n)\}$ to form partitions of arbitrary complexity. We may associate with each S_i, $i > 0$ a constellation S_i' constructed iteratively from the constellations $S_0, S_1, \ldots, S_i$:

$$S_i' = \Lambda(I_c(S_{i-1}', S_i)). \tag{5.50}$$

Each S_i' may also be written in terms of the mappings $H(S_i)$ defined above

$$S_i' = \{h_i h_{i-1} \cdots h_1 S_0 : h_i \in H(S_i)\}. \tag{5.51}$$

5.5 Partition trees.

Many signal space coding techniques are based on iteratively partitioning a signal constellation S into a number of smaller sets [75]. The structure of such a partition can be drawn as a *partition tree.*

Definition 32 *A partition tree of a signal constellation S is an $l+1$ level partition structure $\mathcal{S}_P = \{S_{P_0}, S_{P_1}, \ldots, S_{P_l}\}$, where each S_{P_i} is a partitioning of S. The top partition in the structure is the trivial partition $S_{P_l} = \{S\}$. Each lower partition S_{P_i} is constructed from the partition above, $S_{P_{i+1}}$, by partitioning each set of $S_{P_{i+1}}$ into m_{i+1} subsets, each containing $|S|/|S_{P_{i+1}}|/m_{i+1}$ points. The bottom partition S_{P_0} is the partitioning of S into single points. We have $|S| = \prod_{i=1}^{l} m_i$ and $m_i = |S_{P_i}|/|S_{P_{i-1}}|$.*

For example, the multi-level partitioning of 8-PSK as shown in figure 5.8 is such a partition tree structure with $l = 2$. The partitioning structure is written as $\mathcal{S}_P = \{S_{P_0}, S_{P_1}, S_{P_2}\}$, where $S_{P_0} = \{\{0\}, \{1\}, \{2\}, \{3\}, \{4\}, \{5\}, \{6\}, \{7\}\}$, $S_{P_1} = \{\{0,1\}, \{2,3\}, \{4,5\}, \{6,7\}\}$, and $S_{P_2} = \{\{0,1,2,3,4,5,6,7\}\}$. The partition orders are $m_1 = 2$ and $m_2 = 4$.

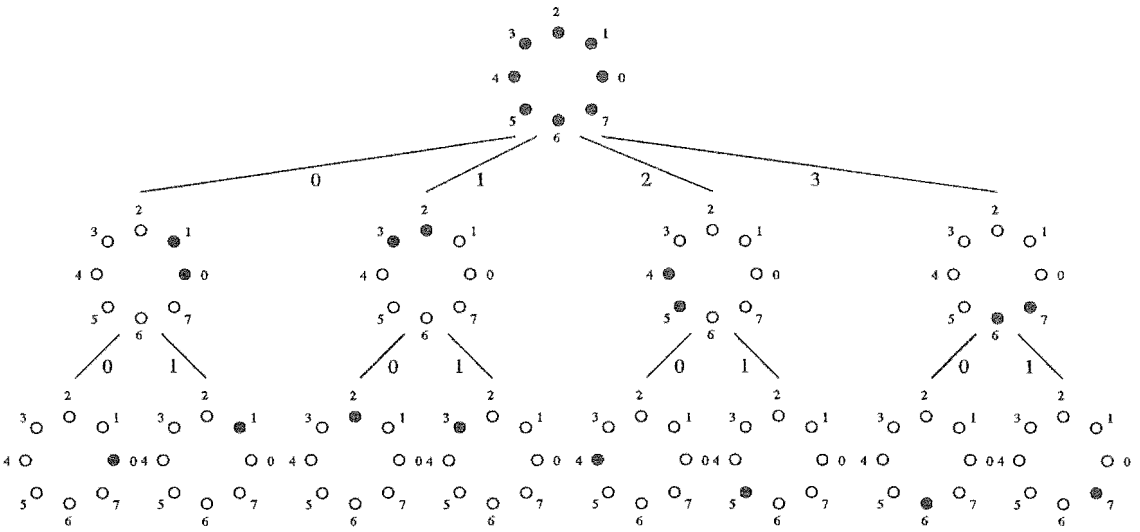


Figure 5.8: A multi-level partitioning of 8-PSK.

The number of sets at level i is

$$|S_{P_i}| = \prod_{j=i+1}^{l} m_j \tag{5.52}$$

A set of a partition at level i is labelled by the string $a_l a_{l-1} \cdots a_{i+1}$ where $0 \leq a_j < m_j$ for $i < j \leq l$. With this convention, a point in S is labelled $a_l a_{l-1} \cdots a_1$. All points whose label starts with the string $a_l a_{l-1} \cdots a_i$ belong to the partition set labelled $a_l a_{l-1} \cdots a_i$. For example the four sets of level 1 of the partition tree of figure 5.8 are labelled 0, 1, 2, and 3 respectively from left to right. The eight points at level 2 of the partition tree of figure 5.8 are labelled 00, 01, 10, 11, 20, 21, 30 and 31 respectively from left to right.

5.5.1 Partition trees of GU constellations.

Geometrically uniform constellations allow a natural partitioning tree based on the group structure of the underlying generating groups. Consider a GU constellation S, and an associated generating group $G(S)$. A normal subgroup chain of length $l+1$ of G is written as $G_0 < G_1 < \cdots < G_l$, where $G_0 = \{e\}$, i.e. the subgroup containing only the identity element, and $G_l = G$. Each G_i is such that it is a normal subgroup of G_l. From the theorems of group algebra, it can be shown that since $G_i \lhd G_l$ for all i, we have $G_i \lhd G_{i+j}$ for $0 \leq j \leq l-i$, i.e. G_i is a normal subgroup of every group above it in the chain. The division of the group G by a normal subgroup G_i

partitions G into G_i and its cosets. From the definition of a generating group, there is a one to one mapping between G and S, namely $S = G(s_0)$ and correspondingly $G/G_i(s_0)$ partitions S into $|G|/|G_i|$ sets.

Definition 33 *Given a GU signal constellation S, a generating group $G(S)$ of S, and a normal subgroup chain $G_0 \lhd G_1 \lhd \cdots \lhd G_l$ of G, a partition tree $\mathcal{S}_P$ of S is*

$$\mathcal{S}_P = \{G/G_i(s_0) : \ 0 \leq i \leq l\} \tag{5.53}$$

The order of the partition at level i is

$$m_i = |G_i|/|G_{i-1}| \tag{5.54}$$

With this definition, the partition branch labels a_i, at level i of the partition tree now correspond to the coset representatives of G_i/G_{i-1}, that is each g_i is a coset representative of G_i/G_{i-1}. With such a branch labelling scheme, a partition set j at level i may be written as the product

$$S_{ij} = g_l g_{l-1} \cdots g_{i+1} G_i(s_0), \tag{5.55}$$

i.e. the sets at level i are the cosets of $G_i(S_0)$.

Distance properties.

Consider the GU partition S_P formed by the quotient group G/G_n, where G is a generating group of a GU constellation S and G_n is a normal subgroup of G.

Theorem 15 *Consider two subsets S_i and S_j of the partition S_P. The distance profiles $DP(S_i, s_{j1})$ and $DP(S_i, s_{j2})$ are equal for any two points s_{j1} and s_{j2} in S_j.*

Proof: The constellation S is GU and therefore there exists an isometry u such that $u(s_{j1}) = u(s_{j2})$. Since s_{j1} and s_{j2} are both elements of a coset of G_n, u must be an element of G_n. Therefore from the definition of equation (5.14) for the distance profile

$$\begin{aligned} DP(S_i, s_{j1}) &= \{D(s_i, s_{j1}) : s_i \in S_i\} \\ &= \{D(u(s_i), u(s_{j1})) : s_i \in S_i\} \\ &= \{D(s_i', s_{j2}) : s_i' \in S_i\} \\ &= DP(S_i, s_{j2}). \end{aligned} \tag{5.56}$$

This theorem implies we can compute the distance profile between any two partitions S_i and S_j of S_P simply by computing the distance profile from *one* point in S_i to the points in S_j.

5.5.2 Partition trees of iterative constructions.

The iterative construction of a GU partition described in section 5.4.5 has a natural partition tree associated with it. For a set of GU constellations $\{S_1, S_2, \ldots, S_l\}$ and corresponding sets of mappings $\{H_1, H_2, \ldots, H_l\}$ as described by definition 30 the partition tree of the iterative construction may be written as

$$\mathcal{S}_P = \{S_{P_0}, S_{P_1}, \ldots, S_{P_l}\} \quad . \tag{5.57}$$

where each partition is defined by

$$S_{P_i} = \{h_l h_{l-1} \ldots h_i S_i'\} \tag{5.58}$$

where S_i' is as defined in equation (5.50).

5.6 Communication utilising partitions.

A conventional uncoded communications system maps k input binary data bits onto a point of a constellation S of order 2^k. This point is transmitted across the channel and the decoder decides on the transmitted data by selecting the point most likely to have been transmitted. Consider the system shown in figure 5.9 as a generalisation of the basic uncoded communication system. This system, instead of mapping the

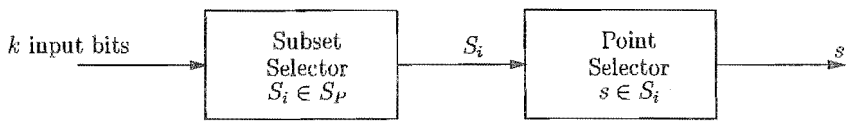


Figure 5.9: An uncoded communication system based on the transmission of partitions.

input data to point in a constellation, maps the data to a *subset* S_i of a partition S_P, where $|S_P| = 2^k$. From the partition S_i, *one* point s is randomly selected for transmission by the point selector. The method of selection is not known by the receiver, however it is assumed that each point is equiprobable. The receiver's task is to decide on the k transmitted data bits.

5.6.1 ML receiver for a partition transmitter.

A maximum likelihood solution for the receiver is to compute the probability density function of the received vector conditioned on the transmitted subset S_i. With the assumption that each point $s \in S_i$ is equiprobable, the pdf may be written as the sum of the conditional pdf's of the points in the partition. Thus

$$p(\mathbf{r}|S_i) = \frac{1}{|S_i|} \sum_{s \in S_i} p(\mathbf{r}|s) \tag{5.59}$$

The ML decision is the subset S_i such that $p(\mathbf{r}|S_i)$ is a maximum. If the subsets of the partition contain only one point each, then the system reduces to a conventional uncoded system. For the Rayleigh fading channel the pdf of the received signal $\mathbf{r}$ conditioned on the transmitted point s and on the receiver obtaining ideal channel state information $\mathbf{v}$ is given by

$$p(\mathbf{r}, \mathbf{v}|s) = \frac{1}{(2\pi N_0)^L} e^{\frac{-||\mathbf{r}-\mathbf{v}s||^2}{2N_0}} \tag{5.60}$$

where L is the length of the received vector $\mathbf{r}$. This may be viewed as the case identical to the additive white Gaussian noise (AWGN) channel with the SNR scaled by the time varying Rayleigh distributed variable $|v|^2$. The ML decoding metric $M(\mathbf{r}, S_i)$ for the Rayleigh fading channel, assuming subsets with equal number of points, therefore is

$$M(\mathbf{r}, \mathbf{v}, S_i) = \sum_{s \in S_i} e^{\frac{-||\mathbf{r}-\mathbf{v}s||^2}{2N_0}} \tag{5.61}$$

and the decoder selects $S_i \in S_P$ such that $M(\mathbf{r}, \mathbf{v}, S_i)$ is a maximum.

5.6.2 Metric calculation for trellis generated subsets.

The receiver of the system of figure 5.9 has to compute the metric of equation (5.59) for each possibly selected subset and it chooses the one with the maximum metric. If the size of the partitions is large (as is possible in a multi-dimensional constellation) then the computation can be expensive. If it is possible to represent the points of a subset by the paths through a trellis then the computation may be greatly simplified by an algorithm similar to the Viterbi algorithm (VA) [38]. Consider a subset S_0 containing $|S_0|$ points, where each point s can be written as an L-tuple, $\mathbf{c}_j = \{c_{1j}, c_{2j}, \ldots, c_{Lj}\}$. Given a received vector $\mathbf{r} = \{r_1, r_2, \ldots, r_L\}$, and a channel

state information vector $\mathbf{v} = \{v_1, v_2, \ldots, v_L\}$, the decoding metric for the partition is

$$M(S_0) = \sum_{j=1}^{|S_0|} e^{-\frac{1}{2N_0}\sum_{i=1}^{L} d_{ij}^2} \tag{5.62}$$

where

$$d_{ij}^2 = |r_i - v_i c_{ij}|^2 \tag{5.63}$$

the squared distance between the i^{th} received symbol and the i^{th} component of the j^{th} signal point. Equation (5.62) may also be written as

$$M(S_0) = \sum_{j=1}^{|S_0|} \prod_{i=1}^{L} \delta_{ij} \tag{5.64}$$

where

$$\delta_{ij} = e^{-d_{ij}^2/2N_0} \tag{5.65}$$

The general algorithm to compute the soft decision metric of equation (5.62) is shown in table 5.3. The number of computations of a term $D = e^{-|r-vc|^2}$ the algorithm requires is

$$\xi_{\text{algorithm}} = \sum_{t=0}^{L-1} N_t B_t \tag{5.66}$$

where B_t is the number of branches leaving each state at time t and N_t is the number of states at time t. The direct computation requires the number of points in the subset $|S_0|$ multiplied by the number of symbols per sequence. The number of points, corresponding to the number of paths through the trellis, is

$$|S_0| = \prod_{t=0}^{L-1} B_t \tag{5.67}$$

and hence the number of computations of D terms for a direct implementation is

$$\xi_{\text{direct}} = L \prod_{t=0}^{L-1} B_t \tag{5.68}$$

Clearly the metric computation algorithm is more efficient. For example, consider the trellis of figure 5.10 used to generate a partition S_{P_i}. The trellis generates the 16 points listed in table 5.4. If for convenience we write the notation

Definitions:

$b_t(s_1, s_2)$: The set of branch symbols connecting states s_1 and s_2 at time t, where $0 \leq t \leq L$.

$m(s,t)$: The distance metric of state s at time t.

Input : The received vector $\mathbf{r} = \{r_1, r_2, \ldots, r_L\}$.
The channel estimate vector $\mathbf{v} = \{v_1, v_2, \ldots, v_L\}$.
The N_t defining the number of states at time t.
The $b_t(s_1, s_2)$ defining the trellis of the subset.

Output : The distance measure $M(\mathbf{r}, S_0)$.

1. $t = 0$
2. $m(0,t) = 0$
3. $t = t + 1$
4. $m(s,t) = m(s,t) + m(s', t-1)e^{-|r_t - v_t b_t(s',s)|^2/2N_0}$, $0 \leq s < N_t, 0 \leq s' < N_{t-1}$
5. if $t < L$ goto 3
6. output $M = \sum_{s=0}^{N_L - 1} m(s, L)$

Table 5.3: An algorithm to compute the decoding measure of a subset generated by a trellis.

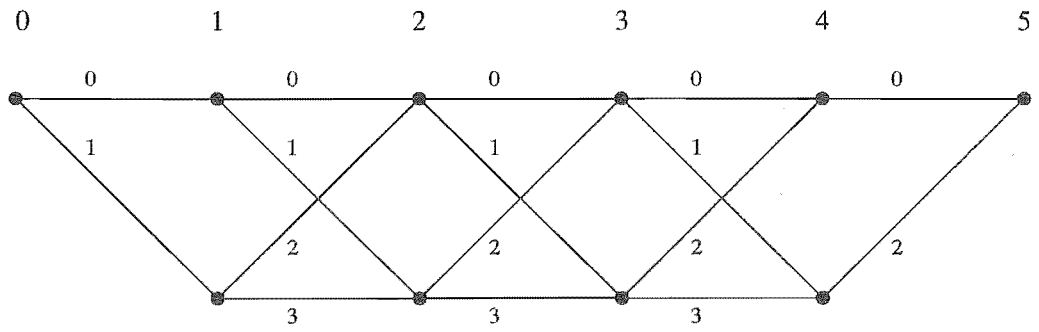


Figure 5.10: Trellis to generate a set of points.

$$D_{ic} = e^{-|r_i - v_i c|^2/2N_0} \tag{5.69}$$

j	$b_{1j}b_{2j}b_{3j}b_{4j}$	$\{c_{1j}c_{2j}c_{3j}c_{4j}c_{5j}\}$	j	$b_{1j}b_{2j}b_{3j}b_{4j}$	$\{c_{1j}c_{2j}c_{3j}c_{4j}c_{5j}\}$
0	0000	00000	8	1000	12000
1	0001	00012	9	1001	12012
2	0010	00120	10	1010	12120
3	0011	00132	11	1011	12132
4	0100	01200	12	1100	13200
5	0101	01212	13	1101	13212
6	0110	01320	14	1110	13320
7	0111	01332	15	1111	13332

Table 5.4: Codewords generated by the trellis of figure 5.10.

to mean the distance measure between the received point $\dot{r}_i$ at time i and the symbol c then the algorithm of table 5.3 generates the following values for the state measures $m(s,t)$ for the example trellis of figure 5.10.

Note the terms of $m(0,5)$ correspond precisely to the codewords of $|S_0|$ listed in

t	$m(s,t)$	
0	$m(0,0) =$	0
1	$m(0,1) =$	D_{10}
	$m(1,1) =$	D_{11}
2	$m(0,2) =$	$D_{10}D_{20} + D_{11}D_{22}$
	$m(1,2) =$	$D_{10}D_{21} + D_{11}D_{32}$
3	$m(0,3) =$	$D_{10}D_{20}D_{30} + D_{11}D_{22}D_{30} + D_{10}D_{21}D_{32} + D_{11}D_{32}D_{32}$
	$m(1,3) =$	$D_{10}D_{20}D_{31} + D_{11}D_{22}D_{31} + D_{10}D_{21}D_{33} + D_{11}D_{32}D_{33}$
4	$m(0,4) =$	$D_{10}D_{20}D_{30}D_{40} + D_{11}D_{22}D_{30}D_{40} + D_{10}D_{21}D_{32}D_{40} + D_{11}D_{32}D_{32}D_{40}+$
		$D_{10}D_{20}D_{31}D_{42} + D_{11}D_{22}D_{31}D_{42} + D_{10}D_{21}D_{33}D_{42} + D_{11}D_{32}D_{33}D_{42}$
	$m(1,4) =$	$D_{10}D_{20}D_{30}D_{42} + D_{11}D_{22}D_{30}D_{42} + D_{10}D_{21}D_{32}D_{42} + D_{11}D_{32}D_{32}D_{42}+$
		$D_{10}D_{20}D_{31}D_{43} + D_{11}D_{22}D_{31}D_{43} + D_{10}D_{21}D_{33}D_{43} + D_{11}D_{32}D_{33}D_{43}$
5	$m(0,5) =$	$D_{10}D_{20}D_{30}D_{40}D_{50} + D_{11}D_{22}D_{30}D_{40}D_{50} + D_{10}D_{21}D_{32}D_{40}D_{50}+$
		$D_{11}D_{32}D_{32}D_{40}D_{50} + D_{10}D_{20}D_{31}D_{42}D_{50} + D_{11}D_{22}D_{31}D_{42}D_{50}+$
		$D_{10}D_{21}D_{33}D_{42}D_{50} + D_{11}D_{32}D_{33}D_{42}D_{50} + D_{10}D_{20}D_{30}D_{42}D_{52}+$
		$D_{10}D_{20}D_{31}D_{43}D_{52} + D_{11}D_{22}D_{31}D_{43}D_{52} + D_{10}D_{21}D_{33}D_{43}D_{52}+$
		$D_{11}D_{32}D_{33}D_{43}D_{52} + D_{11}D_{22}D_{30}D_{42}D_{52} + D_{10}D_{21}D_{32}D_{42}D_{52}+$
		$D_{11}D_{32}D_{32}D_{42}D_{52}$

Table 5.5: Application of metric computation algorithm to the trellis of figure 5.10.

table 5.4.

5.6.3 Variance of decision regions with SNR.

For the system of figure 5.9, transmitting partitions and applying ML decoding, the decision regions implied by equation (5.59), are in general a function of the SNR. For example in figure 5.11 we have plotted the decision regions of two partitions as a function of SNR. Figure 5.11(a) are the decision regions at a high SNR. As the SNR decreases the decision regions change as shown in figure 5.11(b), until at a very low SNR the decision regions become as in figure 5.11(c). Clearly the shape

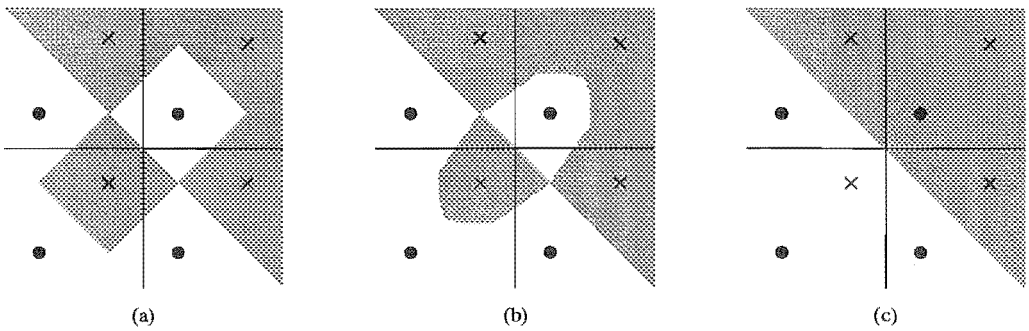


Figure 5.11: The partition decision regions as a function of SNR. (a) High SNR (15 dB). (b) Low SNR (5 dB). (c) Very low SNR (0 dB).

of the regions change significantly, especially at low SNR, however note they always remain congruent as guaranteed by theorem 8.

5.6.4 Decision regions at high SNR.

At a high SNR, equation (5.59) is dominated by the largest term $p(\mathbf{r}|\mathbf{c}_i)$ and we have

$$p(\mathbf{r}|S_i)|_{\text{Large SNR}} \approx \max_{s \in S_i} p(\mathbf{r}|s) \tag{5.70}$$

The final decision by the decoder is the $S_i \in S_P$ with the maximum conditional density function, i.e.

$$\max_{S_i \in S_P} p(\mathbf{r}|S_i) \tag{5.71}$$

Clearly this is equivalent when combined with equation (5.70) to selecting the point $s \in \Lambda(S_P)$ such that $p(\mathbf{r}|s)$ is a maximum and equates to selecting the point in the

$\Lambda(S_P)$ assuming conventional uncoded modulation. The decision region of S_i for this case may be written simply as the union of the decision regions of the points in S_i. Thus

$$R_V(S_i) = \cup_{s \in S_i} R_V(s) \tag{5.72}$$

For the Rayleigh fading channel we have

$$p(\mathbf{r}|s) = \frac{1}{(2\pi N_0)^L} e^{\frac{-\|\mathbf{r}-\mathbf{v}s\|^2}{2N_0}} \tag{5.73}$$

and the decoding strategy is therefore to select the point $s \in \Lambda(S_P)$ with the minimum Euclidean distance to the received point $\mathbf{r}$. The output of the decoder is then the partition S_i corresponding to s.

5.6.5 Suboptimal decoding of partitions.

The ML decoding metric of equation (5.59) for the decoding of partitions over the Rayleigh fading process is computationally more complex than finding the point with the minimum Euclidean distance to the received point. We have shown in section 5.6.4 that for high SNR on the Rayleigh fading channel, the ML solution tends towards a minimum Euclidean distance solution. In this section we investigate the loss in performance due to applying a Euclidean distance metric compared to the ML metric. In figure 5.12 we have plotted the pairwise probability of error of confusing one partition of figure 5.11 for the other as a function of the SNR. This result has been obtained by simulation. The loss in performance is negligible across the range of SNR depicted. Clearly there is little to be gained by the ML decoding metric compared to a Euclidean distance metric.

5.6.6 Conditions for invariance of decision regions.

Although we have empirically shown that minimum Euclidean distance decoding of partitions transmitted across the Rayleigh fading channel performs almost as well as ML decoding, it is possible to design the code partitions such that the two decoding techniques are equivalent. This is done by selecting the partitions such that the decisions boundaries are not a function of SNR. If this is the case then minimum Euclidean distance decoding is equivalent to ML decoding. The result follows from

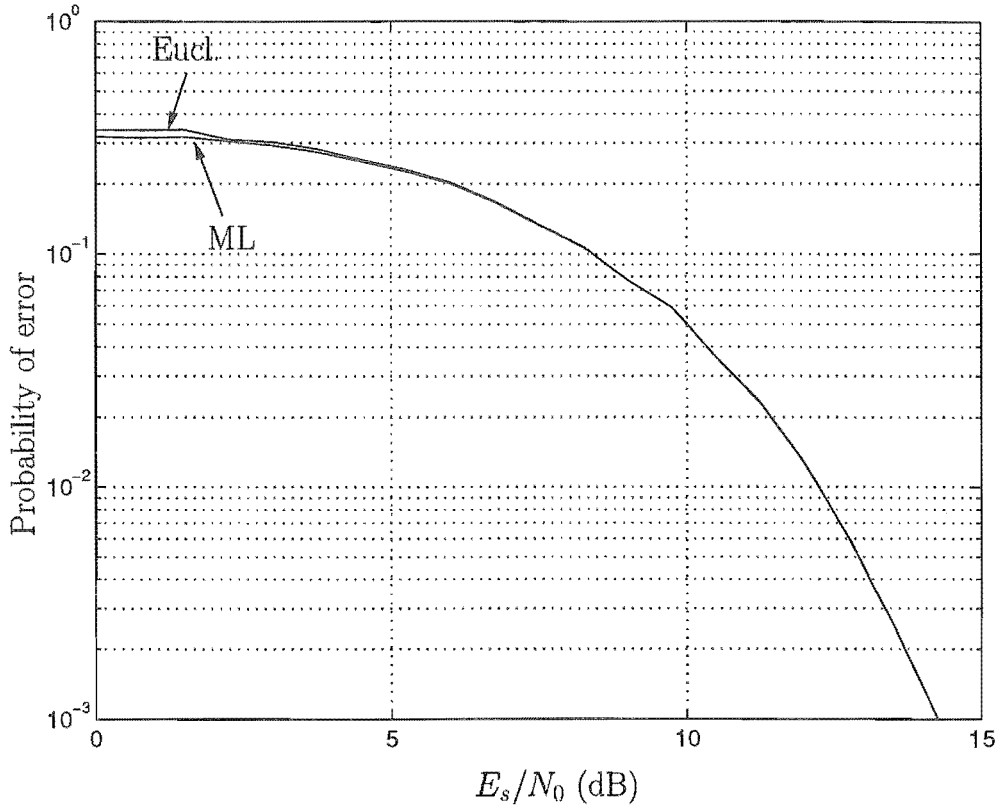


Figure 5.12: Simulation of the probability of confusing one partition of figure 5.11 for the other as a function of SNR, for the ML decoder and the minimum Euclidean distance decoder.

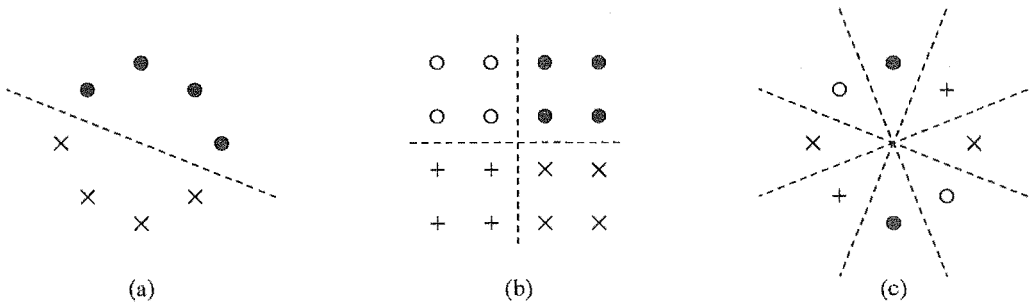


Figure 5.13: Examples of three partitions with decision boundaries invariant with the SNR.

section 5.6.4 where we have shown that at high SNR the ML decision regions are determined by the minimum Euclidean distance to the points, and if the decision regions are invariant to the SNR then the decision regions must be determined by the minimum Euclidean distance for *all* values of the SNR. In figure 5.13 we show three examples of partitions where the decision regions are not a function of the SNR. To ensure the decision regions of a partition $S_P = \{S_0, S_2, \ldots, S_{n-1}\}$ are invariant with SNR, we must have the ML decision metric $M(\mathbf{r}, S_i)$ of equation (5.61) making the

same decision, *independently* of the value of N_0. The decision metric $M(\mathbf{r}, S_i)$ may be written as a function of the set of the Euclidean distances between $\mathbf{r}$ and the points in S_i, i.e.

$$M(\mathbf{r}, S_i) = f(d^2(\mathbf{r}, S_i), N_0) \tag{5.74}$$

Theorem 16 *Consider a partition $S_P = \{S_1, S_2, \ldots, S_n\}$ and a decision metric $f(d^2(\mathbf{x}, S_i))$ which is to be written as a function of the Euclidean distance between $\mathbf{x}$ and each point $s \in S_i$. There exists a set of hyperplanes $P = \{P_1, P_2, \ldots, P_k\}$, bounding the decision regions. Each hyperplane divides the Euclidean space, and consequently the set of partitions S_P into two. If the reflection about every plane P_i maps S_P to S_P then the decision regions as defined by $M(\mathbf{x}, S_i)$ are invariant to the SNR.*

Proof: If P_i is a high SNR decision boundary between partition sets S_1 and S_2, and $\mathbf{r}$ lies in P_i then by definition $f(d^2(\mathbf{r}, S_1)) = f(d^2(\mathbf{r}, S_2))$ for the values of SNR in question. If S_P reflects onto S_P through the plane P_i then S_1 maps onto S_2 and vice versa and clearly $d^2(\mathbf{r}, S_1) = d^2(\mathbf{r}, S_2)$ which implies $f(d^2(\mathbf{r}, S_1))$ equals $f(d^2(\mathbf{r}, S_2))$ for all values of SNR. Since S_P maps onto S_P through every decision boundary plane P_i, none of the decision regions are a function of the SNR.

For example, all partitions of figure 5.12 reflect about all the decision boundary lines to map onto themselves. The example of figure 5.11(a) does not exhibit this property, and as shown by figures 5.11(b) and 5.11(c) the ML decision regions vary with the SNR.

5.6.7 Error performance of uncoded transmission over partitions.

Given the partition set S_i has been selected for transmission, the probability of the decoder selecting a set S_j as the output is given by the average over the probabilities of error of the points in S_i and equals

$$p(S_i \to S_j) = \frac{1}{|S_i|} \sum_{s_i \in S_i} p(s_i \to S_j) \tag{5.75}$$

assuming each $s_i \in S_i$ is equiprobable. The probability $p(s_i \to S_j)$ is the probability of selecting the set S_j as the output, given the point s_i has been transmitted. The

error performance of equation (5.75) is dominated by the highest term $p(s_i \rightarrow S_j)$ in the sum and corresponds to the point in the set S_i with the *worst* distance properties to the points in the set S_j. For example for the partition set of figure 5.11(a) denoted by the crosses, the decoder is more likely to make an incorrect decision when the point in the lower left is transmitted compared to the point in the top right quadrant. This is because the point in the lower left quadrant has *four* neighbours, whilst the point in the top right quadrant only has one neighbour. For the Rayleigh fading channel we may use the high SNR union bound of equation (3.25) as a loose bound on the probability of selecting set S_j given signal point s_i in the set S_i was transmitted

$$p(s_i \rightarrow S_j) \leq \sum_{l=l_0}^{\infty} \rho_l \left(\frac{4}{E_s/N_0} \right)^l \tag{5.76}$$

where ρ_l is a weighted multiplicity term given by

$$\rho_l = \sum_{s_j \in \mathcal{S}_l} \frac{1}{d_p^2(s_i, s_j)} \tag{5.77}$$

The set $\mathcal{S}_l$ is the set of points in S_j at a Hamming distance of l away from the point s_i, and $d_p^2(s_i, s_j)$ is the product distance between the points s_i and s_j. At a high SNR the probability of equation (5.76) is dominated by the term for $l = l_0$. We will associate a performance measure between two partitions S_i and S_j consisting of two parameters; a minimum Hamming distance

$$d_H(S_i, S_j) = \min_{s_i \in S_i, s_j \in S_j} d_H(s_i, s_j) \tag{5.78}$$

and a weighted multiplicity term

$$\rho_l(S_i, S_j) = \sum_{s_j \in \mathcal{S}_l} \frac{1}{d_p^2(s_i, s_j)} \tag{5.79}$$

where $l = d_H(S_i, S_j)$. These two parameters combined characterise the error performance of the pair of sets S_i and S_j.

5.7 Codes over GU partitions.

Consider a partition S_P and an associated generating group $G(S_P)$. A group code [69] C over G is a set of sequences, drawn from the group G, on a discrete index I which forms a group under component-wise group operation. The code is a subgroup

of G^I and therefore, from theorem 13 forms a geometrically uniform partition. The properties of the code may therefore be evaluated by considering only one code sequence.

5.7.1 Transmission of codes over partitions.

The code as defined above is a set of sequences, the elements of which are partitions of a signal constellation, where each partition contains a set of points. In analogy to an uncoded partition, we define our coded transmission system as depicted in figure 5.14. From an input bit stream, the partition encoder generates the code C over the set S_P, and passes the output code set S_i to the signal point selector. The signal point selector randomly selects a point from S_i, to transmit across the channel.

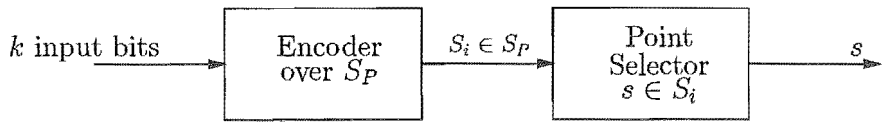


Figure 5.14: The encoder structure for a code over partitions.

5.7.2 Code distance properties.

In section 5.6.7 we developed the distance properties determining the probability of error of uncoded transmission of sets of points. We extend these properties to codes over partitions by simply viewing the code as a large uncoded system. Consider a group code C defined over a group $G(S_P)$. Each code word may be written as a sequence of symbols drawn from G. From the property of geometric uniformity we need only consider the error performance of one codeword. We will use the "all-zero" codeword $\mathbf{c}_0$, the codeword corresponding to the identity element of C, i.e. the sequence of identity elements from G.

$$\mathbf{c}_0 = \{\underbrace{S_0, S_0, \ldots, S_0}_{L}\} \tag{5.80}$$

where S_0 is a partition set from S_P. The codeword $\mathbf{c}_0$ is a sequence of L sets of points S_0 and may also be written as S_0^L, a set of $|S_0|^L$ points. Similarly every code sequence $\mathbf{c}_j$ of C maps to a set of $|S_0|^L$ points and the entire code C can be viewed

simply as a larger partition $C(S_P)$ and the performance properties can be analysed identically to that of an uncoded system. To determine the properties of $C(S_P)$ in terms of the group code C and the properties of S_P, consider another codeword $\mathbf{c}_i$ in C .

$$\mathbf{c}_i = \{S_{b_1}, S_{b_2}, \ldots, \ldots, S_{b_L}\} \tag{5.81}$$

where b_k are indices of the codeword. It is straight-forward to show that the Hamming distance between $\mathbf{c}_0$ and $\mathbf{c}_i$ is given by

$$d_H(\mathbf{c}_0, \mathbf{c}_i) = \sum_{k=1}^{L} d_H(S_0, S_{b_k}) \tag{5.82}$$

and the weighted squared product distance by

$$d^2_{wp}(\mathbf{c}_0, \mathbf{c}_i) = \prod_{\substack{k=1 \\ b_k \neq 0}}^{L} d^2_{wp}(S_0, S_{b_k}) \tag{5.83}$$

5.7.3 Decoding of codes over partitions.

Codes defined over partitions may be decoded using the ML decoding metric defined in section 5.6.1 or simply a minimum Euclidean distance metric. If the code is defined as a trellis code, it can be decoded using the Viterbi algorithm [38]. Since each branch label of the trellis code represents a set of a partition, it is necessary to compute the minimum distance to each set of the partition. For partitions with few elements this is best achieved exhaustively, while for large partitions it is better to apply the algorithm of section 5.6.2 for each partition set. More will be said on this subject in chapter 6 on multi-level codes based on GU partitions.

5.8 Summary.

Conventional codes are defined over points in signal space. With a different approach, the technique of multi-level coding, a signal set written as a hierarchy of subsets and codes are defined over *sets* of points. The error performance of each code of the scheme is now not only determined by the distance properties of the code, but also by the properties of the sets of points over which it is defined. We have extended the definitions of geometric uniformity introduced in chapter 4, to

sets of points. The sets of points when viewed as units can be constructed to exhibit the same types of properties as GU constellations. Therefore the error evaluation of communication systems based on the transmission of a point selected from a set of points can be calculated by considering only the distance properties of one set of points. These definitions form the foundation for the construction of good multi-level codes which are discussed in chapter 6.

Chapter 6

Multi-level Codes.

Multi-level codes are a means of constructing higher complexity signal-space codes with good distance properties. The technique is based on partitioning a signal constellation into a multi-level hierarchy and defining codes over each level. The advantage of this technique lies in the ability to construct a staged decoder with a complexity far less than a ML decoder but with a similar asymptotic probability of error performance. In the work on multi-level codes by Imai and Hirakawa [35] and Sayegh [61], the signal set is partitioned into a binary chain and binary codes are defined over each level. Tanner [72] formally related the minimum Hamming distance of the component codes to the minimum Euclidean distance of the multi-level scheme. Pottie and Taylor [57] generalised the concept of multi-level constructions to include non-binary partitions of multi-dimensional signal constellations and placed no restrictions on the type of component codes. Indeed the component codes may again be multi-level codes. Calderbank's paper shows how to calculate minimum squared distances and path multiplicities in terms of the norms and multiplicities of the different cosets [10]. All of these works and others [44],[76],[71],[79] are for the Gaussian channel where the code performance criteria is the minimum squared Euclidean distance of the code. Subsequent papers describe multi-level codes designed specifically for the Rayleigh fading channel, where the Hamming distance of the code is the most important [64],[65], [80].

In this chapter we construct multi-level codes for the Rayleigh fading channel based on the geometrically uniform partitions developed in chapter 5. Each component code is defined as a group code over the generating group of a GU partition

and consequently it has all the desirable properties of a GU code, including reduced decoding complexity at each code level. Recently related work has been published in the literature [5],[58]. We begin by describing the general structure of a multi-level encoder.

6.1 General multi-level code structure.

In a generalised multi-level construction, a signal constellation S_l is partitioned into a partition chain written as $S_l/S_{l-1}/\cdots/S_0$. Each S_{l-1} is a subset of the set S_i directly above it in the chain such that it divides S_i exactly into S_{i-1} and its cosubsets. The order of the partitioning at level i is $m_i = |S_i|/|S_{i-1}|$. The elements of the set formed by the partitioning of S_i into S_{i-1} and its cosubsets may be labelled by a set of labels $\mathcal{A}_i$, and we write $S_i/S_{i-1} \rightarrow \mathcal{A}_i$, that is S_{i-1} and its cosubsets map onto the elements of $\mathcal{A}_i$. The set $\mathcal{A}_i$ of labels are elements of a discrete alphabet over which a component code C_i can be defined. If we define a code C_i over every label set $\mathcal{A}_i$ of the partition chain, then the combined hierarchy consisting of the partition chain $S_l/S_{l-1}/\cdots/S_0$, the label sets $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_l$, and the set of codes $C_1, C_2, \ldots, C_l$ forms a multi-level code. The general structure is shown in figure 6.1. Each code

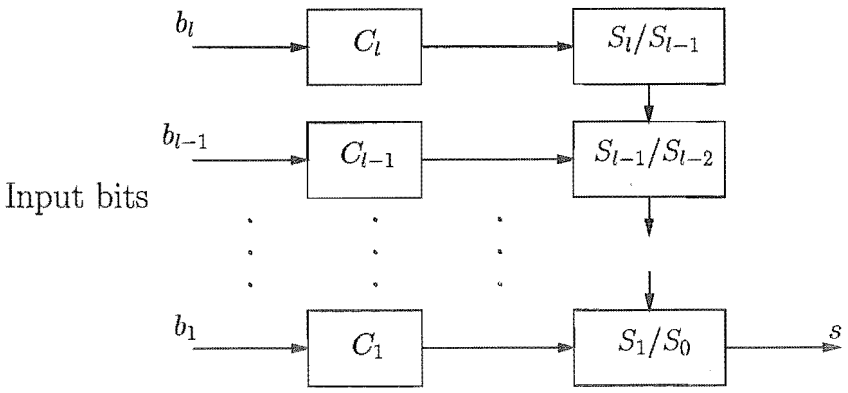


Figure 6.1: General encoder structure for a multi-level code.

C_i accepts b_i input bits and outputs $|S_i|/|S_{i-1}|$ bits. The output of the encoder generating code C_l selects a cosubset of S_l/S_{l-1}. The next encoder C_{l-1} similarly selects a cosubset of S_{l-1}/S_{l-2} etc. until finally the code C_1 selects a single point for transmission. We may associate an overall code C with the l-level construction

of figure 6.1 written as

$$C = C_1 * C_2 * \ldots * C_l \tag{6.1}$$

This code is referred to as the *multi-level* code associated with the partition chain $S_l/S_{l-1}/\cdots/S_0$ and the component codes $C_1, C_2, \ldots, C_l$.

6.1.1 Decoding of multi-level codes over GU partitions.

Multi-level codes are most easily decoded by a staged decoder as shown in figure 6.2. Each decoder D_i in the figure corresponds to the component code C_i. The staged decoder operates in a sequential manner. First the decoder D_l makes a decision on the code C_l and outputs the corresponding b_l data bits and passes the coset decision information to D_{l-1}. The decoder D_{l-1} operates in a similar manner and the process continues down the chain until the received sequence is completely decoded. Since every decoder in the staged decoder requires the decision information

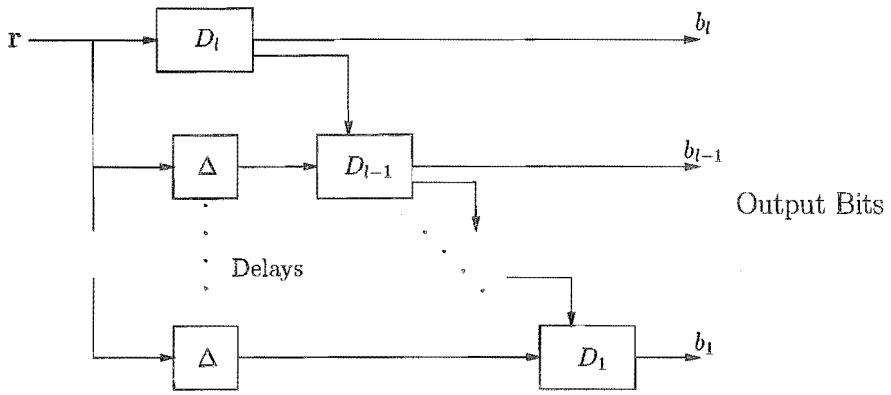


Figure 6.2: General structure of a staged decoder for a multi-level code.

from the previous decoder in the hierarchy, there is a decoding delay proportional to the number of decoding stages. If $\Delta(D_i)$ is the average delay associated with decoder D_i and the decoding delays are statistically independent then the average delay of the staged decoder D is given by

$$\Delta(D) = \sum_{i=1}^{l} \Delta(D_i) \tag{6.2}$$

The fact that each decoder depends on the correctness of the decision information of the previous decoder can cause error propagation in the decoding process. Techniques such as an intralevel interleaver and iterative decoding may be employed to

combat these effects [65]. We will not address these issues and simply assume that the overall probability of error of the multi-level code is dominated by the worst code in the hierarchy.

6.2 Design of multi-level codes.

The design of good multi-level codes is not a straight-forward exhaustive code search as is typical for trellis codes where the only parameter varied is the number of code states. For multi-level codes the designer has the freedom to select the underlying constellation, the partition chain, the number of levels in the chain, and the most suitable codes and code rates for each level. Clearly the number of coding parameters that may be varied is large and this allows for a great number of possible design trade-offs. Typically the main tradeoff of interest when designing multi-level codes is the coding gain versus the decoding complexity. For the trellis codes of chapter 4 designed for the Rayleigh fading channel, each increase in decoder memory led to a doubling in decoding complexity and provided a diminishing increase in coding gain. In the next sections we examine a class of multi-level codes based on the combination of geometrically uniform partitions and geometrically uniform trellis codes. This class of multi-level codes have a decoding complexity far less than that of similarly performing trellis codes.

6.3 Multi-level codes over GU partitions.

We now define the structure of the multi-level codes of our study. Chapter 5 introduced the concept of geometric uniformity over partitions. We defined a general partition tree in section 5.5 and showed that the partitions over each level can be made geometrically uniform by one of two constructions, namely a construction over a normal subgroup chain of a generating group of a geometrically uniform constellations as described in section 5.5.1, and by an iterative construction over geometrically uniform constellations as described in section 5.5.2. In section 5.7 we defined the concept of uncoded transmission over partitions and extended this transmission format to geometrically uniform codes over partitions. We showed that the distance properties controlling the error performance, namely the Hamming distance

and product distance, are determined by the code itself and the minimum distance properties between partitions. We define a multi-level code based on a GU partition tree as follows.

Definition 34 *Given a GU partition tree S_P as defined in section 5.5 and written as the chain $S_l/S_{l-1}/\cdots/S_0$ with associated generating groups Q_i such that*

$$S_i = \{q_i S_{i-1} : q_i \in Q_i\} \tag{6.3}$$

then a multi-level code is constructed by defining a group code C_i over the elements of Q_i.

The code C_i may be any group code, such as a linear block code, or a trellis code. In our code designs we have selected each C_i to be a multi-dimensional trellis code over Q_i. The general structure of the multi-level coding scheme of our study is shown in figure 6.3. At every time step, each encoder C_i accepts b_i bits and outputs a symbol

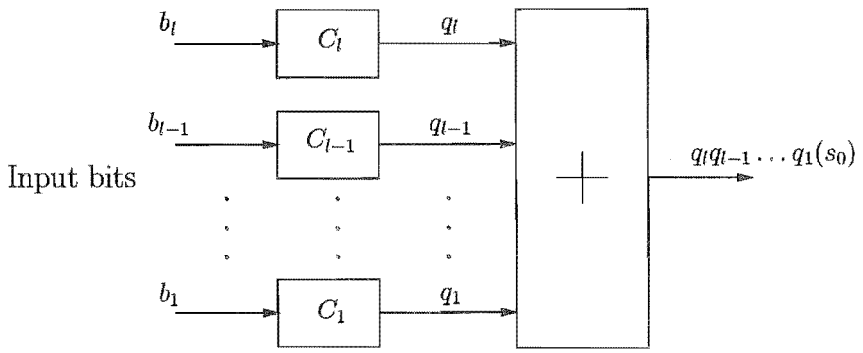


Figure 6.3: General encoder structure for a multi-level code over GU partitions.

q_i from the generating group Q_i of the partition. The mappings are combined by the summer and operate on the signal point s_0 to produce the output signal point $q_l q_{l-1} \cdots q_1(s_0)$.

6.3.1 The component codes.

Although a component code may be any group code over the label group $\mathcal{A}_i$, we have chosen to use a trellis code on each level. This selection was made to make use of the GU trellis codes already discovered in chapter 4. In practice it may be better to use short block codes to reduce the overall decoding delay associated with trellis

codes. The trellis codes are designed in a manner similar to the GU trellis codes of chapter 4. In analogy to a generating group G of a constellation S, the group $\mathcal{A}_i$ through its isomorphic mapping corresponds to a generating group of the underlying partition. In general we construct code generators over the group $(\mathcal{A}_i)^{p_i}$, where p_i is an integer exponent. The general encoder structure of a trellis code C_i over $(\mathcal{A}_i)^{p_i}$ is identical to that shown in figure 4.3 of chapter 4, except that the code generator elements are selected from the label group $\mathcal{A}_i$ instead of a generating group G. Each encoder accepts b_i bits and outputs a sequence of length Lp_i corresponding to a branch sequence. The rate of the code C_i in bits per two-dimensional symbol is given by

$$r_i = \frac{b_i}{Lp_i} \tag{6.4}$$

The ratio $\frac{b_i}{p_i}$ is limited by the size of the label group $\mathcal{A}_i$ to

$$\frac{b_i}{p_i} \leq \log_2 |\mathcal{A}_i| \tag{6.5}$$

The rate r of the multi-level code C, composed of the component codes $C_1, C_2, \ldots, C_l$, in bits per two dimensions is

$$r = \frac{1}{L} \sum_{i=1}^{l} \frac{b_i}{p_i} \tag{6.6}$$

The distance properties of the component codes are determined by the distance properties of code C_i and the label group $\mathcal{A}_i$ over which it is defined. This relationship has been defined in section 5.7.2 of chapter 5.

6.3.2 Decoding of component codes.

The component codes of our multi-level construction are GU trellis codes. We have shown in section 5.6.5 of chapter 5 that minimum squared Euclidean distance is a sufficient metric for the decoding of codes over partitions. The Viterbi algorithm (VA) [38] is an efficient algorithm for the decoding of trellis codes, however before the VA can be applied to a trellis code over partitions we must first compute the minimum distance metric to each element of the label group $\mathcal{A}_i$. For example, consider decoding the code described by the trellis of figure 6.4. This simple code consists of two states, each of which is connected to every other by a pair of parallel

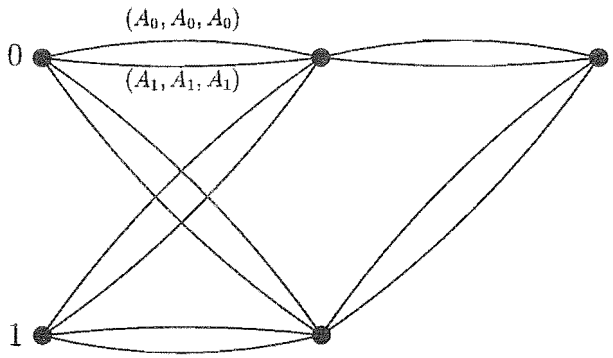


Figure 6.4: An example trellis diagram to illustrate the decoding of partition sets.

branches. Each branch has associated with it a sequence of three symbols from the label group $\mathcal{A}_i$. The label group contains two elements A_0 and A_1, each representing a set of points. To extend the branch metric entering state 0 from state 0 we must first compute the minimum squared Euclidean distance to each of the elements A_0 and A_1 at each of the three time slots. Once these distances have been computed and stored, we may compute the metric for each branch by summing the three appropriate pre-computed distance metrics and proceeding in the normal manner. To compute the metric to A_0 (or A_1) we must calculate the distance from the received vector $\mathbf{r}$ to every element in A_0, i.e.

$$m(\mathbf{r}, A_0) = \min_{a \in A_0} ||\mathbf{r} - a||^2 \tag{6.7}$$

For a large $|A_0|$ an exhaustive comparison calculation is expensive. Since A_0 is a linear group code, we may apply the same technique used for the reduced complexity decoding of the parallel group as presented in section 4.7.3, i.e. we write A_0 as a trellis, and decode with the VA.

6.3.3 Decoding complexity.

To compare the performance of multi-level codes to conventional trellis codes we need a decoding complexity measure. In chapter 4 we modified the *normalised branch complexity* introduced by Wei and others to include parallel transitions to measure the decoding complexity of geometrically uniform trellis codes [77]. The normalised branch complexity calculates the average number of metric computations and comparisons required per decoded two-dimensional symbol. A component code C_i of the multi-level code C is defined over the label group $\mathcal{A}_i$ and as such an extra

decoding step is required. This step is to compute the minimum distance to each element of the label group $\mathcal{A}_i$ as illustrated by example in section 6.3.2. Therefore the three steps in decoding a component trellis code are:

1. Compute the distance metric of the received vector to each element of $\mathcal{A}_i$ and retain the minimum distance. The normalised complexity of this step is given by

$$\xi_A(C_i) = |\mathcal{A}_i|\xi(A_{i_0})/L \tag{6.8}$$

where $\xi(A_{i_0})$ is the complexity of computing the minimum distance to one element in $\mathcal{A}_i$, and $|\mathcal{A}_i|$ is the number of elements in $\mathcal{A}_i$.

2. Compute the distance to the parallel group P_i and its cosets of the trellis.

$$\xi_P(C_i) = |\mathcal{P}_i|\xi(P_i)/Lp_i \tag{6.9}$$

where $\xi(P_i)$ is the complexity of computing the minimum distance to the parallel group P_i, and $|\mathcal{P}_i|$ is the number of cosets of P_i.

3. Compute the trellis branch metrics.

$$\xi_T(C_i) = 2^{v_i+b_{ci}}/Lp_i \tag{6.10}$$

where 2^{v_i} is the number of states of the trellis and b_{ci} is the number of coded bits.

The total normalised decoding complexity of the code component code C_i per symbol is the sum of the decoding complexities of the above three steps. Thus

$$\xi(C_i) = \xi_A(C_i) + \xi_P(C_i) + \xi_T(C_i). \tag{6.11}$$

The decoding complexity of the multi-level code C is the sum of the complexities of the l component codes

$$\xi(C) = \sum_{i=1}^{l} \xi(C_i). \tag{6.12}$$

This measure of decoding complexity is only a guide to the true complexity required for decoding. It does not take into account the decoder memory requirements, possible intra-level interleaving, or any iterative decoding. However it is useful as a *relative* indication of complexity for the comparison of code alternatives.

6.3.4 Design steps.

The steps of the design process of multi-level codes over GU partitions can be split into two parts. First we need to select a partition chain of a multi-dimensional signal constellation. If we construct a partition chain based on the technique of constructing a normal subgroup chain of a generating group of $L \times M$-PSK as described in section 5.5.1 of chapter 4 then the following steps are necessary.

1. Select the parameters L and M for the $L \times M$-PSK constellation S.

2. Select a generating group $G(S)$ of S.

3. Construct a normal subgroup chain $G_0 \lhd G_1 \lhd \cdots \lhd G_l$ of G with desirable distance properties. The orders of the partitioning need to be selected.

4. Compute the quotient groups $Q_i = G_i/G_{i-1}$, and define an isomorphism mapping Q_i to $\mathcal{A}_i$.

Given the set of label groups $\mathcal{A}_i$ and the associated distance properties we need to design the component codes C_i.

1. For the component code C_i over Q_i select an exponent p_i.

2. Construct the group $(\mathcal{A}_i)^{p_i}$.

3. Select the number of input bits b_i to the component trellis encoders such that the sum $r = \frac{1}{L}\sum_{i=1}^{l} \frac{b_i}{p_i}$ gives the desired overall data rate r.

4. Select the number of code states v_i for each encoder to give the design minimum Hamming distance.

5. Select the code generators of the trellis code.

With the large number of design decisions and possibilities, none of which are clearly easy to make, the design of multi-level codes is somewhat of an ad-hoc procedure. In the next section we examine in detail two case studies of multi-level code designs to familiarise the reader more with the material presented so far.

6.4 Case studies.

To clarify the previous sections on the decoding of multi-level codes over geometrically uniform partitions we will examine some case studies and analyse them in detail.

6.4.1 Study 1: A multi-level code over 8-PSK.

In this example we consider a simple multi-level code over the constellation 1×8-PSK. The case of a three level partitioning has previously been studied by Seshadri and Sundberg [65] for the Rayleigh fading channel. As an alternative we consider a two level partitioning scheme. The code will be designed for a spectral efficiency of 2 bits/symbol to allow for the easy comparison with the trellis codes of chapter 4 over 8-PSK constellations. We will construct a partitioning of $S = 8$-PSK by selecting a normal subgroup chain of a generating group of 8-PSK. We consider the generating group

$$G(S) = \{r_0, r_2, r_4, r_6, v_1, v_3, v_5, v_7\} \tag{6.13}$$

isomorphic to D_4. The normal subgroup chain $G_0 \lhd G_1 \lhd G_2$, where $G_0 = \{r_0\}$, $G_1 = \{r_0, r_4\}$ and $G_2 = G$ partitions S into the tree shown in figure 6.5. The orders

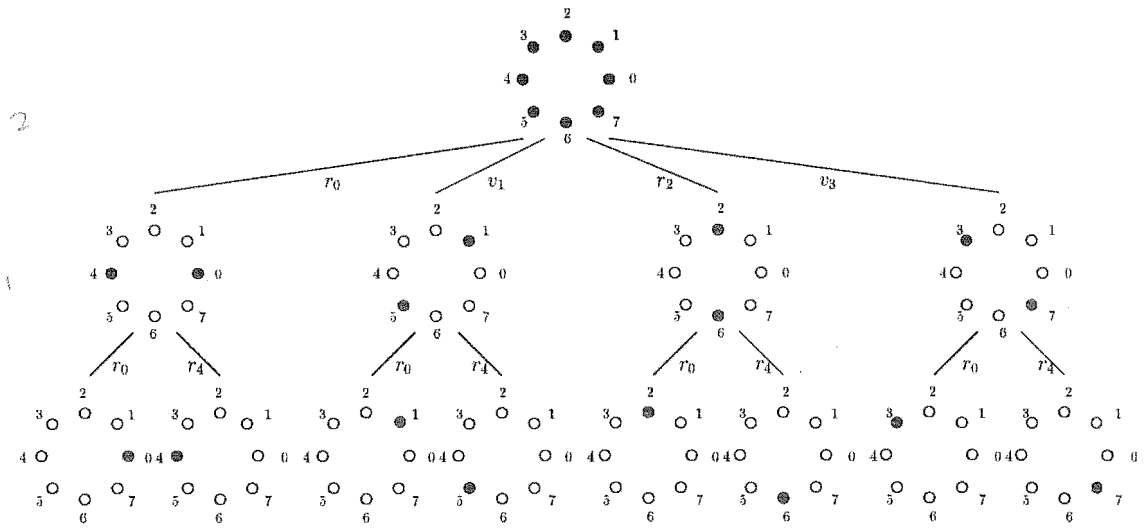


Figure 6.5: The partition tree of 8-PSK generated by the chain $G_0 < G_1 < G_2$.

of the partition are $m_1 = |G_1|/|G_0|$ and $m_2 = |G_2|/|G_1| = 4$. The quotient groups

over which the component codes are defined are

$$Q_1 = G_1/G_0 = \{\{r_0\}, \{r_4\}\} \tag{6.14}$$

$$Q_2 = G_2/G_1 = \{\{r_0, r_4\}, \{r_2, r_6\}, \{v_1, v_5\}, \{v_3, v_7\}\} \tag{6.15}$$

The group Q_1 is isomorphic to Z_2 and Q_2 is isomorphic to $(Z_2)^2$. We define the label group $\mathcal{A}_1$ as $\mathcal{A}_1 = \{0, 1\}$ and associate the following mappings between Q_1 and $\mathcal{A}_1$:

$$\{r_0\} \rightarrow 0 \tag{6.16}$$

$$\{r_4\} \rightarrow 1 \tag{6.17}$$

We define the label group $\mathcal{A}_2$ as $\mathcal{A}_2 = \{00, 01, 10, 11\}$ and associate the mappings between Q_2 and $\mathcal{A}_2$:

$$\{r_0, r_4\} \rightarrow 00 \tag{6.18}$$

$$\{r_2, r_6\} \rightarrow 01 \tag{6.19}$$

$$\{v_1, v_5\} \rightarrow 10 \tag{6.20}$$

$$\{v_3, v_7\} \rightarrow 11 \tag{6.21}$$

From theorem 15 of chapter 5 the distance properties of a GU partition are characterised by the distance properties between any element (say the identity element) and the other elements in the group. The group $\mathcal{A}_1$ is of order two and we need only the distance properties between the elements 0 and 1. Note that the symbols 0 and 1 are labels of $\mathcal{A}_1$ corresponding to the elements in Q_1 and the corresponding symbols from Q_1 need to be used to compute the correct distance properties. We have for $\mathcal{A}_1$:

$$d_H(0, 1) = d_H(\{r_0\}, \{r_4\}) = 1, \tag{6.22}$$

$$d_p^2(0, 1) = d_p^2(\{r_0\}, \{r_4\}) = 4.0 \tag{6.23}$$

and

$$N(0, 1) = N(\{r_0\}, \{r_4\}) = 1 \tag{6.24}$$

For the group $\mathcal{A}_2$ there are four elements each containing two points. We need to know the distance from the identity element to the other three elements:

$$d_H(00, 01) = d_H(\{r_0, r_4\}, \{r_2, r_6\}) = 1 \tag{6.25}$$

$$d_H(00, 10) = d_H(\{r_0, r_4\}, \{v_1, v_5\}) = 1 \tag{6.26}$$

$$d_H(00, 11) = d_H(\{r_0, r_4\}, \{v_3, v_7\}) = 1, \tag{6.27}$$

$$d_p^2(00,01) = d_p^2(\{r_0, r_4\}, \{r_2, r_6\}) = 2.0 \quad (6.28)$$

$$d_p^2(00,10) = d_p^2(\{r_0, r_4\}, \{v_1, v_5\}) = 0.587 \quad (6.29)$$

$$d_p^2(00,11) = d_p^2(\{r_0, r_4\}, \{v_3, v_7\}) = 0.587 \quad (6.30)$$

and

$$N(00,01) = N(\{r_0, r_4\}, \{r_2, r_6\}) = 2 \quad (6.31)$$

$$N(00,10) = N(\{r_0, r_4\}, \{v_1, v_5\}) = 1 \quad (6.32)$$

$$N(00,11) = N(\{r_0, r_4\}, \{v_3, v_7\}) = 1 \quad (6.33)$$

We have summarised the partition chain distance properties in table 6.1. Note that

Partition I of 1 × 8-PSK							
i	G_i/G_{i-1}	m_i	Coset	Map to $\mathcal{A}_i$	d_H	d_p^2	N
1	Z_2	2	r_4	1	1	4.0	1
2	$(Z_2)^2$	4	r_2	01	1	2.0	2
			v_1	10	1	0.587	1
			v_3	11	1	0.587	1

Table 6.1: Summary of the distance properties of the partitioning of 8-PSK corresponding to the chain $G_0 \lhd G_1 \lhd G_2$.

the distance properties of the partition of level one are better than those of level two. Now that the partition chain has been established we need to construct codes over the label groups $\mathcal{A}_1$ and $\mathcal{A}_2$. We desire an overall data rate of 2 bits/symbol hence

$$r = \frac{b_1}{p_1} + \frac{b_2}{p_2} = 2 \quad (6.34)$$

and from equation (6.5) we must have $\frac{b_i}{r_i} \leq \log_2 |\mathcal{A}_i|$, hence by selecting $b_1 = 2$, $p_1 = 3$, $b_2 = 4$ and $p_2 = 3$, we achieve the desired data rate. The selected encoder for the code C_1 over $\mathcal{A}_1$ is shown in figure 6.6. It is a rate 2/3 encoder with two memory elements, hence $v_1 = 2$. The trellis diagram corresponding to the encoder of figure 6.6 is shown in figure 6.7. The code C_1 over $(\mathcal{A}_1)^3$ of figure 6.7 has a minimum Hamming distance of 3, the path of which has been highlighted. Mapping $\mathcal{A}_1$ through the isomorphism back to Q_1 gives the following distance properties for

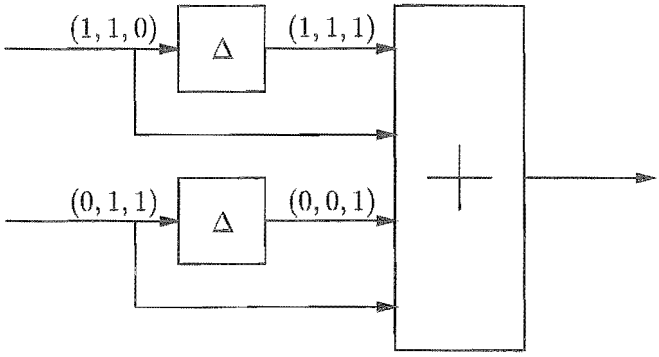


Figure 6.6: Binary trellis encoder over $(\mathcal{A}_1)^3$.

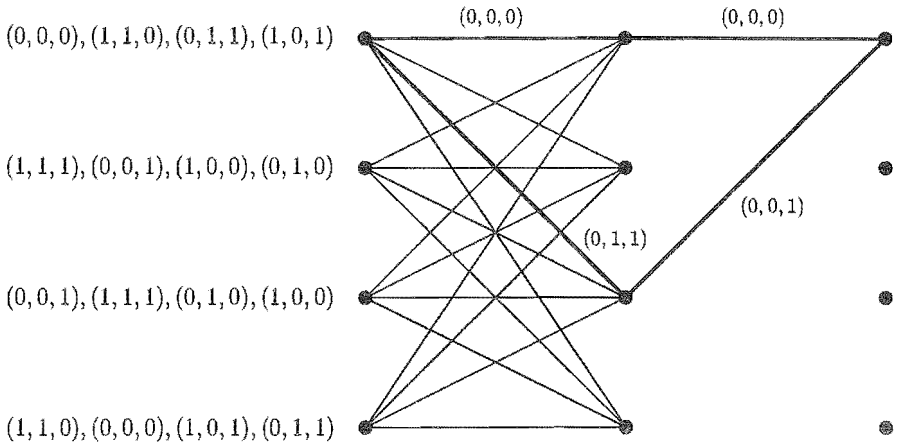


Figure 6.7: Trellis diagram corresponding to the encoder of figure 6.6.

C_1:

$$d_H(C_1) = 3 \tag{6.35}$$

$$d_p^2(C_1) = 4.0^3 = 64.00 \tag{6.36}$$

$$N(C_1) = 1 \tag{6.37}$$

The decoding complexity of code C_1 is as follows. The elements of the label group $\mathcal{A}_1$ correspond to single points and no distance computation to a subset is required hence $\xi_A(C_1) = 0$. The encoder over $\mathcal{A}_1$ does not have parallel paths, hence $\xi_P(C_1) = 0$. The trellis encoder for code C_1 consists of 4 states, each with 4 transitions giving a trellis complexity from equation (6.10) of $\xi_T(C_1) = 16/3$. The total normalised

complexity of decoder C_1 is therefore given by

$$\begin{aligned} \xi(C_1) &= \xi_A(C_1) + \xi_P(C_1) + \xi_T(C_1) \\ &= 0 + 0 + 16/3 \\ &= 16/3 \end{aligned} \tag{6.38}$$

Now we consider a code over $\mathcal{A}_2$. The group $\mathcal{A}_2$ is of order four and is isomorphic to $(Z_2)^2$. We have constructed an eight-state trellis code over $(A_2)^3$. The encoder is shown in figure 6.8. The trellis diagram corresponding to the encoder of code C_2

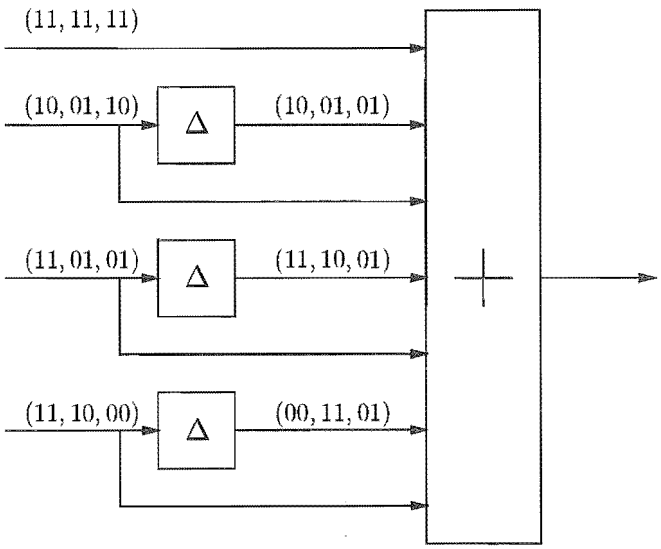


Figure 6.8: Trellis encoder for code C_2 over the label group $(\mathcal{A}_2)^3$.

of figure 6.8 is shown in figure 6.9. The encoder accepts four bits for transmission, three of which are encoded and one is uncoded. Therefore there are two parallel transitions in the trellis diagram corresponding to the uncoded bit of the system. The code C_2 of figure 6.9 has a minimum Hamming distance of 3, corresponding to the parallel branches. Mapping $\mathcal{A}_2$ through the isomorphism back to Q_2 gives the following distance properties for C_2:

$$d_H(C_2) = 3 \tag{6.39}$$

$$d_p^2(C_2) = 2.0^3 = 8.00 \tag{6.40}$$

$$N(C_2) = 2^3 = 8 \tag{6.41}$$

The decoding complexity of code C_2 is as follows. The partition group Q_2 consists of four sets containing two point each. Hence $4 \times 2 = 8$ distance computations are

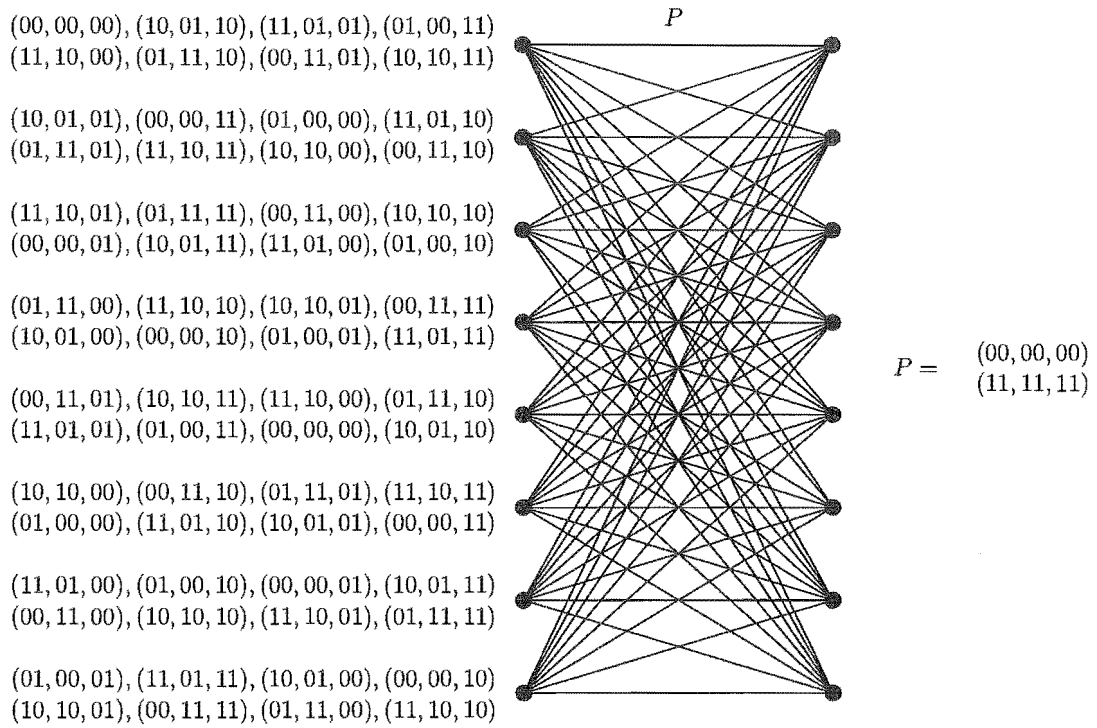


Figure 6.9: Trellis diagram corresponding to the encoder of code C_2 of figure 6.8.

required to compute the minimum distance to each subset and we have $\xi_A(C_1) = 8$. The encoder over $\mathcal{A}_2$ has two parallel paths, and there are 31 distinct cosets of the parallel group. Hence $\xi_P(C_2) = (2 \times 32)/3 \approx 21.3$. The trellis C_2 consists of 8 states, each with 8 transitions and the trellis complexity is given by $\xi_T(C_2) = (8 \times 8)/3$. The total normalised complexity of decoder C_2 is therefore given by

$$\begin{aligned} \xi(C_2) &= \xi_A(C_2) + \xi_P(C_2) + \xi_T(C_2) \\ &= 8 + 64/3 + 64/3 \\ &\approx 50.7 \end{aligned} \tag{6.42}$$

The total decoding complexity of the code $C = C_1 * C_2$ is

$$\begin{aligned} \xi(C) &= \xi(C_1) + \xi(C_2) \\ &= 56 \end{aligned} \tag{6.43}$$

Table 6.2 is a summary of the multi-level code construction. The error performance of the multi-level code C is dominated by the worst code in the construction. Clearly this is the component code C_2 which has a product distance of 8 and multiplicity 8, compared the product distance of 64 and multiplicity 1 of code C_1. In figure 6.10 we

Code over S = 8-PSK, partition I													
i	m_i	$\mathcal{A}_i$	p_i	b_i	r_i	v_i	d_H	d_p^2	N	$\xi_A(C_i)$	$\xi_P(C_i)$	$\xi_t(C_i)$	$\xi(C_i)$
1	2	Z_2	3	2	$\frac{2}{3}$	2	3	64.0	1	0	0	16/3	16/3
2	4	$(Z_2)^2$	3	4	$\frac{4}{3}$	3	3	8.0	8	8	64/3	64/3	50.7
bits/symbol=					2.0	Total complexity							56

Table 6.2: Summary of the properties of the multi-level code of case 1.

have plotted the error curves of the two component codes C_1 and C_2 as well as that of a multi-dimensional geometrically uniform trellis code C' with $v = 4$ described in table 4.10 of chapter 4. The trellis encoder has 16 states and is over the constellation $S = 2 \times$ 8-PSK. It has a Hamming distance of 3, a minimum product distance of $d_p^2 = 8$, and a weighted product distance of $d_{pw}^2 = 1.59$. The code properties are similar to those of code C_2. However the normalised complexity of the trellis code from table 4.10 is higher,

$$\xi(C') = 128 \tag{6.44}$$

and is about 2.5 times more complex to decode than the multi-level code. We see that the multi-level code operates at a similar performance to the trellis code but with a decoding complexity that is significantly lower.

6.4.2 Study 2: A multi-level code over $2 \times$ 8-PSK.

For this case we study a multi-level code over the constellation $2 \times$ 8-PSK. This constellation is more interesting in that the number of design possibilities and trade-offs are far greater. The constellation may be partitioned to give different Hamming distances at different levels. Therefore the code rate at the higher levels can be increased, while the code rate at the lower levels is decreased, making it possible to compensate for the reduced distance properties. In this example we partition the constellation $S = 2 \times$ 8-PSK into three levels. The partitioning is based on the generating group $G(S) \simeq (D_8)^2$. We have selected the normal subgroup chain

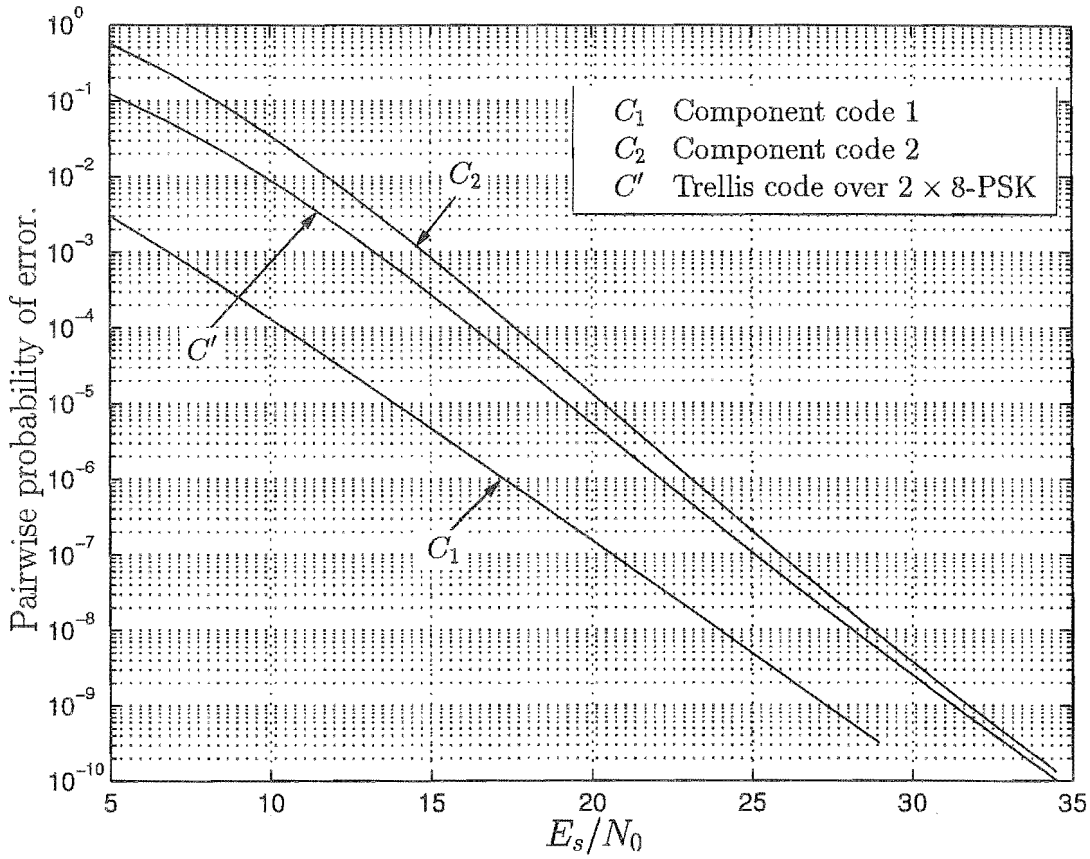


Figure 6.10: The performance of the component codes C_1 and C_2 of the multi-level code of study 1. Plotted for comparison is a multi-dimensional trellis code C' with similar distance properties.

$G_0 \lhd G_1 \lhd G_2 \lhd G_3$, where

$$G_0 = \{(r_0, r_0)\} \tag{6.45}$$

$$G_1 = \{(r_0, r_0), (r_4, r_0), (v_7, r_4), (v_3, r_4)\} \cdot G_0 \tag{6.46}$$

$$G_2 = \{(r_0, r_0), (r_4, r_4), (v_5, v_7), (v_1, v_3)\} \cdot G_1 \tag{6.47}$$

$$G_3 = \{(r_0, r_0), (r_4, r_2), (r_0, v_1), (r_4, v_3)\} \cdot G_2 \tag{6.48}$$

This selection of the partition chain partitions S into three four-way partitions, the distance properties of which are summarised in table 6.3. Each of the three quotient groups G_i/G_{i-1} is isomorphic to $(Z_2)^2$ and the isomorphic mappings to the labels groups $\mathcal{A}_i$ are included in table 6.3 in terms of the coset representatives. Note that the partition levels one and two both have two elements with a Hamming distance of $d_H = 2$. These higher Hamming distance elements can be exploited by the encoder to

Partition I of 2 × 8-PSK							
i	G_i/G_{i-1}	m_i	Coset rep.	Map to $\mathcal{A}_i$	d_H	d_p^2	N
1	$(Z_2)^2$	4	(r_4, r_0)	01	1	4.0	1
			(v_7, r_4)	10	2	2.34	1
			(v_3, r_4)	11	2	13.7	1
2	$(Z_2)^2$	4	(r_4, r_4)	01	2	0.343	1
			(v_5, v_7)	10	1	0.587	1
			(v_1, v_3)	11	2	1.17	2
3	$(Z_2)^2$	4	(r_4, r_2)	01	1	2.0	2
			(r_0, v_1)	10	1	0.587	1
			(r_4, v_3)	11	1	0.587	2

Table 6.3: Summary of the distance properties of a three level partitioning of 2 × 8-PSK.

improve upon the minimum Hamming distance otherwise possible. We have selected to construct each of the component codes C_i over $(\mathcal{A}_i)^2$, i.e. $p_i = 2$. To choose a rate of each encoder, we compute the overall rate of the code in bits per symbol

$$\begin{aligned} r &= \frac{b_1}{Lp_1} + \frac{b_2}{Lp_2} + \frac{b_3}{Lp_3} \\ &= \frac{b_1}{2 \cdot 2} + \frac{b_2}{2 \cdot 2} + \frac{b_3}{2 \cdot 2} \\ &= \frac{b_1 + b_2 + b_3}{4} \end{aligned} \tag{6.49}$$

To achieve an overall rate of 2 bits/symbol we need $b_1 + b_2 + b_3$ to equal 8, and hence we select $b_1 = 3$, $b_2 = 3$ and $b_3 = 2$. Note that the lower bit rate of code C_3 allows us to compensate for the worse distance properties of the underlying partition. We have selected the codes C_1 and C_2 to have a constraint length of 3 and the code C_3 to have a constraint length of 4. The encoders are summarised by the generator matrices

$$G(C_1) = \begin{pmatrix} (01,10) & (10,01) \\ (10,01) & (01,01) \\ (00,11) & (01,11) \end{pmatrix} \tag{6.50}$$

$$G(C_2) = \begin{pmatrix} (00,11) & (11,10) \\ (10,01) & (01,10) \\ (01,10) & (00,11) \end{pmatrix} \tag{6.51}$$

$$G(C_3) = \begin{pmatrix} (10,01) & (01,11) & (11,11) \\ (01,10) & (10,10) & (01,10) \end{pmatrix} \tag{6.52}$$

These trellis generators combined with the partition properties of table 6.3 gives the following distance properties for each of the three component codes:

$$\begin{aligned} d_H(C_1) &= 5 \quad d_p^2(C_1) = 128.0 \quad N(C_1) = 1 \\ d_H(C_2) &= 5 \quad d_p^2(C_2) = 0.235 \quad N(C_2) = 16 \\ d_H(C_3) &= 6 \quad d_p^2(C_3) = 0.471 \quad N(C_3) = 136 \end{aligned} \tag{6.53}$$

The decoding complexity of code C_1 is as follows. The partition group Q_1 consists of four sets containing one point each and $\xi_A(C_1) = 0$. The encoder C_1 has no parallel paths, hence $\xi_P(C_1) = 0$. The trellis defining C_1 has of 8 states, each with 8 transitions and the trellis complexity is given by $\xi_T(C_1) = (8 \times 8)/4 = 16$. The total normalised complexity of decoder C_1 is therefore given by

$$\begin{aligned} \xi(C_1) &= \xi_A(C_1) + \xi_P(C_1) + \xi_T(C_1) \\ &= 0 + 0 + 16 \\ &= 16 \end{aligned} \tag{6.54}$$

The partition group Q_2 consists of four sets containing four points each and as such $(4 \times 4)/2 = 8$ comparisons are required to compute the minimum distance to each subset. Hence we have $\xi_A(C_2) = 8$. The encoder C_2 has no parallel paths, hence $\xi_P(C_2) = 0$. The trellis over of C_2 has 8 states, each with 8 transitions and the trellis complexity is given by $\xi_T(C_2) = (8 \times 8)/4 = 16$. The total normalised complexity of decoder C_2 is therefore given by

$$\begin{aligned} \xi(C_2) &= \xi_A(C_2) + \xi_P(C_2) + \xi_T(C_2) \\ &= 8 + 0 + 16 \\ &= 24 \end{aligned} \tag{6.55}$$

The partition group Q_3 consists of four sets of sixteen points each. The number of comparisons required per symbol is $\xi_A(C_3) = (4 \times 16)/2 = 32$. There are no parallel paths in the code trellis of C_3 and $\xi_P(C_3) = 0$. The encoder consists of sixteen states, each with four branches and the trellis complexity is given by $\xi_T(C_3) = (4 \times 16)/4 =$

16. The total normalised complexity of decoder C_3 is therefore given by

$$\begin{aligned} \xi(C_3) &= \xi_A(C_3) + \xi_P(C_3) + \xi_T(C_3) \\ &= 32 + 0 + 16 \\ &= 48 \end{aligned} \tag{6.56}$$

and the total complexity of the multi-level code is

$$\begin{aligned} \xi(C) &= \xi(C_1) + \xi(C_2) + \xi(C_3) \\ &= 16 + 24 + 48 \\ &= 88 \end{aligned} \tag{6.57}$$

Table 6.4 is a summary of the multi-level code construction. This multi-level code

Code over $S = 2 \times 8$-PSK, partition I													
i	m_i	$\mathcal{A}_i$	p_i	b_i	r_i	v_i	d_H	d_p^2	N	$\xi_A(C_i)$	$\xi_P(C_i)$	$\xi_T(C_i)$	$\xi(C_i)$
1	4	$(Z_2)^2$	2	3	$\frac{3}{4}$	3	5	128.00	1	0	0	16	16
2	4	$(Z_2)^2$	2	3	$\frac{3}{4}$	3	5	0.235	16	8	0	16	24
3	4	$(Z_2)^2$	2	2	$\frac{1}{2}$	4	6	0.471	136	32	0	16	48
Code rate in bits/symb.					2.0							Total complexity	88

Table 6.4: Summary of the properties of the multi-level code of case 2.

has a decoding complexity of only 88. In comparison, from table 4.10 of chapter 4, for a geometrically uniform trellis code over 2×8-PSK transmitting 2 bits/symbol to achieve a diversity of 5, requires a trellis with 256 states. The normalised complexity of such a trellis is $2^{4+8}/2 = 2048$ which is about 23 times more complex. Of course the trellis code has a much higher minimum product distance and lower multiplicity. Figure 6.11 shows the pairwise probability curves of each of the three codes C_1, C_2 and C_3 and the trellis code C' for comparison. Note that the worst component code in the multi-level hierarchy is code C_2. The asymptotic performance loss relative to the trellis code C' is about 3 dB.

6.5 Selecting component codes.

As we have seen from the example case studies of section 6.4 the error performance of the overall multi-level code is dominated by the worst component code in the

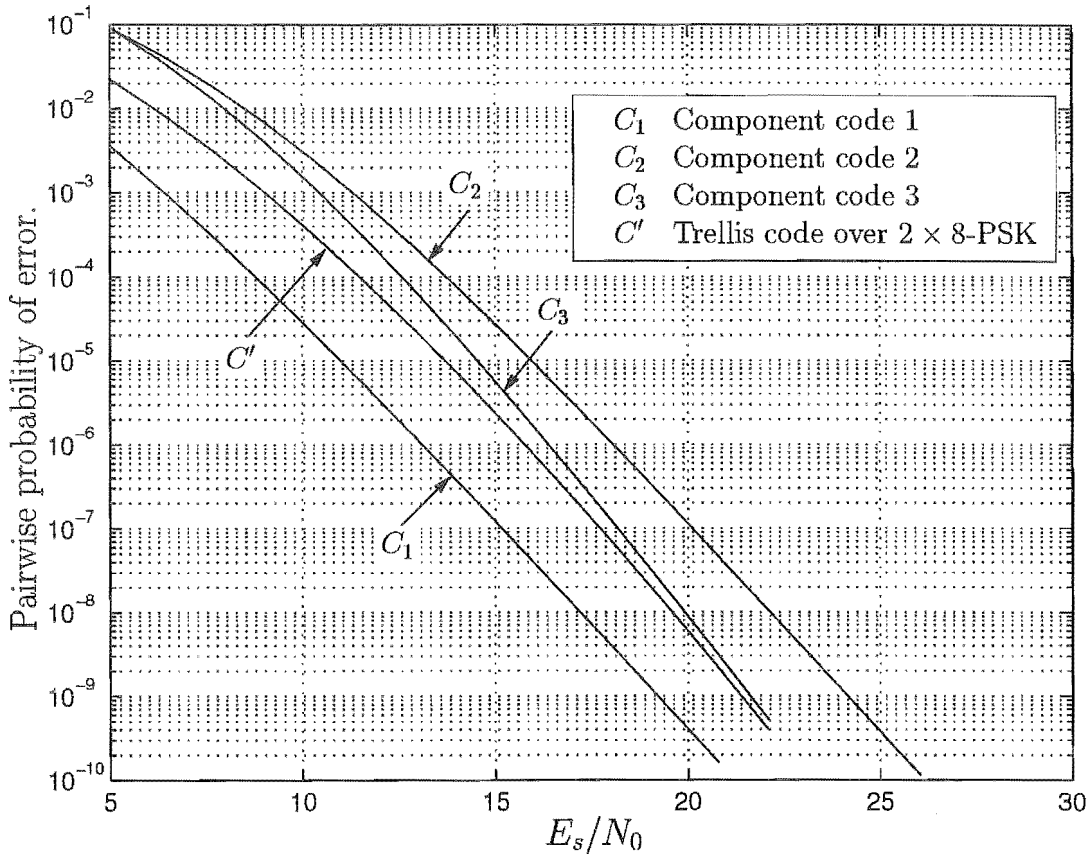


Figure 6.11: The performance of the component codes C_1, C_2, C_3 of the multi-level code of study 2. Plotted for comparison is the trellis code C' with the same Hamming distance.

hierarchy. Each component code C_i is defined as a convolutional group code over a label subgroup $\mathcal{A}_i$, where each element of $\mathcal{A}_i$ corresponds to a set of points from the partition chain. The overall distance properties of C_i are determined by the distance properties of $\mathcal{A}_i$ and by those of the convolutional code C_i itself. The main design parameter of interest for the Rayleigh fading channel is the Hamming distance of the code. Therefore to aid in the design of multi-level codes over geometrically uniform partitions we have constructed tables listing the achievable minimum Hamming distance in terms of the group and distance properties of $\mathcal{A}_i$ and in terms of the constraint length v_i of the convolution code C_i. This makes selection of the partition chain and component code rates easier, since determining the Hamming distance of the component code becomes a matter of table lookup.

6.5.1 Component codes over $\mathcal{A}_i \simeq Z^2$.

Table 6.5 lists the Hamming distances of codes over $(\mathcal{A}_i)^{p_i}$, for $p_i = 2$ to $p_i = 6$, where $\mathcal{A}_i \simeq Z_2$, the binary group. The label group $\mathcal{A}_i = \{A_0, A_1\}$ contains just two elements and has Hamming distance $d_H(\mathcal{A}_i)$. The Hamming distances of C_i listed in table 6.5 have been normalised to $d_H(\mathcal{A}_i)$. The first column of the table contains the constraint length v_i of C_i. The subsequent columns contain the normalised Hamming distance $d_H(C_i)/d_H(\mathcal{A}_i)$ for a given code rate b_i/p_i, where b_i is the number of input bits. The product distance and multiplicity of each code may be computed by the

d_H of component codes over $(Z_2)^{p_i}$								
v_i	b_i/p_i							
	1/2	1/3	2/3	2/4	3/4	3/5	4/5	5/6
1	3	5	2	4	2	3	2	2
2	5	8	3	5	3	4	2	2
3	6	10	4	6	4	5	3	3
4	7	12	5	7	4	6	4	3
5	8		6	8	5	6	4	4

Table 6.5: Normalised Hamming distance of rate b_i/p_i component codes over $(Z_2)^{p_i}$.

following formulas:

$$d_p^2(C_i) = \left(d_p^2(\mathcal{A}_i)\right)^{d_H(C_i)/d_H(\mathcal{A}_i)} \tag{6.58}$$

$$N(C_i) = \left(N(\mathcal{A}_i)\right)^{d_H(C_i)/d_H(\mathcal{A}_i)}. \tag{6.59}$$

Note that the code parameters of table 6.5 correspond simply to binary convolutional codes.

6.5.2 Component codes over $\mathcal{A}_i \simeq (Z_2)^2$.

The group $\mathcal{A}_i \simeq (Z_2)^2$ contains four elements and a greater number of possible Hamming distance combinations are possible than a group isomorphic to Z_2. From the case study 1 of section 6.4.1 we see that the level two partition label group is isomorphic to $(Z_2)^2$ and the Hamming distance profile can be described by the set $DP(\mathcal{A}_i, d_H) = \{1, 1, 1\}$. This means that every element is at a Hamming distance of

one from every other element. For case study 2 of section 6.4.2, the label groups for partition levels one and two are also isomorphic to $(Z_2)^2$, however they both have a Hamming distance profile described by $DP(\mathcal{A}_i, d_H) = \{2, 2, 1\}$ (see table 6.8). Therefore a component code over the latter distance profile will generally be better than a component code over the former. Tables 6.6, 6.7, and 6.8 list the greatest attainable Hamming distance of the component codes over $(Z_2)^2$ as a function of the normalised distance profiles $DP(\mathcal{A}_i, d_H) = \{1, 1, 1\}$, $DP(\mathcal{A}_i, d_H) = \{1, 1, 2\}$, and $DP(\mathcal{A}_i, d_H) = \{1, 2, 2\}$ respectively. The corresponding squared product distance d_p^2 and code multiplicity N, need to be evaluated on a case by case basis.

d_H of component codes over $(\mathcal{A}_i)^{p_i}$. $\mathcal{A}_i \simeq (Z_2)^2$. $DP(\mathcal{A}_i, d_H) = \{1, 1, 1\}$							
v	b_i/p_i						
	1/1	1/2	2/2	3/2	3/3	4/3	5/3
1	2	4	2	2	3	2	2
2	3	6	4	2	3	3	2
3	4	8	4	3	5	3	2
4	5	10	6	3	6	4	3

Table 6.6: Normalised Hamming distance of rate b_i/p_i component codes over $(\mathcal{A}_i)^{p_i}$, for $\mathcal{A}_i \simeq (Z_2)^2$ with normalised Hamming distance profile $DP(\mathcal{A}_i, d_H) = \{1, 1, 1\}$.

d_H of component codes over $(\mathcal{A}_i)^{p_i}$. $\mathcal{A}_i \simeq (Z_2)^2$. $DP(\mathcal{A}_i, d_H) = \{1, 1, 2\}$							
v	b_i/p_i						
	1/1	1/2	2/2	3/2	3/3	4/3	5/3
1	3	7	4	2	4	3	2
2	5	10	5	3	5	4	2
3	6	12	6	4	6	4	3
4	7	15	8	4	7	5	4

Table 6.7: d_H of rate b_i/p_i component codes over $(\mathcal{A}_i)^{p_i}$, for $\mathcal{A}_i \simeq (Z_2)^2$ with normalised distance profile $DP(\mathcal{A}_i, d_H) = \{1, 1, 2\}$.

Normalised Hamming distance of component codes over $(\mathcal{A}_i)^{p_i}$. $\mathcal{A}_i \simeq (Z_2)^2$. $DP(\mathcal{A}_i, d_H) = \{1, 2, 2\}$							
v	b_i/p_i						
	1/1	1/2	2/2	3/2	3/3	4/3	5/3
1	4	8	4	3	5	4	2
2	6	12	6	4	6	5	3
3	7	15	8	5	8	6	4
4	8	18	10	5	10	7	5

Table 6.8: Normalised Hamming distance of rate b_i/p_i component codes over $(\mathcal{A}_i)^k$, for $\mathcal{A}_i \simeq (Z_2)^2$ with normalised distance profile $DP(\mathcal{A}_i, d_H) = \{1, 2, 2\}$.

We observe, as expected, that the attainable Hamming distance of component over $(Z_2)^2$ increases as the distance profile of the underlying partition improves. The Hamming distances of the component codes over the profile $DP(\mathcal{A}_i, d_H) = \{1, 2, 2\}$ are almost double those over $DP(\mathcal{A}_i, d_H) = \{1, 1, 1\}$.

6.5.3 Component codes over Z_4.

The other label group of order four occurring in partition chains is the group isomorphic to Z_4. Generally component group codes over this group do not achieve the same Hamming distances as codes over $(Z_2)^2$ for the same Hamming distance profile. This is due to the more constrained algebraic structure of the group Z_4 compared to $(Z_2)^2$. Tables 6.9 and 6.10 list the attainable Hamming distance of a component code constructed over Z_4 for the Hamming distance profiles $d_H = \{1, 1, 1\}$ and $d_H = \{1, 2, 2\}$ respectively. The Hamming distance of one in the latter profile corresponds to the element $g \neq e$ of Z_4 such that $g^2 = e$. The profile $d_H = \{1, 1, 2\}$ is not considered because it does not occur in any partition chains examined. By inspection of tables 6.9 and 6.10, the codes over Z_4 do not achieve Hamming distances nearly as good as those over $(Z_2)^2$. In fact the change of Hamming distance profile of the codes over Z_4 from $DP(\mathcal{A}_i, d_H) = \{1, 1, 1\}$ to $DP(\mathcal{A}_i, d_H) = \{1, 2, 2\}$ did little to improve the achievable Hamming distances of the component codes. This is due to the Hamming distance of one corresponding to the element $g \in Z_4$, such that $g^2 = e$ and it is this element which with the identity element forms the only

d_H of component codes over $(\mathcal{A}_i)^{p_i}$. $\mathcal{A}_i \simeq Z_4$. $DP(\mathcal{A}_i, d_H) = \{1,1,1\}$						
v	b_i/p_i					
	1/2	2/2	3/2	3/3	4/3	5/3
1	3	2	1	2	2	1
2	5	3	2	3	2	2
3	6	3	3	4	3	2
4	7	5	3	5	4	3

Table 6.9: Normalised Hamming distance of rate b_i/p_i component codes over $(\mathcal{A}_i)^{p_i}$, for $\mathcal{A}_i \simeq Z_4$ with normalised distance profile $\{1,1,1\}$.

d_H component codes over $(\mathcal{A}_i)^{p_i}$. $\mathcal{A}_i \simeq Z_4$. $DP(\mathcal{A}_i, d_H) = \{1,2,2\}$						
v	k/n					
	1/2	2/2	3/2	3/3	4/3	5/3
1	3	2	1	2	2	1
2	5	3	2	3	2	2
3	6	3	3	5	3	2
4	7	5	3	5	5	3

Table 6.10: Normalised Hamming distance of rate b_i/p_i component codes over $(\mathcal{A}_i)^k$, for $\mathcal{A}_i \simeq Z_4$ with normalised distance profile $\{1,2,2\}$.

non-trivial subgroup of Z_4. In general it is therefore preferable to construct codes over label groups $\mathcal{A}_i$ isomorphic to $(Z_2)^2$ rather than Z_4.

6.6 Codes over 1×8-PSK.

The paper by Seshadri and Sundberg constructs multi-level codes for the Rayleigh fading channel over an 8-PSK constellation [65]. Their construction is based on the conventional Ungerboeck three-level partitioning of 8-PSK [75] and the same partitioning is also achieved by any normal subgroup chain over a generating group of 8-PSK. We have summarised the properties of the partition in table 6.11. The label group of each of the three levels is isomorphic to Z_2 and a multi-level code is

Partition II of 1×8-PSK							
i	G_i/G_{i-1}	m_i	Coset	Map to $\mathcal{A}_i$	d_H	d_p^2	N
1	Z_2	2	r_4	1	1	4.0	1
2	Z_2	4	r_2	1	1	2.0	2
3	Z_2	4	r_1	1	1	0.587	2

Table 6.11: Summary of the distance properties of a three level partitioning of 8-PSK.

constructed by defining three linear binary codes over Z_2. Seshadri and Sundberg used identical rate 2/3 binary convolutional codes over each level to give an overall data rate of 2 bits per symbol and achieved the minimum Hamming distances listed in the column labelled "2/3" of table 6.5 as a function of the constraint lengths of the convolution encoders. The error performance of this construction is dominated by the code with the worst distance properties. This is the code defined over the third level partition which has a minimum product distance of 0.587. In table 6.12 we have listed the minimum distance properties and normalised code complexity of this code construction as a function of the constraint length of the component codes. The normalised complexity of the code increases exponentially with the

v_i	d_H	d_p^2	N	$\xi(C)$
1	2	0.34	4	10
2	3	0.20	24	20
3	4	0.12	32	36
4	5	0.070	160	68
5	6	0.041	1088	132
6	7	0.024	2176	260

Table 6.12: Code parameters of multilevel constructions over 1×8-PSK.

increasing constraint length of the component codes. In figure 6.12 we have plotted the probability of error curves of the component codes of table 6.12. Note that each unit increase in the component code constraint length leads to a diminishing return in coding gain. The multi-level codes outperform the best trellis codes of chapter 4 also transmitting 2 bits/symbol in terms of code complexity. For example curve 5 of figure 6.12, corresponding to a multi-level code with $v = 5$ and $d_H = 6$, is similar to

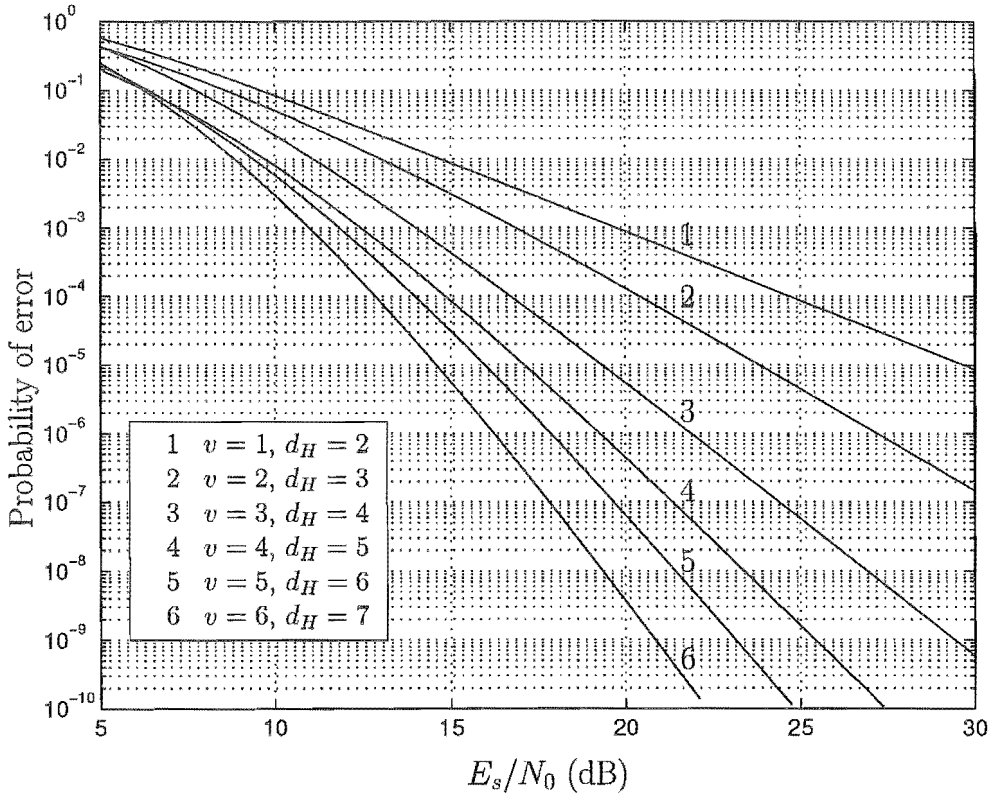


Figure 6.12: The error performance of multi-level codes over partition II of 1 × 8-PSK. The overall code rate is 2 bits/symbol.

the performance of a trellis code over 2 × 16-PSK with $v = 7$ and $d_H = 4$ at a BER of 10^{-9}. However the normalised complexity of the multi-level code is 132 compared to 1024 of the trellis code, and represents a reduction in complexity of a factor of about 8.

We may attempt to improve upon the minimum squared product distance of the partition described by the label group $\mathcal{A}_3$ by employing the two-level partitioning of the 8-PSK constellation used in case study 1 of section 6.4.1, and is illustrated by figure 6.5. This two-level partitioning is based on the generating group $G(S) = \{r_0, r_1, r_2, r_3, v_0, v_1, v_2, v_3\}$, isomorphic to D_4, partitioned using the partition chain consisting of the normal subgroups $G_1 = \{r_0, r_4\}$ and $G_2 = G$. Of the two levels of the partition, one is two-way and the other is four-way and the distance properties are described by table 6.1 of section 6.4.1. Note that the four-way partition contains one element with a product distance of $d_p^2 = 2.0$ which can be exploited by the component code to produce a better overall product distance. In table 6.13 we have listed the properties of 4 multi-level code designs. The component codes for C_1 are

over $(\mathcal{A}_1)^3$ and have 2 input bits each. The component codes for C_2 are over $(\mathcal{A}_2)^3$ and have 4 input bits each to give an overall rate of 2 bits per symbol for each code. Although we have improved upon the minimum product distance (and multiplicity)

Design	C_1, $b_1 = 2$, $r_1 = \frac{1}{2}$					C_2, $b_2 = 4$, $r_2 = \frac{4}{3}$					C, $r = 2.0$
	v_1	d_H	d_p^2	N	$\xi(C_1)$	v_2	d_H	d_p^2	N	$\xi(C_2)$	$\xi(C)$
1	1	2	16.0	1	4	1	2	1.172	6	44/3	18.7
2	2	3	64.0	3	16/3	2	3	0.689	6	88/3	34.7
3	3	4	256	2	32/3	4	4	0.402	6	264/3	98.7
4	4	5	1024	5	64/3	5	5	0.236	18	520/3	194.7

Table 6.13: Four multi-level code designs over partition I of $1 \times$ 8-PSK.

for a given Hamming distance by using a two-level partitioning, it has been at the cost of an increase in decoding complexity. This is consistent, as effectively the design is "closer" to a conventional trellis code over 8-PSK, relative to a three level multi-level code design. In general, the more levels in the multi-level encoder, the lower the decoder complexity for the code distances achieved.

6.7 Codes over $2 \times$ 8-PSK.

Multi-level codes over partitions of $2 \times$ 8-PSK are more interesting because they allow for much more design flexibility and design trade-offs than multi-level codes over $1 \times$8-PSK. The partition chain can be selected such that some of the levels have a Hamming distance of two allowing the data rates of the component codes to be varied to give better coding gains. The number of possible partitions of $2 \times$ 8-PSK is large and we will only examine one of the more interesting ones. First we need to decide on the number of partition levels, from 1 to 6, as well as the orders of each partition. For example, in the case study 2 of section 6.4.2 we partitioned $2 \times$8-PSK into three levels, each of order 4. Alternatively we may partition the constellation into six levels of order two each. Consider the six level binary partitioning of $2 \times$ 8-PSK based on the generating group $G(S) \simeq (Z_8)^2$. Table 6.14 summarises the properties of the the partition chain we have selected. Each level is isomorphic to the binary group Z_2. Note that the first three levels of the partition have a minimum

Hamming distance of two, while the remaining three levels have a Hamming distance of one. We may take advantage of the difference in Hamming distances of the levels by increasing the data rate of the top three levels and reducing the data rate of the lower three levels. By doing so we can make the distance properties of the component codes at each level similar. Each label group is the same and equals $\mathcal{A}_i = \{0, 1\}$. By

partition II of 2 × 8-PSK						
i	G_i/G_{i-1}	m_i	Coset	d_H	d_p^2	N
1	Z_2	2	(r_4, r_4)	2	16.0	1
2	Z_2	2	(r_2, r_6)	2	4.0	2
3	Z_2	2	(r_1, r_3)	2	2.0	4
4	Z_2	2	(r_0, r_4)	1	4.0	2
5	Z_2	2	(r_4, r_2)	1	2.0	4
6	Z_2	2	(r_2, r_3)	1	0.586	4

Table 6.14: Summary of the distance properties of a six level binary partition of 2 × 8-PSK.

raising each label group to the power of six, we can elect to transmit five bits per level for the top three levels, and three bits per level on the lower three levels. This choice, while making good use of the higher level distance properties, maintains an overall data rate of two bits per symbol:

$$\begin{aligned} r &= \frac{1}{L}\sum_{i=1}^{6}\frac{b_i}{p_i} \\ &= \frac{1}{2}(\frac{5}{6}+\frac{5}{6}+\frac{5}{6}+\frac{3}{6}+\frac{3}{6}+\frac{3}{6}) \\ &= 2.0 \text{ bits/symbol} \end{aligned} \tag{6.60}$$

To select the component codes we utilise table 6.5 which lists the minimum distances of codes over $(Z_2)^{p_i}$. For a minimal complexity design, we select $v_i = 1$ for each level, and conveniently this choice gives a minimum Hamming distance for every level of four. The properties of each component code are summarised in table 6.15. The complexity of this code is about 40% higher than the code of table 6.12 over 1 × 8-PSK with equal minimum Hamming distance. Clearly it is better to utilise the the code over 1 × 8-PSK to achieve a Hamming distance of four. However if we desire to design codes with a higher Hamming distance then a code over 2 × 8-PSK can have

Code over $S = 2 \times 8$-PSK, partition II													
i	m_i	$\mathcal{A}_i$	p_i	n_i	r_i	v_i	d_H	d_p^2	N	$\xi_A(C_i)$	$\xi_P(C_i)$	$\xi_T(C_i)$	$\xi(C_i)$
1	2	Z_2	6	5	$\frac{5}{12}$	1	4	256.0	10	0	16/3	1/3	5.7
2	2	Z_2	6	5	$\frac{5}{12}$	1	4	16.0	40	2	16/3	1/3	7.7
3	2	Z_2	6	5	$\frac{5}{12}$	1	4	4.0	160	4	16/3	1/3	9.7
4	2	Z_2	6	3	$\frac{3}{12}$	1	4	256.0	48	8	4/3	1/3	9.7
5	2	Z_2	6	3	$\frac{3}{12}$	1	4	16.0	768	8	4/3	1/3	9.7
6	2	Z_2	6	3	$\frac{3}{12}$	1	4	0.12	768	8	4/3	1/3	9.7
Code rate =					2.0	Total complexity							52

Table 6.15: Summary of the properties of a multi-level code design over partition II of 2×8-PSK. Design distance $d_H = 4$.

less complexity. By selecting each component code to have a constraint length of $v_i = 3$ we achieve a Hamming distance of $d_H = 6$ and the properties of this design are summarised in table 6.16. This multi-level code has a minimum squared

Code over $S = 2 \times 8$-PSK, partition II													
i	m_i	$\mathcal{A}_i$	p_i	n_i	r_i	v_i	d_H	d_p^2	N	$\xi_A(C_i)$	$\xi_P(C_i)$	$\xi_T(C_i)$	$\xi(C_i)$
1	2	Z_2	6	5	$\frac{5}{12}$	3	6	4096	6	0	5.3	5.3	10.7
2	2	Z_2	6	5	$\frac{5}{12}$	3	6	64	48	2	5.3	5.3	12.7
3	2	Z_2	6	5	$\frac{5}{12}$	3	6	8	384	4	5.3	5.3	14.7
4	2	Z_2	2	1	$\frac{1}{4}$	3	6	4096	64	8	0	4	12
5	2	Z_2	2	1	$\frac{1}{4}$	3	6	64	4096	8	0	4	12
6	2	Z_2	2	1	$\frac{1}{4}$	3	6	0.04	4096	8	0	4	12
Code rate =					2.0	Total complexity							74

Table 6.16: Summary of the properties of a multi-level code over partition II of 2×8-PSK. Design distance $d_H = 6$.

product distance of $d_p^2 = 0.04$, the same as the $v = 5$ code over 1×8-PSK of table 6.12 however the multiplicity is approximately a factor of two greater. The decoding complexity of the code is only 74 while for the code over 1×8-PSK it is 132 and represents a saving of 46%. By increasing the constraint lengths of the component codes to $v_i = 5$ we achieve a Hamming distance of $d_H = 8$ for a squared product

distance of $d_p^2 = 0.014$ as shown in table 6.17. The decoding complexity for this

Code over $S = 2 \times 8$-PSK, partition II													
i	m_i	$\mathcal{A}_i$	p_i	n_i	r_i	v_i	d_H	d_p^2	N	$\xi_A(C_i)$	$\xi_P(C_i)$	$\xi_T(C_i)$	$\xi(C_i)$
1	2	Z_2	6	5	$\frac{5}{12}$	5	8	65536	16	0	0	85.3	85.3
2	2	Z_2	6	5	$\frac{5}{12}$	5	8	256	256	2	0	85.3	87.3
3	2	Z_2	6	5	$\frac{5}{12}$	5	8	16	4096	4	0	85.3	89.3
4	2	Z_2	2	1	$\frac{1}{4}$	5	8	65536	256	8	0	16	24
5	2	Z_2	2	1	$\frac{1}{4}$	5	8	256	65536	8	0	16	24
6	2	Z_2	2	1	$\frac{1}{4}$	5	8	0.014	65536	8	0	16	24
Code rate =					2.0	Total complexity							334

Table 6.17: Summary of the properties of a multi-level code over partition II of 2 × 8-PSK. Design distance $d_H = 8$.

code design is 334 and is a significant increase relative to the $d_H = 6$ code, however it still represents a 35% reduction in complexity relative to a multi-level code over 1×8-PSK from table 6.12. In figure 6.13 we have plotted the union bound on the probability of error of the three multi-level code designs of tables 6.15, 6.16 and 6.17.

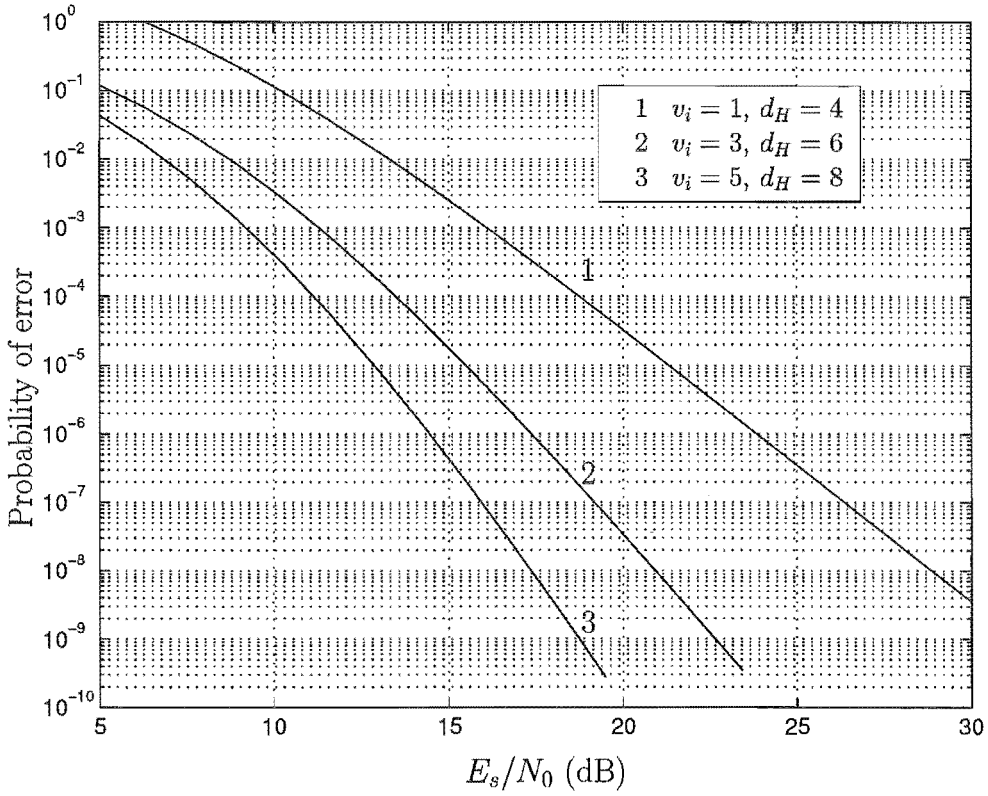


Figure 6.13: The error performance of multi-level codes over partition II of 2 × 8-PSK. The overall code rate is 2 bits/symbol.

6.8 Codes over 16-QAM.

The constellation 16-QAM is not a GU constellation. However as shown in section 5.3 of chapter 5 it is possible to write 16-QAM as a GU partition tree. The conventional four-way GU partitioning is shown in figure 6.14. This partition of 16-QAM

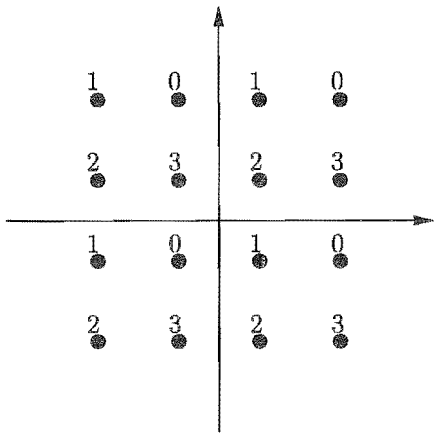


Figure 6.14: The conventional 4-way partitioning of 16-QAM.

divides the constellation into four subsets of four points each and is written as

$$\begin{aligned} S_P &= \{S_{P_0}, S_{P_1}, S_{P_2}, S_{P_3}\} \\ &= \{\{(3,3),(-1,3),(-1,-1),(3,-1)\}, \\ &\quad \{-3,-3),(1,3),(1,-1),(-3,-1)\}, \\ &\quad \{-3,-3),(1,-3),(1,1),(-3,1)\}, \\ &\quad \{(3,-3),(-1,-3),(-1,1),(3,1)\}\}. \end{aligned} \tag{6.61}$$

The four subsets form a GU partition with a symmetry group identical to the symmetry group of a 4-PSK constellation, that is the partition has the same set of symmetries. The distance properties of this partition, after the normalisation to unit energy of the constellation, are listed in table 6.18. The table lists the distance properties from the subset S_{P_0} to each of the other three subsets, S_{P_1}, S_{P_2} and S_{P_3}. Note that in order to calculate the minimum distance properties between S_{P_0} and another subset S_{P_i} we must consider every point S_{P_0} and determine the point with the worst distance properties to the points in S_{P_i}. If the partition was based on a GU constellation then only *one* point would need to be considered as shown in section 5.5.1 of chapter 5. Each subset of the partition is a GU constellation and is identical to 4-PSK with a shifted origin. Clearly the subsets also have a symmetry group

	d_H	d_p^2	N
S_{P_1}	1	0.4	2
S_{P_2}	1	0.8	4
S_{P_3}	1	0.4	2

Table 6.18: Minimum distance properties from subset S_{P_0} to each other subset.

isomorphic to that of 4-PSK. The distance properties of the subset S_{P_0} are listed in table 6.19. This partitioning gives two levels with quite different distance properties.

	d_H	d_p^2	N
1	1	1.6	1
2	1	3.2	1
3	1	1.6	1

Table 6.19: Minimum distance properties of the subset S_{P_0}. Distances from the point 0 to each other point are listed.

The level with the lower squared product distances also has a higher multiplicity. Since a multi-level code is dominated by the worst code in the hierarchy, there is little point in having two levels with quite different distance properties. In fact we are better off with the partitioning shown in figure 6.15. This method of partitioning has

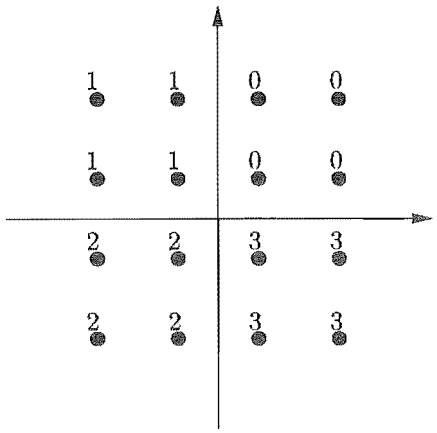


Figure 6.15: A 4-way partitioning of 16-QAM more suitable for multi-level coding.

the advantage of giving exactly the same distance properties on each level. These

are summarised in table 6.20. To define a multi-level code over this partitioning of

Partition I of 16-QAM							
i	$\mathcal{A}_i$	m_i	Coset	Map to $\mathcal{A}_i$	d_H	d_p^2	N
1	$(Z_2)^2$	4	r_2	01	1	0.4	1
			v_1	10	1	0.8	1
			v_3	11	1	0.4	1
2	$(Z_2)^2$	4	r_2	01	1	0.4	1
			v_1	10	1	0.8	1
			v_3	11	1	0.4	1

Table 6.20: Summary of the distance properties of an alternate partitioning of 16-QAM.

16-QAM we may simply use two rate 1 bit/symbol GU codes over 4-PSK listed in the code tables of chapter 4. Two 1 bit/symbol encoders in the multi-level structure will give a total rate of two bits/symbol. The distance properties (except for the Hamming distance) of this code will be different from the corresponding code over 4-PSK due to the underlying partition properties. In table 6.21 we have listed three codes with Hamming distance of 4, 6 and 8 respectively, based on this construction. The three code designs of table 6.21 all exhibit a good Hamming distance versus decoder complexity trade-off. The first code has a Hamming distance of four and normalised decoding complexity of 24. In comparison to the multi-level code over 1×8-PSK of table 6.12 with the same Hamming distance, the code over 16-QAM has approximately a 33% lower decoding complexity, however its product distance is a factor of two worse. The second multi-level code over 16-QAM has a Hamming distance of six, and a normalised decoding complexity of 72. Again this code improves over the same Hamming distance code over 1×8-PSK of table 6.12 in terms of decoding complexity by about 45%, but has a worse squared product distance by about a factor of five. However it does not improve on the multi-level code over 2×8-PSK of table 6.16 which also has a Hamming distance of six, a comparable decoder complexity, but a better product distance. Finally the third code of table 6.21 has a Hamming distance of eight and decoding complexity of 264. This code also improves over the code with the same Hamming distance of table 6.12 in terms of decoding complexity by a factor of two, at the cost of a reduced squared product

Code 1 over S = 16-QAM, partition I.													
i	m_i	$\mathcal{A}_i$	p_i	b_i	r_i	v_i	d_H	d_p^2	N	$\xi_A(C_i)$	$\xi_P(C_i)$	$\xi_T(C_i)$	$\xi(C_i)$
1	4	$(Z_2)^2$	2	2	$\frac{2}{2}$	2	4	5.1×10^{-2}	2	0	0	8	8
2	4	$(Z_2)^2$	2	2	$\frac{2}{2}$	2	4	5.1×10^{-2}	2	8	0	8	24
					Rate =	2.0						Total complexity	24

Code 2 over S = 16-QAM, partition I													
i	m_i	$\mathcal{A}_i$	p_i	b_i	r_i	v_i	d_H	d_p^2	N	$\xi_A(C_i)$	$\xi_P(C_i)$	$\xi_T(C_i)$	$\xi(C_i)$
1	4	$(Z_2)^2$	2	2	$\frac{2}{2}$	4	6	8.2×10^{-3}	2	0	0	32	32
2	4	$(Z_2)^2$	2	2	$\frac{2}{2}$	4	6	8.2×10^{-3}	2	8	0	32	40
					Rate =	2.0						Total complexity	72

Code 3 over S = 16-QAM, partition I													
i	m_i	$\mathcal{A}_i$	p_i	b_i	r_i	v_i	d_H	d_p^2	N	$\xi_A(C_i)$	$\xi_P(C_i)$	$\xi_T(C_i)$	$\xi(C_i)$
1	4	$(Z_2)^2$	2	2	$\frac{2}{2}$	6	8	1.3×10^{-3}	1	0	0	128	128
2	4	$(Z_2)^2$	2	2	$\frac{2}{2}$	6	8	1.3×10^{-3}	1	8	0	128	136
					Rate=	2.0						Total complexity	264

Table 6.21: Summary of the properties of three multi-level code designs over partition I of 16-QAM.

distance by a factor of 10.

6.8.1 Four level partitions of 16-QAM.

We may improve upon the distance properties of the multi-level code over 16-QAM of the previous section by partition each of the two partition levels into two again, resulting in a four level binary partition, the properties of which are listed in table 6.22. Each level of this four-level partition is isomorphic to Z_2 and we may use any binary code to code over the sublevels. From table 6.5, column one, we see that rate 1/2 binary codes achieve a Hamming distance of six and eight with an encoder constraint length of three and five respectively. The code properties of the multi-level code construction over the partition of table 6.22 and the binary convolutional codes of table 6.5 are listed in table 6.23. The first code attains a Hamming distance of $d_H = 4$ at a complexity of 40. This is not as good as the $d_H = 4$ code of table 6.21. However the codes with $d_H = 6$ and $d_H = 8$ both improve over the codes of table

Partition II of 16-QAM							
i	$\mathcal{A}_i$	m_i	Coset	Map to $\mathcal{A}_i$	d_H	d_p^2	N
1	Z_2	2	r_2	1	1	0.8	1
2	Z_2	2	v_1	1	1	0.4	2
3	Z_2	2	r_2	1	1	0.8	1
4	Z_2	2	v_1	1	1	0.4	2

Table 6.22: Summary of the distance properties of a four-level binary partition of 16-QAM.

Code 1 over S = 16-QAM, partition II													
i	m_i	$\mathcal{A}_i$	p_i	b_i	r_i	v_i	d_H	d_p^2	N	$\xi_A(C_i)$	$\xi_P(C_i)$	$\xi_T(C_i)$	$\xi(C_i)$
1	2	Z_2	4	2	$\frac{2}{4}$	1	4	0.41	5	0	2	1	3
2	2	Z_2	4	2	$\frac{2}{4}$	1	4	0.026	80	4	2	1	7
3	2	Z_2	4	2	$\frac{2}{4}$	1	4	0.41	5	8	2	1	11
4	2	Z_2	4	2	$\frac{2}{4}$	1	4	0.026	80	16	2	1	19
				Rate =	2.0							Total complexity	40

Code 2 over S = 16-QAM, partition II													
i	m_i	$\mathcal{A}_i$	p_i	b_i	r_i	v_i	d_H	d_p^2	N	$\xi_A(C_i)$	$\xi_P(C_i)$	$\xi_T(C_i)$	$\xi(C_i)$
1	2	Z_2	2	1	$\frac{1}{2}$	3	6	0.26	1	0	0	8	8
2	2	Z_2	2	1	$\frac{1}{2}$	3	6	4.1×10^{-3}	64	4	0	8	12
3	2	Z_2	2	1	$\frac{1}{2}$	3	6	4.1×10^{-3}	1	8	0	8	16
4	2	Z_2	2	1	$\frac{1}{2}$	3	6	0.0041	64	16	0	8	24
				Rate =	2.0							Total complexity	60

Code 3 over S = 16-QAM, partition II													
i	m_i	$\mathcal{A}_i$	p_i	b_i	r_i	v_i	d_H	d_p^2	N	$\xi_A(C_i)$	$\xi_P(C_i)$	$\xi_T(C_i)$	$\xi(C_i)$
1	2	Z_2	2	1	$\frac{1}{2}$	5	8	0.16	1	0	0	32	32
2	2	Z_2	2	1	$\frac{1}{2}$	5	8	6.6×10^{-4}	256	4	0	32	36
3	2	Z_2	2	1	$\frac{1}{2}$	5	8	0.16	1	8	0	32	40
4	2	Z_2	2	1	$\frac{1}{2}$	5	8	6.6×10^{-4}	256	16	0	32	48
				Rate =	2.0							Total complexity	156

Table 6.23: Summary of the properties of three multi-level code designs over partition II of 16-QAM.

6.21 in terms of decoding complexity, however at the cost of the minimum squared product distance. In figure 6.16 we have plotted the probability of error curves of

the three 16-QAM multi-level codes of table 6.23.

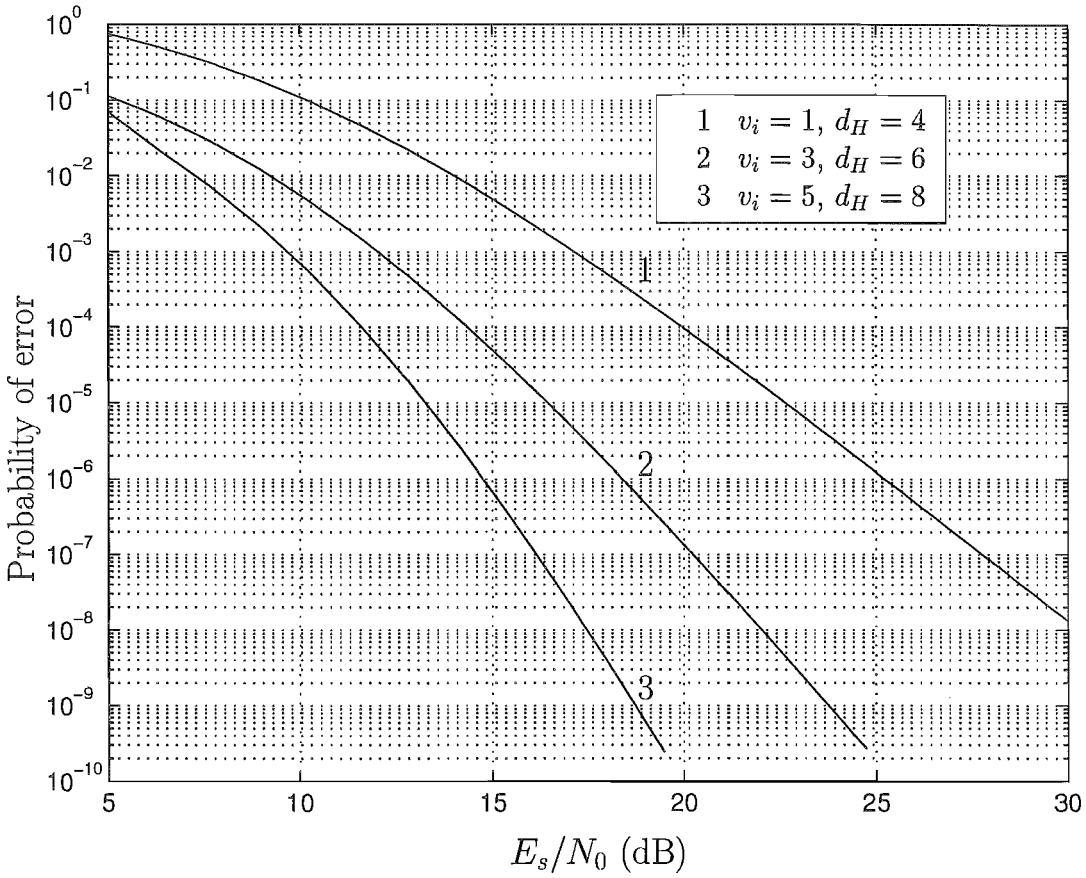


Figure 6.16: The performance curves of three 16-QAM multi-level code designs.

6.9 Complexity versus coding gain.

We have shown that the technique of multi-level coding combined with staged decoding reduces the decoding complexity for a given code minimum Hamming distance. This reduction in complexity however is in general at the cost of the minimum squared product distance and code multiplicity, which in turn affects the coding gain. The interplay and tradeoffs between the various parameters makes it difficult to judge which code is best. In a practical system we are interested in achieving a given probability of error at the minimum possible signal power, while maintaining an affordable level of complexity. To better judge the efficiency of the presented codes, both the trellis codes of chapter 4 and the multi-level codes of this chap-

ter, we have plotted, for a given probability of error, the required SNR versus the code complexity. In figure 6.17 we have plotted this relationship for some of the codes presented for a probability of error level of 10^{-9}. The closer to the origin

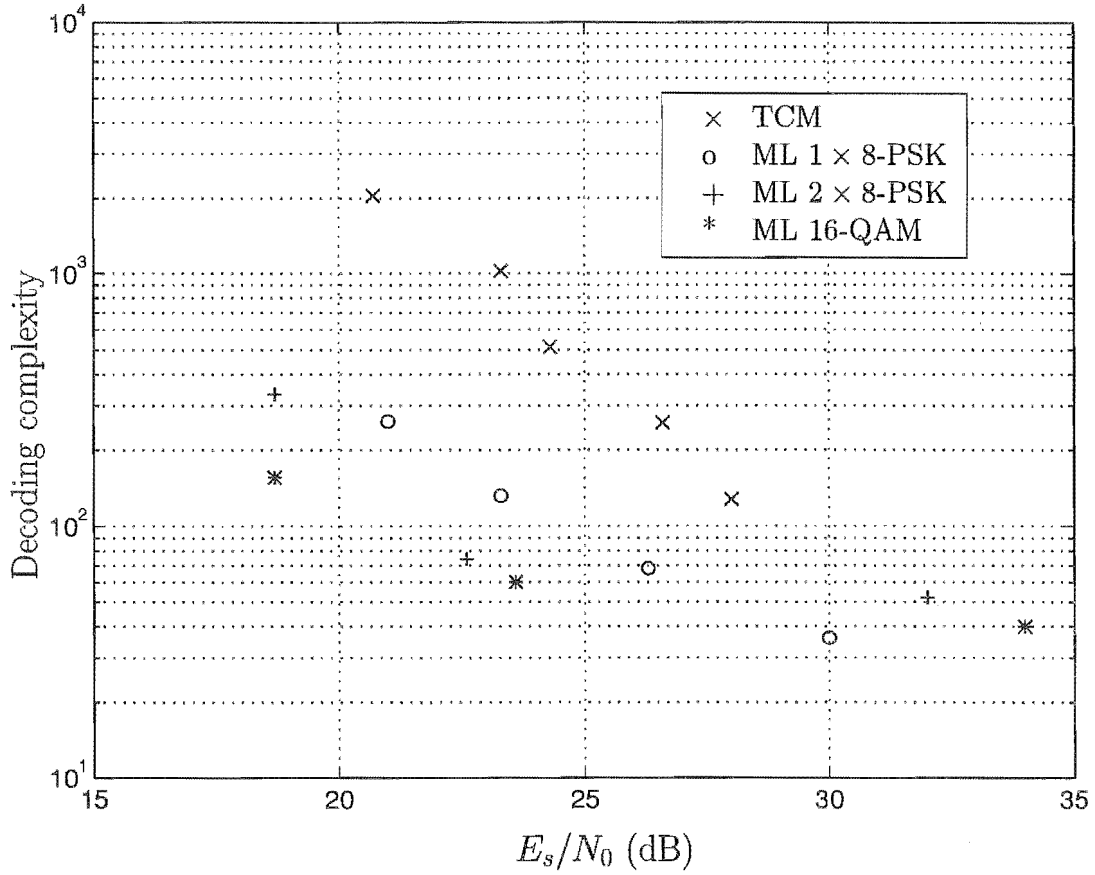


Figure 6.17: Comparison of SNR at a probability of error of 10^{-9} versus decoding complexity for different coding schemes.

the code point lies, the greater the code efficiency. The TCM schemes of chapter 4 are as expected the least efficient. Multi-level codes over 1×8-PSK are a significant improvement, and multi-level codes over multi-dimensional constellation such as 2×8-PSK improve the code efficiency further.

6.10 Summary.

We have investigated some of the design trade-offs of multi-level codes over GU partitions. The fact that each component code is defined over a GU partition implies

the error properties of that code may be completely characterised by considering only the all-zero codeword. In general this does not imply that the overall multi-level code is GU. We have found that multi-level codes can achieve the same probability of error as GU trellis codes for a lower decoding complexity. This reduction in decoding complexity is most pronounced for higher Hamming distance codes, for which the complexity of trellis codes is large. We have by no means exhausted the code design possibilities and we believe that multi-level codes over multi-dimensional constellations with $L \geq 4$ can achieve extremely good performance with modest complexity.

Chapter 7

Multiple symbol differential detection.

In this chapter we combine the results and techniques from chapters 2, 4, 5 and 6 into one system. For the code search results presented in chapters 4 and 6 we evaluated the code performance assuming ideal channel state information, however in practice a real channel state estimate is required. In section 2.4 of chapter 2 we showed that the method of pilot tone aided detection performs at a penalty of a few dB relative to ideal channel state information. Pilot tone aided detection has its drawbacks, such as the difficulty of passing the signal through a non-linear amplifier, and as an alternative we examine differential detection as a means of obtaining a channel state estimate. In a conventional differentially detected system, the data is encoded in the difference between the phases of adjacent symbols and this means that an absolute phase reference is not required at the receiver. For the Rayleigh fading channel differential encoding implies that a channel state estimate for the current symbol is obtained from the previously received symbol. However because of the time varying nature of the channel, the accuracy of the estimate is a function of the rate of change of the channel state as determined by the normalised fade rate parameter $f_D T$. If the channel state varies slowly with time then the estimate of it will be quite accurate. However, as the speed of variation increases, the accuracy of the estimate decreases. The inaccuracy of the channel state estimate leads to an error floor in the performance of conventional differential detection on the Rayleigh fading channel. Figure 7.1 illustrates this phenomena. We have plotted the probability of error, as a

function of SNR, of an uncoded conventional differentially encoded system operating on the Rayleigh fading channel. Curves for $f_DT = 0.1$, $f_DT = 0.03$, $f_DT = 0.01$,

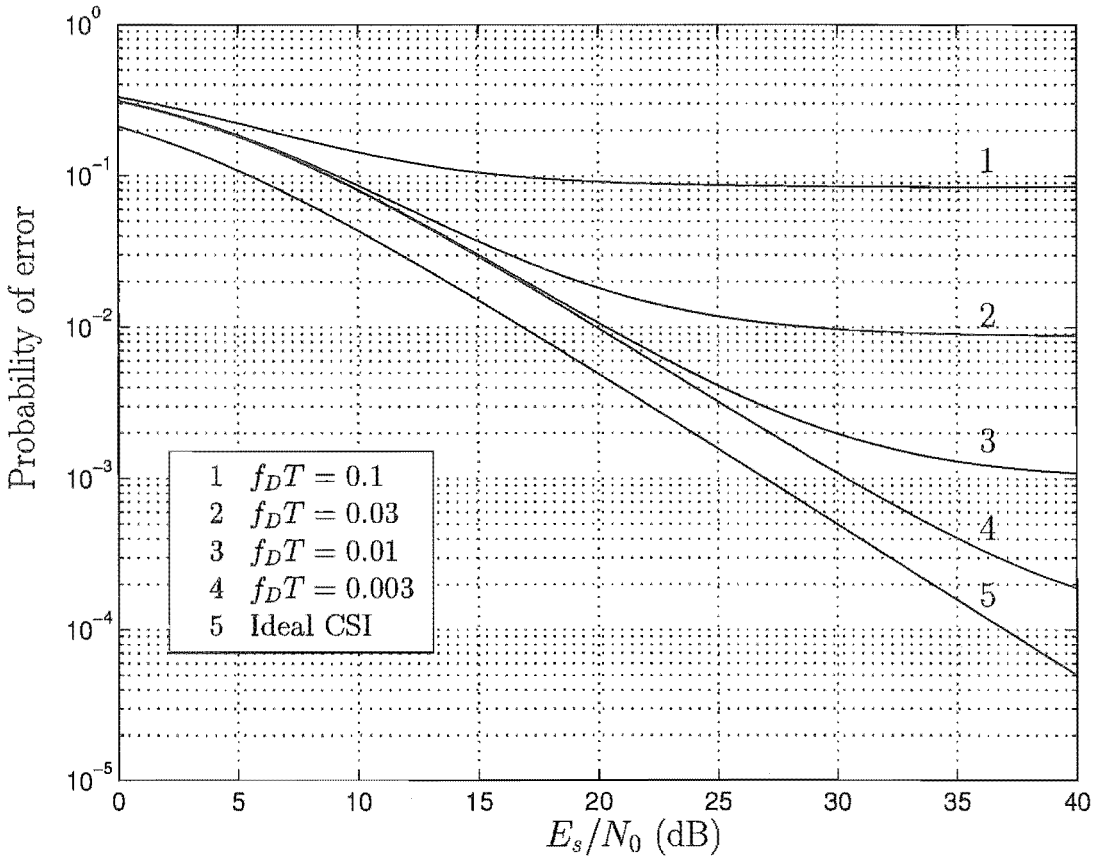


Figure 7.1: Performance curves of a conventional differentially detected 4-PSK system operating on the Rayleigh fading channel.

$f_DT = 0.003$ and ideal channel state information have been plotted. At fast fading (curve 1, $f_DT = 0.1$) the error floor bottoms out at a probability of error of about 9×10^{-2}. As the rate of fading is reduced, the error floor is lowered, until at very slow fading the channel looks Gaussian and differential detection performs at a penalty of 3 dB relative to coherent detection. Clearly conventional differential detection is not well suited for operation on the Rayleigh fading channel. Divsalar and Simon developed the technique of multiple symbol differential detection for the AWGN channel [21],[24] and extended it, as did Ho and Fung, to the Rayleigh fading channel [20],[32],[67]. MSDD improves significantly upon conventional differential detection [25],[33],[68],[78], [81]. In this technique a decision is made on blocks of

L symbols combined with a phase reference symbol. This method is very effective and reduces the error floor dramatically even for relatively fast fading ($f_D T \approx 0.1$). The technique exploits the correlation of the fading process to estimate the channel state more accurately. In this chapter we use MSDD and combine it with multilevel coding to further improve the performance of the system.

We know from section 2.5 of chapter 2 that for coding to be effective, the code symbols in the sequence must be independently faded. This is normally achieved by symbol interleaving. On the other hand, for multiple symbol differential detection to be effective we require the differentially encoded symbols to be highly correlated. This is a conflict in requirements. Our proposal to solve this problem is to interleave symbols in *blocks* of L symbols, rather than on a symbol by symbol basis. Each block is differentially encoded, and decoded using multiple symbol differential detection. We define a code over the blocks, treating the blocks as symbols from a larger alphabet. The interleaving/deinterleaving process ensures that the fading affecting each block is independent from block to block and code diversity is obtained by the appropriate definition of the code. The code will be a multi-level construction based on geometrically uniform (where geometric uniformity will be defined relative to MSDD) partitions of $L \times M$-PSK constellations. First we present and analyse the model of this system.

7.1 System Model.

The block diagram of the system model is shown in figure 7.2. A sequence of vectors

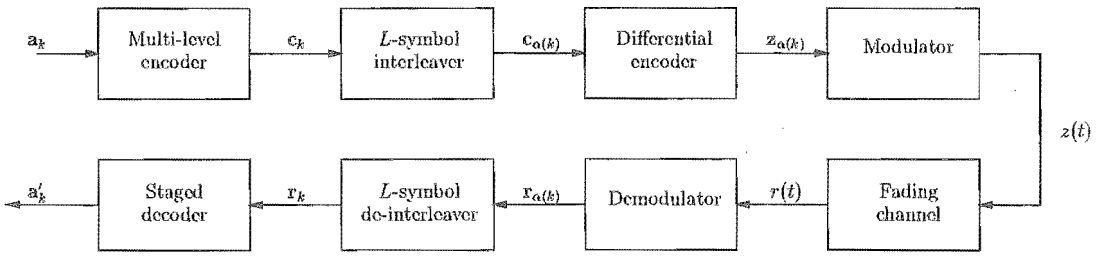


Figure 7.2: Block diagram of a multiple symbol differentially detected system, with a multi-level code.

of b bits, $\tilde{\mathbf{a}} = (\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_K)$, where $\mathbf{a}_k = (a_{k1}, a_{k2}, \ldots, a_{kb})$, and $a_{ki} = \{0, 1\}$, enters

the multi-level encoder. The output is the code sequence $\tilde{\mathbf{c}} = (\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_K)$ where each symbol $\mathbf{c}_k$ is a block of L unit energy M-PSK symbols $\mathbf{c}_k = (c_{k1}, c_{k2}, \ldots, c_{kL})$. The rate of the encoder is $r = \frac{b}{L}$ bits per two dimensional symbol. The sequence $\tilde{\mathbf{c}}$ is passed through an ideal interleaver which interleaves the blocks $\mathbf{c}_k$ as units. The output of the interleaver is the sequence of blocks $\tilde{\mathbf{c}}' = (\mathbf{c}_{\alpha(1)}, \mathbf{c}_{\alpha(2)}, \ldots, \mathbf{c}_{\alpha(K)})$, where $\alpha(k)$ is the inverse input/output map of the interleaver. Each block of L PSK symbols $\mathbf{c}_{\alpha(k)}$ is encoded differentially into the block $\mathbf{z}_{\alpha(k)} = (z_{\alpha(k)1}, z_{\alpha(k)2}, \ldots, z_{\alpha(k)L})$, where

$$z_{\alpha(k)i} = \begin{cases} z_{\alpha(k-1)L} c_{\alpha(k)1} & i = 1 \\ z_{\alpha(k)(i-1)} c_{\alpha(k)i} & 2 \leq i \leq L \end{cases} \tag{7.1}$$

In other words each code symbol is encoded as the phase advance relative to the previously transmitted M-PSK symbol. The sequence $\tilde{\mathbf{z}} = (\mathbf{z}_{\alpha(1)}, \mathbf{z}_{\alpha(2)}, \ldots, \mathbf{z}_{\alpha(K)})$ is transmitted across the flat fading Rayleigh channel with the same assumption as made in chapter 2, i.e. that there is no or negligible inter-symbol interference (ISI). The sampled demodulator output is $\tilde{\mathbf{r}}' = (\mathbf{r}_{\alpha(1)}, \mathbf{r}_{\alpha(2)}, \ldots, \mathbf{r}_{\alpha(K)})$. The deinterleaver performs the inverse process of the interleaver and its output is $\tilde{\mathbf{r}} = (\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_K)$. The deinterleaver is such that it retains the phase reference sample r_{k0} of the block $\mathbf{r}_k$. This is the last sample of the previously received block prior to deinterleaving, i.e. $r_{k0} = r_{\alpha(k-1)L}$. Each received sample r_{ki} is related to the differentially encoded symbol z_{ki} by

$$r_{ki} = z_{ki} u_{ki} + n_{ki}. \tag{7.2}$$

The n_{ki}'s are statistically independent identically distributed complex Gaussian variables with a variance

$$\sigma_n^2 = \frac{1}{2} E[|n_{ki}|^2] = N_0, \tag{7.3}$$

the one-sided power spectral density of the additive white Gaussian noise. The u_{ki}'s are a sequence of correlated, zero-mean, complex Gaussian random variables and represent the fading process experienced by the transmitted sequence. The variance of the u_{ki}'s is

$$\sigma_u^2 = \frac{1}{2} E[|u_{ki}|^2] = E_s. \tag{7.4}$$

The ratio E_s/N_0 represents the average signal-to-noise ratio (SNR) of the signal. The amplitude $|u_{ki}|$, of the complex Gaussian variables u_{ki} is Rayleigh distributed [51]. The autocorrelation of the u_{ki}'s is [43]

$$\frac{1}{2}E[u_{ki}u^*_{k(i+m)}] = \rho(m) = E_s J_0(2\pi m f_D T), \;\; 0 \leq i+m \leq L \tag{7.5}$$

where f_D is the maximum Doppler frequency, T is the symbol period, and $J_0(\bullet)$ is the zero order Bessel function. We assume a slow rate of fading ($f_D T < 0.1$) such that the differential detection process is effective in producing a channel state estimate. The received signal sequence $\tilde{\mathbf{r}}$, is passed to the decoder which produces an estimate $\mathbf{a}'$ of the transmitted binary sequence. The decoder and its performance analysis are described in the next sections.

7.2 The Maximum Likelihood (ML) Decoder.

The input to the decoder is the sequence of received L-sample blocks

$$\tilde{\mathbf{r}} = (\mathbf{r}_1, \mathbf{r}_2, \ldots, r_K), \tag{7.6}$$

and the corresponding phase reference samples $r_{k0} = r_{\alpha(k-1)L}$. The ideal interleaver/deinterleaver pair ensures that the effect of fading is independent from block to block. For the purpose of analysis we combine $\mathbf{r}_k$ and r_{k0} into the matrix R_k

$$R_k = \begin{pmatrix} r_{k0} \\ r_{k1} \\ \vdots \\ r_{kL} \end{pmatrix} \tag{7.7}$$

and the entire received sequence is represented by

$$R = \begin{pmatrix} R_1 \\ R_2 \\ \vdots \\ R_K \end{pmatrix}. \tag{7.8}$$

If the set of possible codewords of the encoding scheme is denoted by

$$\mathcal{C} = \{\tilde{\mathbf{c}} = (\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_N)\} \tag{7.9}$$

where each $\mathbf{c}_k = (c_{k1}, c_{k2}, \ldots, c_{kL})$, a sequence of $L \times M$-PSK symbols, then corresponding to each $\mathbf{c}_k$ is the differentially encoded matrix

$$Z_k = \begin{pmatrix} z_{k0} & & & \\ & z_{k1} & & \\ & & \ddots & \\ & & & z_{kL} \end{pmatrix}, \tag{7.10}$$

where $z_{ki} = z_{k(i-1)}c_{ki}$, $1 \leq i \leq L$, and z_{k0} is an arbitrary phase reference which we set to 1 for the purpose of code construction. With z_{k0} set to one, each length L differential sequence $(z_{k1}, z_{k2}, \ldots, z_{kL})$ uniquely corresponds to the code sequence $(c_{k1}, c_{k2}, \ldots, c_{kL})$. The complete differentially encoded sequence is represented by the square matrix

$$Z = \begin{pmatrix} Z_1 & & & \\ & Z_2 & & \\ & & \ddots & \\ & & & Z_K \end{pmatrix}, \tag{7.11}$$

Note that $Z^\dagger Z = ZZ^\dagger = I$. A maximum likelihood decoder will select the differential code sequence Z for which the probability $P(Z|R)$ is the largest, i.e. given the received sample sequence R, the decoder selects the code sequence Z most likely to have been transmitted. This is equivalent to choosing the sequence Z with the largest conditional probability density function,

$$p(R|Z). \tag{7.12}$$

To determine the probability density function (pdf) of $p(R|Z)$, we write the received vector from equation (7.2) in matrix form

$$R = ZU + N \tag{7.13}$$

where

$$U = \begin{pmatrix} U_1 \\ U_2 \\ \vdots \\ U_K \end{pmatrix} \tag{7.14}$$

and $U_k = (u_{k0}, u_{k1}, \ldots, u_{kL})^T$, the samples of the fading process. The additive white Gaussian noise is

$$N = \begin{pmatrix} N_1 \\ N_2 \\ \vdots \\ N_K \end{pmatrix} \tag{7.15}$$

with $N_k = (n_{k0}, n_{k1}, \ldots, n_{kL})^T$. For a fixed Z, R is the sum of two vectors of zero-mean complex Gaussian variables, each of length $K(L+1)$, and is Gaussian also. The pdf of $p(R|Z)$ is the general form of a zero-mean complex multi-variate Gaussian distribution:

$$p(R|Z) = \frac{e^{-\frac{1}{2}R^\dagger \Phi_{RR}^{-1} R}}{(2\pi)^{K(L+1)} \det(\Phi_{RR})} \tag{7.16}$$

where Φ_{RR} is the autocorrelation matrix of R:

$$\Phi_{RR} = Z\Phi_{UU}Z^\dagger + \Phi_{NN}. \tag{7.17}$$

The inverse of Φ_{RR} is

$$\begin{aligned} \Phi_{RR}^{-1} &= (Z\Phi_{UU}Z^\dagger + \Phi_{NN})^{-1} \\ &= Z(\Phi_{UU} + N_0 I)^{-1} Z^\dagger \\ &= Z\Psi Z^\dagger \end{aligned} \tag{7.18}$$

where $\Psi = (\Phi_{UU} + N_0 I)^{-1}$, and equals

$$\Psi = \begin{pmatrix} \Psi_1 & & & \\ & \Psi_2 & & \\ & & \ddots & \\ & & & \Psi_K \end{pmatrix} \tag{7.19}$$

with $\Psi_k = (\Phi_{U_k U_k} + N_0 I)^{-1}$. The determinant of Φ_{RR} is

$$\begin{aligned} \det \Phi_{RR} &= \det(Z\Phi_{UU}Z^\dagger + \Phi_{NN}) \\ &= \det Z \det(\Phi_{UU} + N_0 I) \det Z^\dagger \\ &= \det(\Psi^{-1}) \end{aligned} \tag{7.20}$$

and is independent of Z. The autocorrelation of the fading process U is

$$\Phi_{UU} = \begin{pmatrix} \Phi_{U_1U_1} & & & \\ & \Phi_{U_2U_2} & & \\ & & \ddots & \\ & & & \Phi_{U_KU_K} \end{pmatrix}, \tag{7.21}$$

where the off-diagonal terms are zero due to the block independence introduced by the interleaving and deinterleaving processes. The fading process correlation across the k^{th} block is

$$\Phi_{U_kU_k} = \begin{pmatrix} \rho_k(0) & \rho_k(1) & \cdots & \rho_k(L) \\ \rho_k(1) & \rho_k(0) & \cdots & \rho_k(L-1) \\ \vdots & \vdots & \ddots & \vdots \\ \rho_k(L) & \rho_k(L-1) & \cdots & \rho_k(0) \end{pmatrix} \tag{7.22}$$

where the function $\rho_k(m)$ is as defined in equation (7.5). Note that $\rho_k(m)$ is a function of the Doppler frequency f_D at time k and is assumed to be nearly constant over the block interval. The additive noise terms n_{ki}, of equation (7.2), are assumed to be statistically independent and identically distributed with variance N_0, and therefore

$$\Phi_{NN} = N_0 I \tag{7.23}$$

A maximum likelihood decoder selects the codeword Z that maximises equation (7.16). This is equivalent to minimising the measure

$$\begin{aligned} M(R,Z) &= R^\dagger \Phi_{RR}^{-1} R \\ &= R^\dagger Z \Psi Z^\dagger R \\ &= \sum_{k=1}^{K} R_k^\dagger Z_k \Psi_k Z_k^\dagger R_k \\ &= \sum_{k=1}^{K} M_k(R_k, Z_k) \end{aligned} \tag{7.24}$$

The term from equation (7.16) containing $\det(\Phi_{RR})$ is dropped because it is independent of Z as shown by equation (7.20) and therefore does not affect the decision. The distance measure $M(R,Z)$ is in an additive form suitable for use with the Viterbi algorithm (VA). Note that the choices of z_{k0} do not affect the value of the metric and are arbitrary. We may show that the decoding metric of equation (7.24) is a general form of the result derived by Ho and Fung [32].

7.3 The pairwise probability of error.

We investigate the probability of the ML decoder of the previous section making an incorrect decision. Let the transmitted codeword be Z. The decoder will pick the erroneous codeword Z' if

$$M(R, Z') < M(R, Z), \tag{7.25}$$

or equivalently if

$$\sum_{k=1}^{K} R_k^\dagger Z' \Psi_k Z'^\dagger_k R_k \quad < \quad \sum_{k=1}^{K} R_k^\dagger Z \Psi_k Z_k^\dagger R_k \tag{7.26}$$

where $M(R, Z)$ is the decoding measure of equation (7.24). The probability of an error event can be written as

$$P(D < 0) \tag{7.27}$$

where the decision variable D is

$$\begin{aligned} D &= \sum_{k=1}^{K} R_k^\dagger (Z' \Psi_k Z' - Z \Psi_k Z) R_k \\ &= \sum_{k=1}^{K} R_k^\dagger F_k R_k \end{aligned} \tag{7.28}$$

which is a sum of independent quadratic forms for the Hermitian matrices F_k

$$F_k = Z' \Psi_k Z'^\dagger - Z \Psi_k Z^\dagger \tag{7.29}$$

The two sided Laplace transform characteristic function of the decision variable D is the product of the characteristic functions of the random variable $R_k^\dagger F_k R_k$:

$$\Phi_D(s) = \prod_{k=1}^{K} \phi_k(s) \tag{7.30}$$

where $\phi_k(s)$ is the result from [63, Appendix B]

$$\phi_k(s) \quad = \quad \frac{1}{\det(I + 2s\Phi_{R_k R_k} F_k)} \tag{7.31}$$

The pairwise event probability can be found by the appropriate integration of the inverse Laplace transform of $\Phi_D(s)$ (which gives the probability density function of D). However, following [13] it is simpler to calculate

$$P(Z \to Z') = -\sum \text{Residue}\left[\Phi_D(s)/s\right]_{RP_{\text{poles}}} \tag{7.32}$$

instead. The notation RP_{poles} refers to the right hand plane poles of $\Phi_D(s)$. To calculate (7.32) it is useful that the $2(L+1)$ poles p_{k_i}, of (7.31) are related to the eigenvalues λ_{k_i}, of $\Phi_{R_k R_k} F_k$ by

$$p_{k_i} = \frac{-1}{2\lambda_{k_i}} \tag{7.33}$$

and equation (7.30) can be written as

$$\Phi_D(s) = \prod_{k=1}^{K} \prod_{i=1}^{2(L+1)} \frac{-p_{k_i}}{s - p_{k_i}} \tag{7.34}$$

In the next section we examine the behaviour of the pairwise error probability as a function of the normalised fade rate $f_D T$ of the system.

7.4 Pairwise error behaviour.

The equations of section 7.3 analytically describe the pairwise probability of error of codewords encoded differentially and transmitted across the Rayleigh fading channel. Unfortunately they do not give a clear insight, by inspection, into the behaviour of the probability of error as a function of the parameters $f_D T$, SNR, L, and code sequences Z and Z'. Therefore we will illustrate the behavioural trends empirically. First we examine the error probability as a function of the number of symbols L over which the differential detection is applied. In figure 7.3 we have plotted the pairwise probability of error as a function of the SNR of sequences of length $L = 1$ (conventional differential detection), $L = 2$, $L = 3$, $L = 5$, $L = 10$ and for comparison the case of coherent detection. The sequences are of the form

$$\tilde{\mathbf{c}} = \{0, 0, \ldots, 0\} \tag{7.35}$$

$$\tilde{\mathbf{c}}' = \{1, 0, \ldots, 0\} \tag{7.36}$$

and have a Hamming distance of one. The code symbols are drawn from a 4-PSK constellation. The parameter $K = 1$, i.e. we are considering the behaviour of just one block of symbols, and the fade rate is $f_D T = 0.03$ for all curves. The conventional differential detection case ($L = 1$) quickly reaches an error floor at a probability of error of about 9×10^{-3}. By increasing the length of the multiple symbol differential detection to $L = 2$, the error floor drops significantly to about 8×10^{-5}. For larger values of L the error floor is eliminated for practical values of

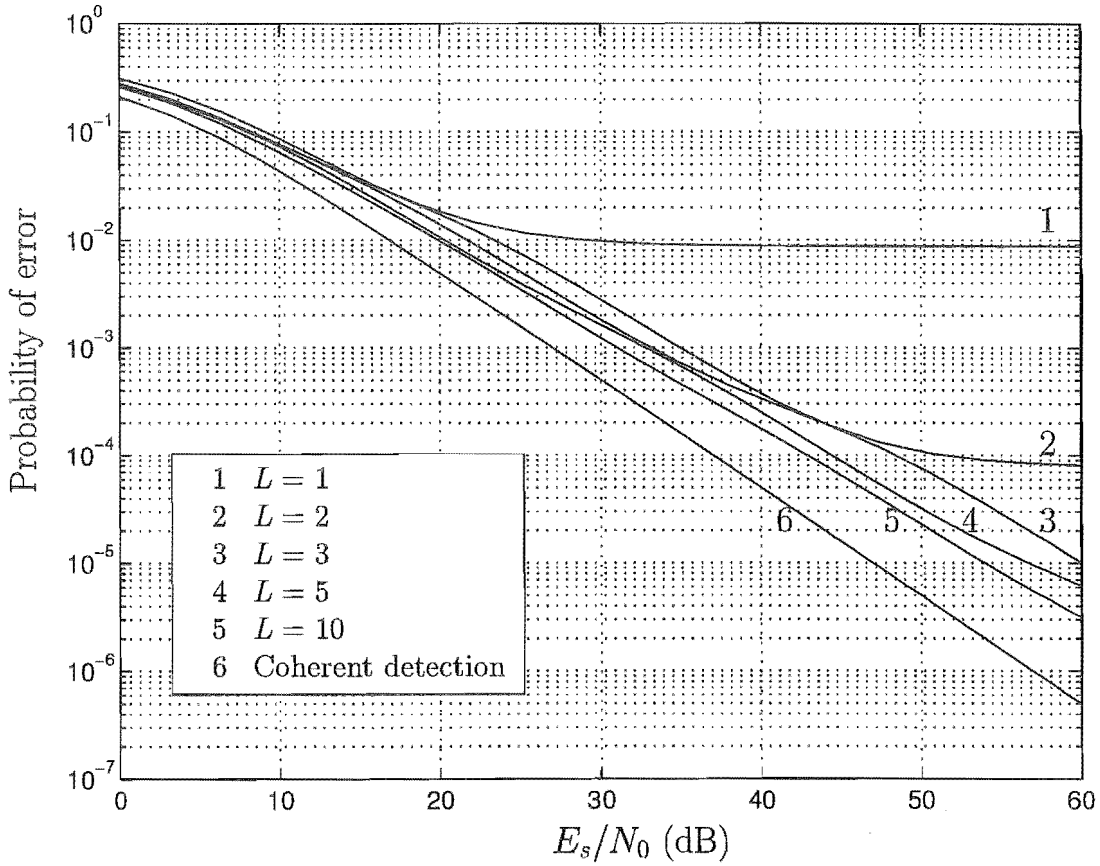


Figure 7.3: Performance curves of a multiple symbol differentially detected system operating on the Rayleigh fading channel with $f_D T = 0.03$.

SNR (there will be an error floor at sufficiently high SNR). As L is increased the error curves tend toward but at a loss compared to coherent detection. The loss is a function of the fade rate parameter $f_D T$. Clearly multiple symbol differential detection eliminates the problem of the error floor, for low values of $f_D T$. Next we examine the behaviour of the probability of error as a function of the fade rate $f_D T$. In figure 7.4 we have plotted the probability of error of the same pairs of sequences as a function of the fade rate $f_D T$ at a SNR of 30 dB. We see that for each case the probability of error is a monotonically increasing function with $f_D T$. At low $f_D T$ each curve reaches a steady state probability of error. The undulation of the curves is due to the oscillatory nature of the correlation function.

In figure 7.5 we consider sequences of the same lengths but with Hamming dis-

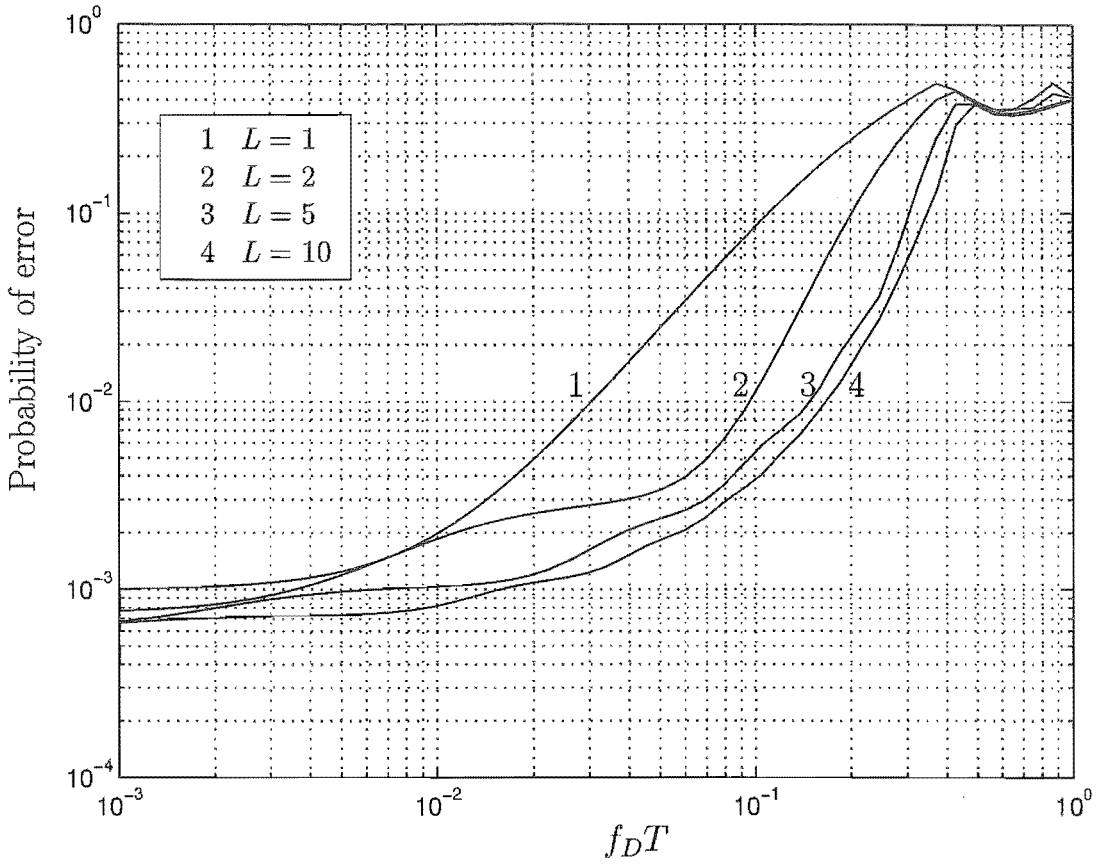


Figure 7.4: Performance curves of a multiple symbol differentially detected system operating on the Rayleigh fading channel with $E_s/N_0 = 30$ dB.

tance two, and of the form:

$$\tilde{\mathbf{c}} = \{0, 0, \ldots, 0, 0\} \tag{7.37}$$

$$\tilde{\mathbf{c}}' = \{1, 0, \ldots, 0, 1\} \tag{7.38}$$

i.e. sequences beginning and ending in a non-zero symbol. The behaviour of the curves for $L > 2$ is now quite different. As f_DT increases, the probability of error actually decreases until about $f_DT \approx 0.03$, after which it again rapidly increases. The reduction in error with increasing f_DT is due to the *diversity* of the codewords starting to play a part. As f_DT increases the correlation between the fading affecting the symbols in the codeword decreases (the placement of the non-zero symbols on the ends of the codewords minimises the correlation) and the effect of diversity takes place. However counteracting this phenomena is the decrease in channel state accuracy with increasing f_DT until at sufficiently fast fading the probability of error increases again. Clearly communication becomes essentially impossible for $f_DT >$

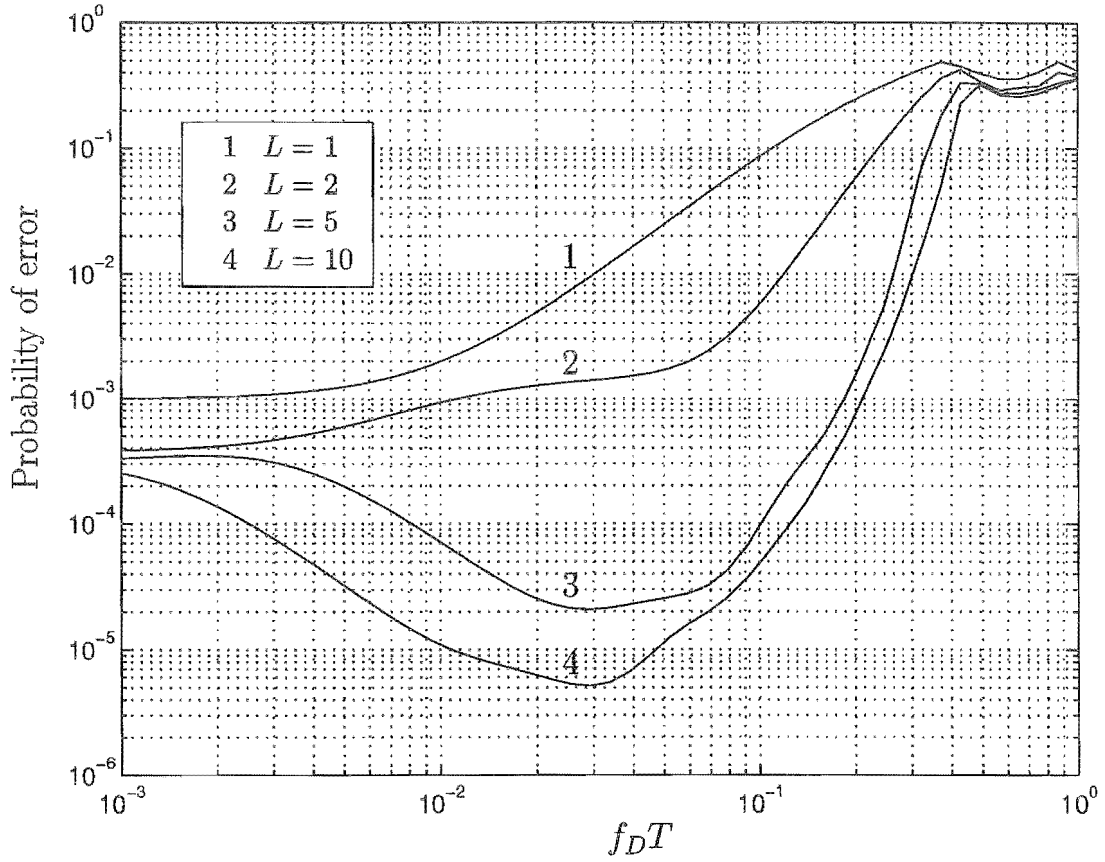


Figure 7.5: Performance curves of a multiple symbol differentially detected system operating on the Rayleigh fading channel with $E_s/N_0 = 30$ dB and sequences with Hamming distance of two.

0.5.

7.4.1 Error performance for slow fading.

From inspection of the curves of figure 7.4 and 7.5 it is not easy to relate the properties of the code sequences to the probability of error of a differentially detected system. We can derive such a relationship however for the case of slow fading. In the limit of $f_DT \to 0$, equation (7.31) equals

$$\phi_k(s) = \frac{p_{k_1}p_{k_2}}{(s - p_{k_1})(s - p_{k_2})} \tag{7.39}$$

i.e. there are just two finite poles. The product $p_{k_1}p_{k_2}$ is

$$p_{k_1}p_{k_2} = \frac{E_s/N_0(L+1)+1}{(E_s/N_0)^2\delta_{z_k}^2} \tag{7.40}$$

where E_s/N_0 is the SNR of the system, and the term $\delta^2_{z_k}$ is given by:

$$\delta^2_{z_k} = \sum_{n=0}^{L} \sum_{m=n+1}^{L} |z_{kn} z^*_{km} - z'_{kn} z'^*_{km}|^2. \tag{7.41}$$

Combining equations (7.30) and (7.39) and assuming high SNR, the characteristic function, $\Phi_D(s)$ is approximated by

$$\Phi_D(s) \approx \prod_{\substack{k=1 \\ Z_k \neq Z'_k}}^{K} \frac{1}{d^2_{z_k} E_s/N_0} \frac{1}{(s - p_{k_1})(s - p_{k_2})} \tag{7.42}$$

where $d^2_{z_k} = \frac{\delta^2_{z_k}}{L+1}$. Using equation (B.7) of appendix B for high SNR, the pairwise probability, $P(Z \to Z')$ is tightly bounded by

$$P(Z \to Z') \leq \frac{(2l-1)!}{l!(l-1)!} \prod_{\substack{k=1 \\ Z_k \neq Z'_k}}^{K} \frac{1}{(E_s/N_0) d^2_{z_k}} \tag{7.43}$$

where l is the number of code blocks Z_k and Z'_k different between Z and Z'. Note the analogy between equation (7.43) and equation (2.73) of chapter 2 describing the probability of error of an infinitely interleaved system with ideal CSI. If η is the set of k such that $Z_k \neq Z'_k$ then we define the differential squared product distance as

$$d^2_{p_z} = \prod_{k \in \eta} d^2_{zk} \tag{7.44}$$

A good code design aims to maximise the Hamming distance $l = |\eta|$ and the differential product distance $d^2_{p_z}$.

Uncoded transmission.

Consider now the special case of uncoded transmission in a very slowly fading channel. In general the probability of error is given by equation (7.43). In an uncoded system the probability of error is dominated by the sequence with the smallest differential squared product distance relative to the all-zero codeword. For M-PSK these are the sequences with one non-zero symbol, i.e.

$$\tilde{\mathbf{c}} = \{0, 0, \ldots, 0\} \tag{7.45}$$

$$\tilde{\mathbf{c}}' = \{1, 0, \ldots, 0\} \tag{7.46}$$

and the pairwise probability of error is

$$p(\tilde{\mathbf{c}} \to \tilde{\mathbf{c}}') \leq \frac{1}{E_s/N_0 d^2_{p_z}} \tag{7.47}$$

where the squared differential distance between $\tilde{\mathbf{c}}$ and $\tilde{\mathbf{c}}'$ is

$$d_{p_z}^2 = \frac{Ld^2}{L+1} \tag{7.48}$$

where L is the length of the sequences, and d^2 is the squared Euclidean distance between the symbols 0 and 1. In comparison to coherent detection, the pairwise probability of error from chapter 2 is

$$p(\tilde{\mathbf{c}} \to \tilde{\mathbf{c}}') \leq \frac{1}{E_s/N_0 d^2} \tag{7.49}$$

where d^2 is the squared Euclidean distance between the sequences $\tilde{\mathbf{c}}$ and $\tilde{\mathbf{c}}'$. From equation (7.48) the performance of MSDD for large L tends towards coherent detection. The loss relative to coherent detection is

$$\gamma = 10 \log_{10}(\frac{L+1}{L}) \text{ dB} \tag{7.50}$$

and shows that as $L \to \infty$, MSDD performs identically to coherent detection.

7.5 Geometrically uniform code design.

We desire to design codes for the differentially detected Rayleigh fading channel with the property of geometric uniformity. To start we need to determine the relevant distance measures for the system and the associated isometries.

7.5.1 Distance measures and isometries for MSDD.

From equation (7.24) the decoding measure between the received vector R and the postulated codeword Z is

$$\begin{aligned} M(R,Z) &= \sum_{k=1}^{K} R_k^\dagger Z_k \Psi_k Z_k^\dagger R_k \\ &= \sum_{k=1}^{K} M_k(R_k, Z_k) \end{aligned} \tag{7.51}$$

where $M_k(X,Y) = X^\dagger Y \Psi_k Y^\dagger X$, a distance measure between X and Y (X is a vector and Y a diagonal matrix). Using the same argument as used for general product distance, the isometries of $M(R,Z)$ are the permutations of the elements R_k and Z_k, and the mappings $u(X)$ of the form

$$u(X) = \{u_1(X_1), u_2(X_2), \ldots, u_K(X_K)\} \tag{7.52}$$

where each u_k is an isometry mapping relative to M_k. Consider the mapping $u_k = AX$ with A such that

$$A^\dagger A = AA^\dagger = I \tag{7.53}$$

i.e., a unitary matrix. The distance measure M_k is invariant under u_k because

$$\begin{aligned} M_k(u_k(X), u_k(Y)) &= (AX)^\dagger AY\Psi_k(AY)^\dagger AX \\ &= X^\dagger A^\dagger AY\Psi_k Y^\dagger A^\dagger AX \\ &= X^\dagger Y\Psi_k Y^\dagger X \\ &= M_k(X,Y) \end{aligned} \tag{7.54}$$

Hence $u_k(X) = AX$ is an isometry relative to the decoding distance measure M_k.

The probability of confusing codeword Z' for Z is completely described by the characteristic function of the decision variable D

$$\begin{aligned} \Phi_D(s) &= \prod_{k=1}^{K} \frac{1}{\det(I + 2s\Phi_{R_kR_k}F_k)} \\ &= \prod_{k=1}^{K} \frac{1}{\det(I + 2s(Z_k\Psi_k^{-1}Z_k^\dagger Z_k'\Psi_k Z'^\dagger - I))} \end{aligned} \tag{7.55}$$

From which we may deduce a performance distance measure polynomial to be

$$P_k(Z_k, Z_k', s) = \det(I + 2s(Z_k\Psi_k^{-1}Z_k^\dagger Z_k'\Psi_k Z_k'^\dagger - I)) \tag{7.56}$$

We can show that this distance measure is invariant under the mapping $u_k(X) = AX$ also

$$\begin{aligned} P_k(AZ_k, AZ_k', s) &= \det(I + 2s(AZ_k\Psi_k^{-1}(AZ_k)^\dagger AZ_k'\Psi_k(AZ_k')^\dagger - I)) \\ &= \det(I + 2sA(Z_k\Psi_k^{-1}Z_k^\dagger Z_k'\Psi_k Z_k'^\dagger - I)A^\dagger) \\ &= \det(A)\det(I + 2s(Z_k\Psi_k^{-1}Z_k^\dagger Z_k'\Psi_k Z_k'^\dagger - I))\det(A^\dagger) \\ &= \det(I + 2s(Z_k\Psi_k^{-1}Z_k^\dagger Z_k'\Psi_k Z_k'^\dagger - I)) \\ &= P_k(Z_k, Z_k', s) \end{aligned} \tag{7.57}$$

Therefore both the decoding measure and the error performance measures are invariant under the mapping described by equation (7.52).

7.5.2 Symmetries of L × M-PSK relative to MSDD.

In the previous section we have shown that the isometries of an $L+1$ symbol sequence Z_k for differential encoding and decoding are of the form $u(Z_k) = AZ_k$ where A is a unitary matrix. Each sequence Z_k of length $L+1$ has z_{k0} arbitrarily set to 1 and therefore each each Z_k corresponds uniquely to a length L sequence $\mathbf{c}_k$. Our code design will be over the length L sequences $\mathbf{z}_k$ which map to the corresponding code sequences $\mathbf{c}_k$ by

$$c_{ki} = \begin{cases} z_{ki} & i = 1 \\ z_{ki} z^*_{k(i-1)} & i \geq 2 \end{cases} \tag{7.58}$$

The symmetries of a constellation S is the set of isometries which maps S to itself. From equation (7.53) we deduce that the symmetries of S are

1. Rotations.

2. Permutations.

3. Compositions of rotations and permutations.

Note that reflections are not symmetries under MSDD.

7.6 Multi-level codes.

We have selected multi-level codes as a method of constructing diversity between the sequences Z and Z'. The multi-level codes are based on binary partitions of $L \times M$-PSK constellations and as such all of the results of chapter 5 on geometrically uniform partitions hold. The binary partitions are based on generating groups constructed only from rotation symmetries (not the permutation symmetries) of the underlying constellation. The properties of the partition are not very important to the code performance because the inherent diversity in the partition cannot be exploited, and the code is dominated by the bottom level partition. Note that the codes are constructed over the sequences $\mathbf{z}_k$ and not over $\mathbf{c}_k$. The final encoder needs to be mapped to $\mathbf{c}_k$ using the mapping of equation (7.58).

$S = 2 \times$ 8-PSK, Partition I					
i	G_i/G_{i-1}	m_i	Coset	$d^2_{p_z}$	N
1	Z_2	2	(r_4, r_4)	2.67	1
2	Z_2	2	(r_2, r_6)	2.67	2
3	Z_2	2	(r_0, r_4)	1.33	2
4	Z_2	2	(r_1, r_3)	1.06	2
5	Z_2	2	(r_4, r_2)	0.39	2
6	Z_2	2	(r_2, r_3)	0.39	4

Table 7.1: Binary partitions of $2 \times$ 8-PSK.

7.6.1 $2 \times$ 8-PSK ML codes.

Table 7.1 shows a partitioning of $S = 2 \times$ 8-PSK. It is a six level binary partition and the distance properties of $d^2_{p_z}$ and multiplicity N are listed in the fifth and sixth columns respectively. As is typical of partition chains, the distance properties decrease at the lower level partitions. We investigate using identical rate 2/3 binary encoders over each level to improve upon the code properties. This selection ensures that the diversity of each level is identical, and gives an overall spectral efficiency of 2 bits per symbol. In table 7.2 we have listed the code properties of the six component codes for a selection of $v = 3$. In figure 7.6 we have plotted the union bound

Constellation $S = 2 \times$ 8-PSK. Partition I.													
i	m_i	$\mathcal{A}_i$	p_i	n_i	r_i	v_i	d_H	$d^2_{p_z}$	N	$\xi_A(C_i)$	$\xi_P(C_i)$	$\xi_T(C_i)$	$\xi(C_i)$
1	2	Z_2	3	2	$\frac{2}{3}$	3	4	50.6	2	0	0	16	16
2	2	Z_2	3	2	$\frac{2}{3}$	3	4	50.6	32	2	0	16	18
3	2	Z_2	3	2	$\frac{2}{3}$	3	4	3.16	32	4	0	16	20
4	2	Z_2	3	2	$\frac{2}{3}$	3	4	1.26	32	8	0	16	24
5	2	Z_2	3	2	$\frac{2}{3}$	3	4	0.023	32	8	0	16	24
6	2	Z_2	3	2	$\frac{2}{3}$	3	4	0.023	512	8	0	16	24
Code rate =					2.0	Total complexity							126

Table 7.2: Summary of the properties of a multi-level code design over partition I of $2 \times$ 8-PSK. Design distance $d_H = 4$.

on the pairwise error probability of each of six component codes of table 7.2. The

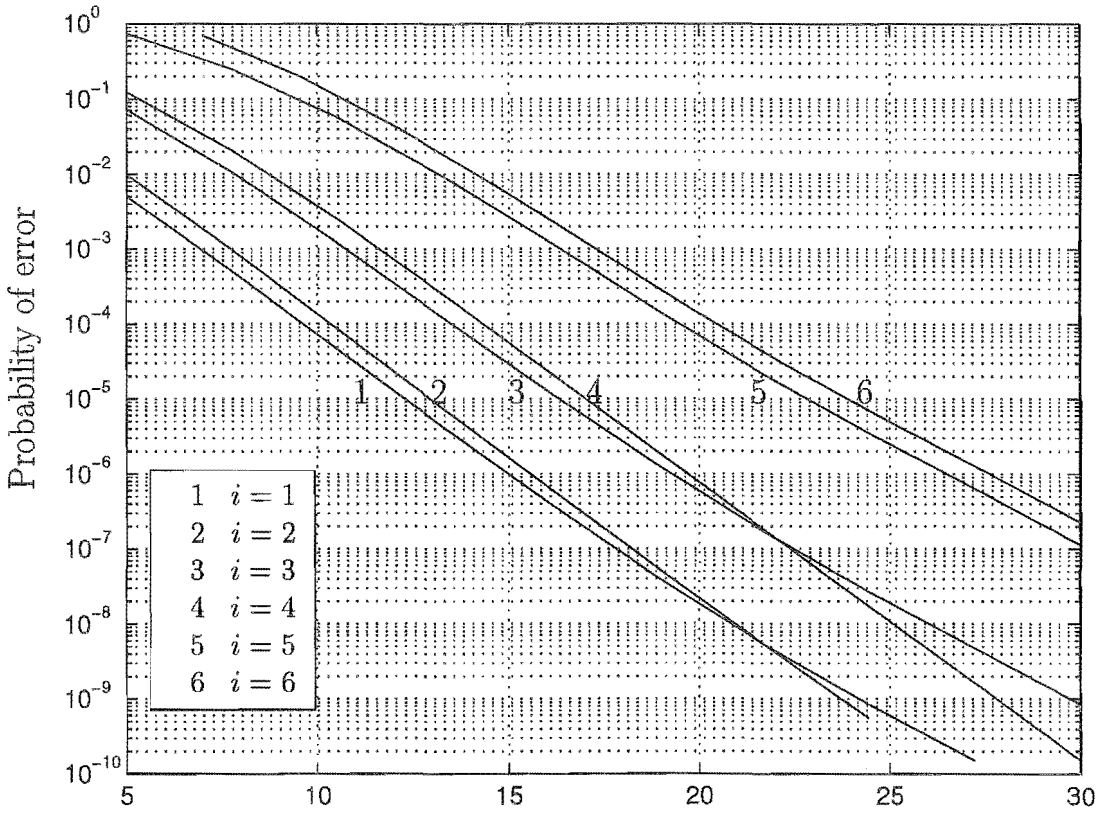


Figure 7.6: Performance curves of the component codes of a six level multi-level code over 2×8-PSK using six rate 2/3 binary convolutional encoders each with $v = 3$. $f_D T = 0.01$.

overall multi-level code is dominated by the bottom level component code. The fact that the component codes over levels $i = 1$ and $i = 2$ perform much better than the component code over level $i = 6$ means that the constraint length of the component codes over $i = 1$ and $i = 2$ may be reduced to $v = 2$ while still outperforming the level $i = 6$ component code. In practice short-length linear binary block codes may be more desirable as a means of minimising the decoding delay associated with convolutional codes.

The performance of the multi-level codes can be improved by increasing the constraint lengths of the component codes. In figure 7.6 we have plotted the probability of error of the level $i = 6$ component code of the multi-level construction over 2×8-PSK as a function of the constraint length of the encoder, for a fade rate of 0.01. Each increase in encoder constraint length produces a diminishing increase in coding

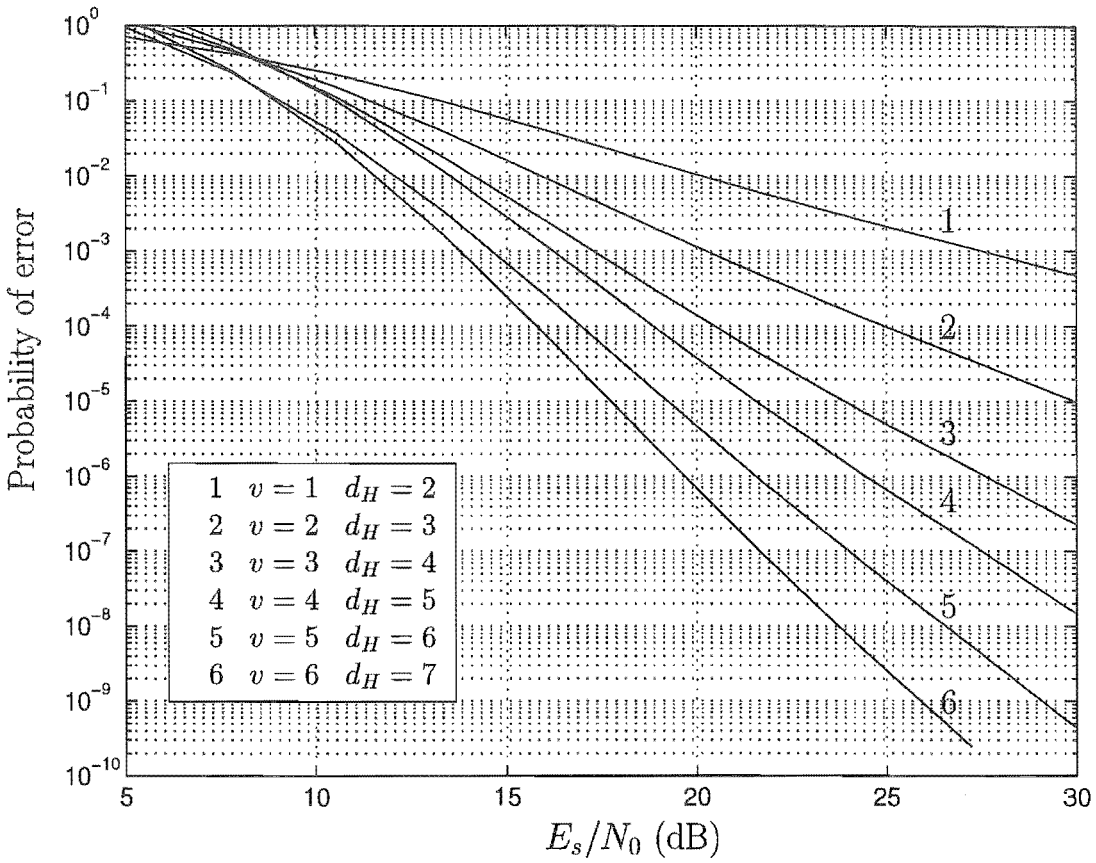


Figure 7.7: Performance curves of the dominant component code of a six level multi-level code over 2 × 8-PSK as a function of the constraint length v of the encoder. $f_DT = 0.01$.

gain. The gains over uncoded MSDD are easily over 20 dB at a PER of 10^{-4} and much larger at a PER of 10^{-8}. To determine the performance penalty relative to coherent detection we compare the differential squared product distance with the squared product distance d_p^2 of the 2×8-PSK partitions of chapter 6. For the bottom level partition the squared differential product distance is $d_{p_z}^2 \approx 0.39$ in comparison to $d_p^2 = 0.587$ for the same partition, a factor of $\frac{3}{2}$ different. From equation (7.50) this means the loss in dB relative to coherent detection for slow fading is

$$10\log_{10}(\frac{3}{2}) \approx 1.77\,\text{dB} \tag{7.59}$$

For faster fading the loss is far larger. To quantify the loss we have plotted the performance of the level $i = 6$ component code of table 7.2 as a function of f_DT in figure 7.8. For comparison the performance with coherent detection is also plotted. The performance at fade rates up to $f_DT = 0.003$ is good. Although for $f_DT = 0.01$, there is no error floor, the error performance is deteriorating at high SNR. For

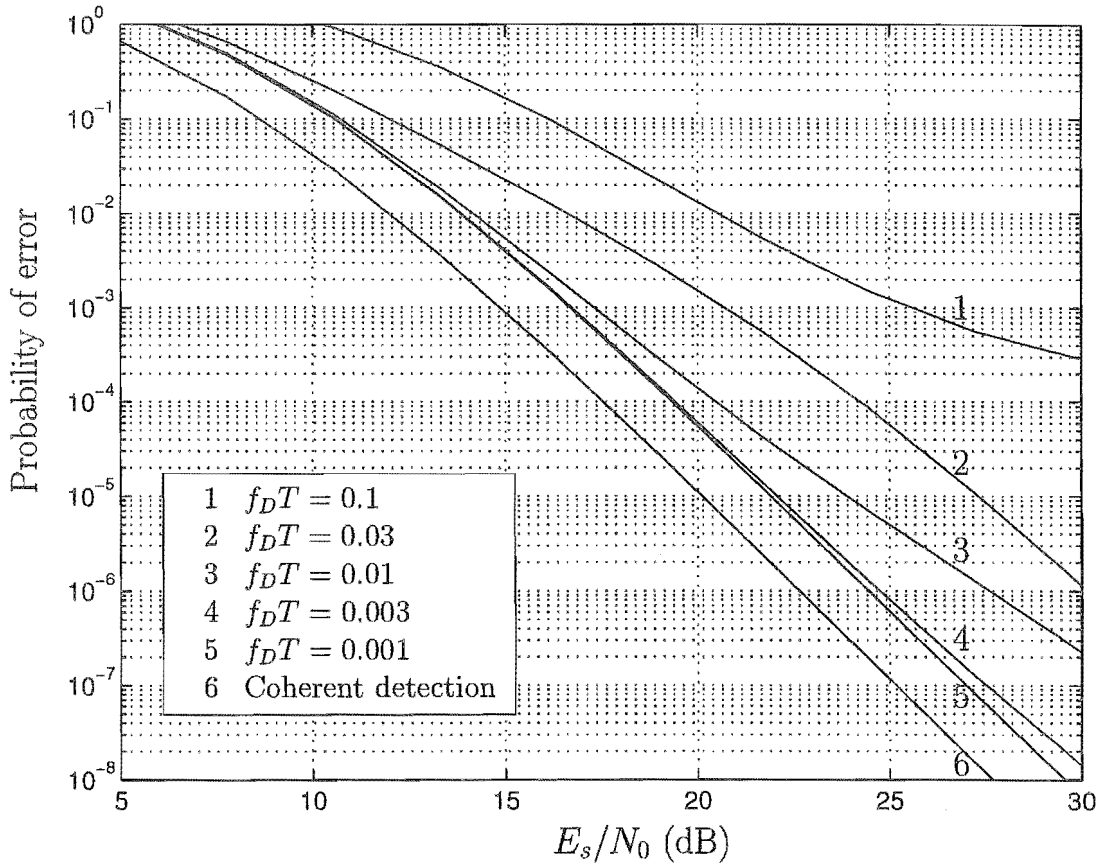


Figure 7.8: Performance curves of the dominant error component code of a six level multi-level code over 2 × 8-PSK as a function of f_DT. $v = 3$.

$f_DT = 0.1$ an error floor is appearing and the error performance is poor relative to slower fading. We may improve upon the performance at higher fade rates by increasing the number of symbols over which MSDD detection is applied.

$S = 3 \times 8$-PSK, Partition I					
i	G_i/G_{i-1}	m_i	Coset	$d^2_{p_z}$	N
1	Z_2	2	(r_4, r_4, r_0)	4.00	1
2	Z_2	2	(r_2, r_6, r_4)	4.00	2
3	Z_2	2	(r_1, r_7, r_6)	2.29	2
4	Z_2	2	(r_0, r_4, r_0)	2.29	2
5	Z_2	2	(r_2, r_6, r_0)	2.00	2
6	Z_2	2	(r_4, r_2, r_4)	1.09	2
7	Z_2	2	(r_2, r_3, r_6)	0.44	2
8	Z_2	2	(r_2, r_3, r_0)	0.44	2
9	Z_2	2	(r_4, r_1, r_5)	0.44	4

Table 7.3: A binary partition of 3×8-PSK.

7.6.2 3×8-PSK ML codes.

We can improve upon to the tolerance of the system to faster fading by increasing L, the number of symbols over which MSDD is applied. In table 7.3 we have listed a nine level partitioning of 3×8-PSK and the associated distance properties of the partitions. Note that the bottom level partition has a slightly higher differential squared product distance compared to 2-PSK. As mentioned only the distance properties of the bottom partition are of importance and if we construct a multilevel code with nine binary rate 2/3 encoders then the performance will be only slightly better than 2×8-PSK at low fade rates because the distance properties of the bottom partition are almost identical and the performance curves as a function of v, the constraint length of the component codes will be similar to those of figure 7.7. However at faster fade rates the multiple symbol differential detection is more effective over three symbols than over two and this is illustrated in figure 7.9 where we have plotted the error performance as a function of the fade rate f_DT in a similar manner to figure 7.8 for 2×8-PSK codes. We see that the codes over 3×8-PSK easily cope with fade rates up to $f_DT = 0.03$ and only lose about 2.5 dB at the very fast fading rate of $f_DT = 0.1$. Beyond a rate of fading of $f_DT = 0.1$ the performance deteriorates quickly and error floors become apparent. For slow fading ($f_DT < 0.3$)

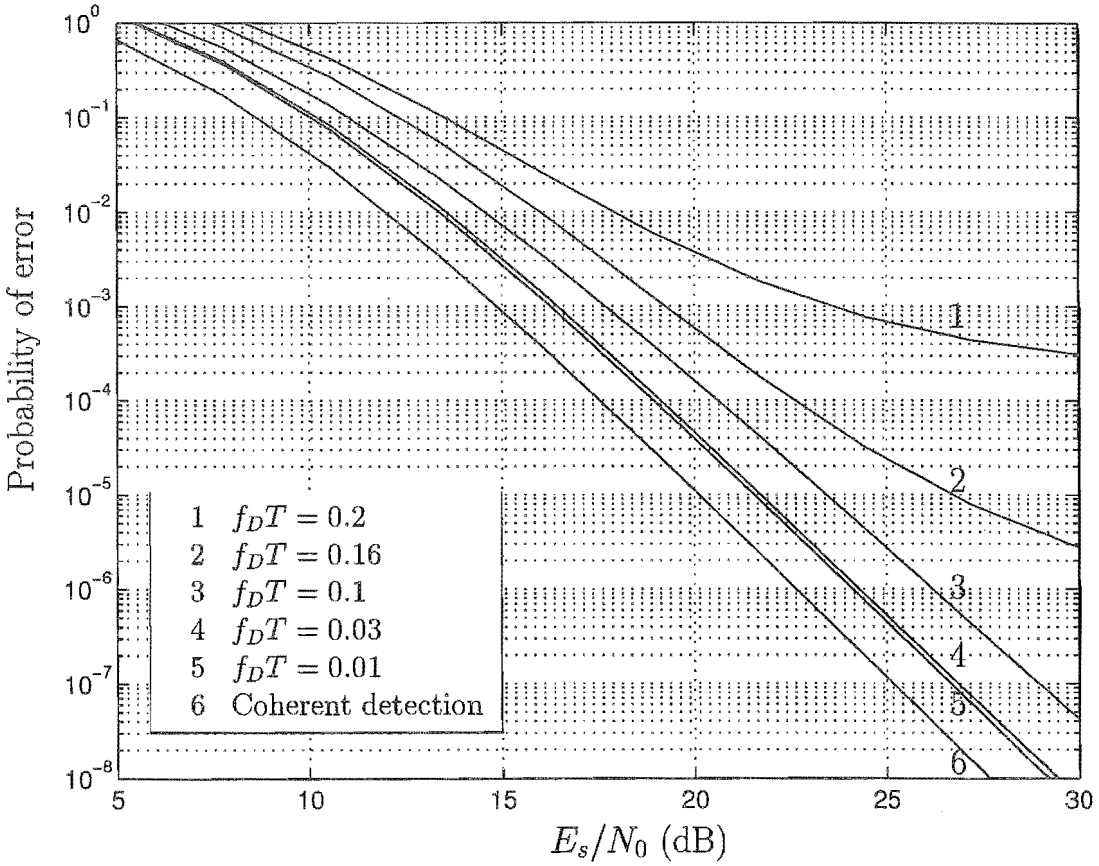


Figure 7.9: Performance curves of the dominant component code of a six level multi-level code over 3×8-PSK as a function of the fade rate f_DT. $v = 3$.

the loss relative to coherent detection is only

$$10 \log_{10}(\frac{4}{3}) \approx 1.25 \text{ dB} \tag{7.60}$$

7.7 Summary.

We have investigated the combination of multiple symbol differential detection and multi-level coding as a means of attaining a channel state estimate in practice while still achieving a good error performance. MSDD very effectively eliminates the error floor associated with conventional differential detection by making a decision on a block of L symbols. The need for maintaining the symbols in blocks for the purpose of detection means the interleaving required for coding cannot be on a symbol by symbol basis and instead we interleave over the blocks of L symbols. The construction of a multi-level code over the blocks gives the coding gain we

require to improve performance. The same flexibility in the multi-level code design is not available for MSDD compared to coherent detection of chapter 6 because the Hamming distance within the partition levels does not count toward coding gain. We have therefore only investigated multi-level codes with binary component codes over 2×8 and 3×8-PSK constellations for the purpose of illustrating the behaviour of the error performance as a function of the system parameters.

Chapter 8

Conclusions and future work.

In this chapter we discuss the main conclusions of the work presented and consider some future work issues.

8.1 Conclusions.

The first chapter introduced the aetiology of the Rayleigh fading channel model. We presented the channel capacity bound derived by W.C.Y. Lee [46] on the Rayleigh fading channel and showed that uncoded modulation, even with ideal channel state information, performs very poorly relative to the capacity of the channel. This is due to the probability of error of uncoded modulation decreasing only linearly with the SNR, compared to decreasing exponentially for the case of the classical AWGN channel. Quantitively it means uncoded modulation performs 22 dB away from capacity at an error probability of only 10^{-4}, and about 72 dB at the probability of error of 10^{-9} required for reliable digital communications. This observation justifies the need for more sophisticated signalling transmission and decoding techniques.

The analysis of chapter 2 shows that through the technique of maximum likelihood sequence estimation (MLSE), where a decision is made on a *sequence* of symbols rather than on a symbol by symbol basis, the probability of error drops inversely to the SNR raised to the power of the Hamming distance of the code sequences. This effect is referred to as *code diversity*. Therefore we may control the slope of the probability of error curve through the application of a high Hamming distance code and the corresponding ML decoder. Paramount to the success of such a code however is

the process of interleaving. It has been found that the fading affecting the symbols in the code sequences must be independent in order for coding to be effective. One method of meeting this requirement is through the process of sequence interleaving at the transmitter and the through applying the inverse operation of deinterleaving at the receiver. The consequence of the interleaving and deinterleaving process is the introduction of an irreducible transmission delay proportional to the minimum fade rate of the system and the diversity of the code. The only method of reducing this delay is by introducing forms of diversity, such as space or frequency diversity, that are not dependent on time.

The technique we examined for the construction of codes for the Rayleigh fading channel was multi-dimensional trellis coding. With such a coding scheme, a multiple number of symbols, typically from two to four, are transmitted per trellis branch. At a spectral efficiency of 2 bits per symbol the resulting trellis typically has a large number of parallel branches, the dominant error event usually contained therein. The upperbound on the attainable Hamming distance of multiple-symbol-per-branch derived in chapter 3 shows the maximum Hamming distance as a function of the rate r in bits per symbol, L the number of symbols per branch and v the constraint length of the code. This function shows that the upperbound on Hamming distance does not necessarily increase with increasing L or v and that there are a number of tradeoffs available to the code designer.

A class of multi-dimensional trellis codes are geometrically uniform trellis codes. These codes are defined over $L \times M$-PSK constellations and have the desirable property of uniform error probability (UEP) which implies the code properties may be entirely characterised by considering only one code word. This property leads to reduced code search efforts and reduced complexity decoding structures. The multi-dimensional codes discovered appear to be superior to those reported in the literature for the same number of code states. However when the decoding complexity is taken into account this is not necessarily the case. Most of the codes discovered meet the bound on Hamming distance of chapter 3 with equality. It was found that expanding the signal set by more than a factor of two relative to that required for uncoded modulation reaps rewards in terms of the resulting minimum squared product distance, and in some cases in terms of the minimum Hamming distance of the codes. This is in contrast to the AWGN channel, for which Ungerboeck [75]

showed that most of the coding gain is achieved expanding the signal set by a factor of two.

The extension of the theory of the properties of geometric uniformity from points to partitions of signal constellations allows for less constrained structures to be considered. We may now define geometric uniformity of the partitions of non-geometrically uniform constellations such as 16-QAM. The definition of geometrical uniformity of partitions allows for a convenient construction of multi-level codes such that each code over each partition has the uniform error property. The overall code however is not necessarily GU. The advantage of multi-level codes is the ability to decode them in a staged manner. Although staged decoding is not maximum likelihood, it does represent a significant reduction in decoding complexity. To measure the decoding complexity we defined a normalised decoding complexity measure that takes into account the need for the decoding of the partition sets, the parallel branches of the code and the trellis code itself. We have presented multi-level codes over partitions of multi-dimensional constellations, which outperform trellis codes in terms decoding complexity. High diversity codes are easily obtained with multi-level coding at a moderate complexity.

The work on trellis codes and multi-level codes has assumed ideal channel state information for the purpose of code performance comparison. A real-world method of obtaining channel state information is through differential detection. Conventional differential detection is known to perform poorly on the Rayleigh fading channel due to the fluctuation of the channel state with time. The works by Ho and Fung [32] and Divsalar and Simon introduce the method of multiple symbol differential detection (MSDD), a technique where a decision is made on a sequence of symbols, rather than on a per symbol basis. MSDD vastly improves performance over conventional differential detection even when decoding over just a few symbols, by exploiting knowledge of the correlation of the fading process. MSDD works best when the channel fading is highly correlated - a contradicting requirement to diversity. We circumvent this problem by interleaving blocks of symbols over which the MSDD is computed, and constructing a multi-level code over these blocks of symbols. The resulting multi-level codes achieve high coding gains and can perform very well in relatively fast fading environments, that is, values of $f_D T$ as high as 0.1.

8.2 Future work.

There are many areas in which further work is required. Some of the areas and ideas we have envisaged are:

1. The derivation of an upperbound on the pairwise probability of error of pilot aided detection including channel correlation. This case was not derived in chapter 2, however both the cases of pilot tone aided detection assuming ideal interleaving, and channel correlation assuming ideal CSI have been presented. We imagine the amalgamation of the two should be possible.

2. Further refinement of the upperbound on the obtainable Hamming distance of multi-dimensional trellis codes. The bound presented in chapter 3 makes certain assumptions on the number of symbols in the underlying constellation. The question is how does this number affect the validity of the bound?

3. It was found in chapter 4 that the code search effort generally increases very rapidly with increasing number of states and symbols per trellis branch. An area of research that needs to addressed is the reduction of code search effort. In the code search algorithms used, the symmetries of the underlying constellations were exploited to reduce the code search effort, however the symmetries inherent in the encoder itself were not. This is an area that can potentially yield a few orders of magnitude of reduction in code search effort.

4. In chapter 5 we examined some of the decision region issues of geometrically uniform partitions. An open topic is to establish the conditions on partitions of geometrically partitions such that their decision regions are invariant to the SNR of the system. The invariance of the decision regions to SNR means the decoder does not require knowledge of the current value of SNR.

5. For the construction of multi-level codes over $L \times M$-PSK constellations we needed to partition the multi-dimensional constellation into a partition chain. For small constellations good partitions can be discovered through a brute force search approach. However for values of L and M larger than about four and eight respectively, the search space becomes prohibitively large. It is envisaged that good partition chains may be constructed through the application of block code type constructions and further investigation is required in this area.

6. From chapter 6 it is clear that the reduced complexity coding technique of multi-level codes can easily outperform trellis codes in terms of probability of error for the same level of decoder complexity. We feel that other reduced complexity techniques, such as turbo codes, have the potential to perform even better and we would suggest such an investigation.

7. Finally, the gains obtained on the Rayleigh fading channel are obtained through the exploitation of time diversity through coding. The penalty for this gain is an irreducible time delay in the transmission of the data. An alternative source of diversity is frequency diversity, characteristic of the wide band mobile channel. The exploitation of frequency diversity in the mobile radio channel through the techniques of coding can lead to a low delay and low error rate communication system.

Appendix A

Matrix inverse identity.

A very useful identity for the inverse of a partitioned matrix

$$A = \begin{pmatrix} P & Q \\ R & S \end{pmatrix} \tag{A.1}$$

is, if P is non-singular,

$$A^{-1} = \begin{pmatrix} X & -P^{-1}QW \\ -WRP^{-1} & W \end{pmatrix} \tag{A.2}$$

where $W = (S - RP^{-1}Q)^{-1}$ and $X = P^{-1} + P^{-1}QWRP^{-1}$. If S is non-singular then the inverse of A is also given by

$$A^{-1} = \begin{pmatrix} X & -XQS^{-1} \\ -S^{-1}RX & W \end{pmatrix} \tag{A.3}$$

where now $W = S^{-1} + S^{-1}RXQS^{-1}$ and $X = (P - QS^{-1}R)^{-1}$. If both P and S are non-singular then

$$(S - RP^{-1}Q)^{-1} = S^{-1} + S^{-1}RXQS^{-1} \tag{A.4}$$

and

$$(P - QS^{-1}R)^{-1} = P^{-1} + P^{-1}QWRP^{-1}. \tag{A.5}$$

The proofs of these identities is from the text by Noble and Daniel [50].

Appendix B

Residue calculation.

We want to find the closed form expression for S_r defined by

$$S_r = \sum \text{Residue}[\frac{\Phi(s)}{s}]_{\text{RP}_{\text{poles}}} \tag{B.1}$$

for the case of

$$\Phi(s) = \frac{1}{s^l(s-p)^l} \tag{B.2}$$

where $p > 0$ and l is an integer. The poles of $\Phi(s)$ in the right-half plane are $s = p$ of order l. The residue r of an l^{th} order pole p of a function $f(s)$ is given by [45]

$$r = \lim_{s \to p} \frac{1}{(l-1)!} \frac{d^{l-1}}{ds^{l-1}} (s-p)^l f(s) \tag{B.3}$$

Applying this result, equation (B.1) may be written as

$$S_r = \lim_{s \to p} \frac{1}{(l-1)!} \frac{d^{l-1}}{ds^{l-1}} \frac{1}{s^{l+1}} \tag{B.4}$$

$$= \lim_{s \to p} \frac{(2l-1)!(-1)^l}{(l-1)!l!s^{2l}} \tag{B.5}$$

$$= \frac{(2l-1)!}{(l-1)!l!(-p^2)^l} \tag{B.6}$$

For $p = \frac{1}{2}$ we have

$$S_r = \frac{(-4)^l(2l-1)!}{(l-1)!l!} \tag{B.7}$$

Appendix C

Pole product calculation.

We have the equation for the characteristic function $\Phi_D(s)$ of the decision variable D.

$$\Phi_D(s) = \frac{1}{\det I + 2s\Phi'_{WW}F'} = \prod_{k\in\eta} \frac{p_k p_{k+L}}{(s-p_k)(s-p_{k+L})} \tag{C.1}$$

We want to determine the product of the poles of $\Phi_D(s)$. The poles p_i are related to the eigenvalues λ_i of $\Phi_{WW}F$ by

$$p_k = \frac{-1}{2\lambda_k} \tag{C.2}$$

From the characteristic polynomial of $\Phi'_{WW}F'$

$$\det(\Phi'_{WW}F' - xI) = \prod_{k\in\eta}(x-\lambda_k)(x-\lambda_{k+L}) \tag{C.3}$$

and setting x to zero gives

$$\det \Phi'_{WW}F' = \prod_{k\in\eta} \lambda_k\lambda_{k+L} \tag{C.4}$$

Combining equations (C.2) and (C.4) gives the product of the poles as

$$\prod_{k\in\eta} p_k p_{k+L} = \frac{1}{4^l \det(\Phi'_{WW}F')} \tag{C.5}$$

Appendix D

Group algebra revision.

Much of the theory of geometrically uniform construction is based on group algebra. We will revise the basic group algebra concepts [60] to define the notation used in chapters 4 and 5, in the context of our requirements.

D.1 Preliminaries.

Definition 35 *If S is a set, then a function*

$$* : S \times S \to S$$

is called a binary operation *on S.*

An element from $S \times S$ may be written as (a, b) where $a \in S$ and $b \in S$. The operation of the function $*$ on (a, b) is $*((a, b))$ which using infix notation is written as $a*b$, an element of S. For example the function of addition, conventionally written as $+$, is a binary operation on the set R of real numbers. Other commonly known examples of binary operators are multiplication on R, and matrix multiplication on $N_{2\times 2}$.

Definition 36 *An* algebraic system $(S, *)$ *is a set S together with a binary operation $*$ defined on it.*

For example, the real numbers R with the operation of addition is the algebraic system $(R, +)$.

Definition 37 *An algebraic system* $(S, *)$ *is* associative *if* $(x * y) * z = x * (y * z)$ *for all* x, y, *and* z *in* S.

Definition 38 *In an algebraic system* $(S, *)$ *an element* $e \in S$ *is an* identity element *if for any* s *in* S

$$s * e = e * s = s.$$

For example the number 0 is the identity element in the system $(R, +)$.

Definition 39 *If* $(S, *)$ *is an algebraic system* with identity e, *and* $s \in S$, *then an element* t *in* S *is said to be an* inverse for s *if*

$$s * t = t * s = e.$$

The inverse for s *is denoted by* s^{-1}, *i.e.* $t = s^{-1}$.

For example in $(R, +)$, the inverse of 2 is -2 since $2 + -2 = -2 + 2 = 0$, the identity.

D.2 Definition of a group.

Definition 40 *If* G *is a set and* $*$ *is a binary operation on* G, *then the system* $(G, *)$ *is a* group *if it satisfies all of the following properties:*

1. $(G, *)$ *is associative, i.e.* $x * (y * z) = (x * y) * z$ *for all* x, y *and* z *in* G.
2. G *has an identity element* e, *i.e. there is an element* $e \in G$ *with the property* $e * x = x * e = x$ *for all* x *in* G.
3. *Every element* x *in* G *has an inverse* x^{-1} *in* G, *i.e. for every element* x *in* G *there is an element* x^{-1} *in* G *with the property* $x * x^{-1} = x^{-1} * x = e$.

For example consider the system $(R, +)$. Addition is associative so property 1 holds. There exists an identity element, namely 0 which has the property $0 + x = x + 0 = x$, and finally for every x there is an inverse $-x$ such that $x + -x = -x + x = 0$. Therefore $(R, +)$ is a group.

Definition 41 *If* $(G, *)$ *is a group and* G *has an infinite number of elements, then* $(G, *)$ *is said to be a group of* infinite order. *If* G *has a finite number of elements, then the* order *of* G *is the number of elements in* G, *written as* $|G|$.

Definition 42 *A group* $(G, *)$ *is called an* abelian group *if*

$$x * y = y * x$$

for all $x, y \in G$.

That is the group G is *commutative.*

An example of a finite group important to our study is based on the symmetries of a 4-PSK constellation. Consider a 4-PSK constellation with the points labelled as shown in figure D.1. A symmetry is a mapping such that the constellation maps onto

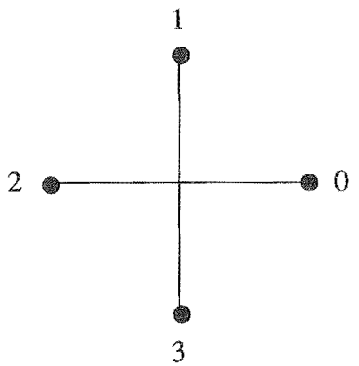


Figure D.1: A 4-PSK constellation.

itself. For example a 90 degree rotation about the origin, or a reflection about the horizontal axis are symmetries. By inspection there are a total of eight symmetries, namely four rotation symmetries and four reflection symmetries. The four rotation symmetries are shown in figure D.2 and are labelled r_0, r_1, r_2, and r_3 corresponding to anti-clockwise rotations of 0, 90, 180 and 270 degrees about the origin respectively. The four reflection symmetries are shown in figure D.3 and are labelled v_0, v_1, v_2 and v_3 corresponding to reflections about the line passing through the origin meeting the horizontal at 0, 45, 90, and 135 degrees respectively. Consider the set of symmetries $G = \{r_0, r_1, r_2, r_3, v_0, v_1, v_2, v_3\}$. A binary operation $\circ$ is defined on G by considering the result when one mapping of G is followed by another mapping G, as shown in the example of figure D.4. Here the original 4-PSK constellation is mapped through the symmetry v_0 followed by the symmetry r_1. The result is the same as the mapping of the symmetry v_3, hence we have $v_0 \circ r_1 = v_3$. The complete set of operations, listed in table D.1 is found by considering the results of the operation of all pairs of elements. The system $(G, \circ)$ is a group since the operation of any two elements of

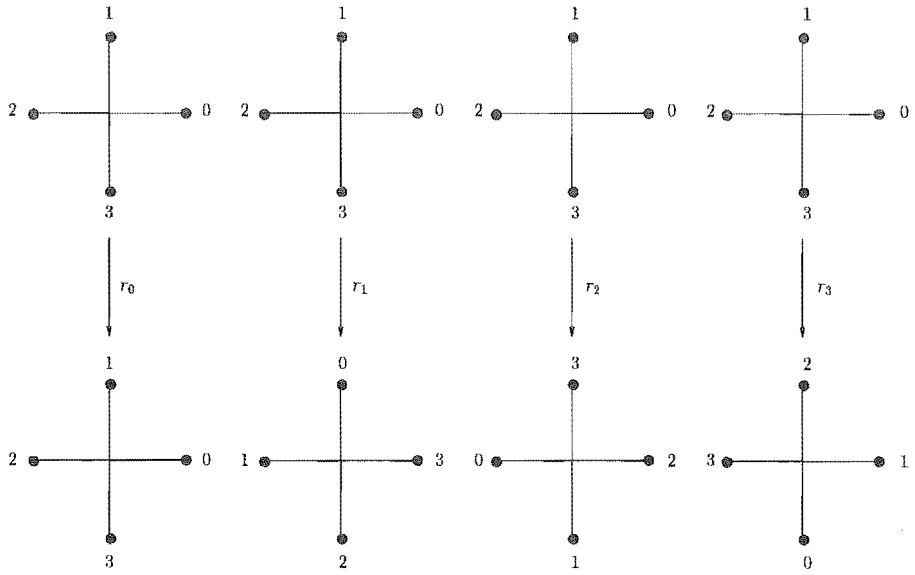


Figure D.2: The four rotation symmetries of a 4-PSK constellation.

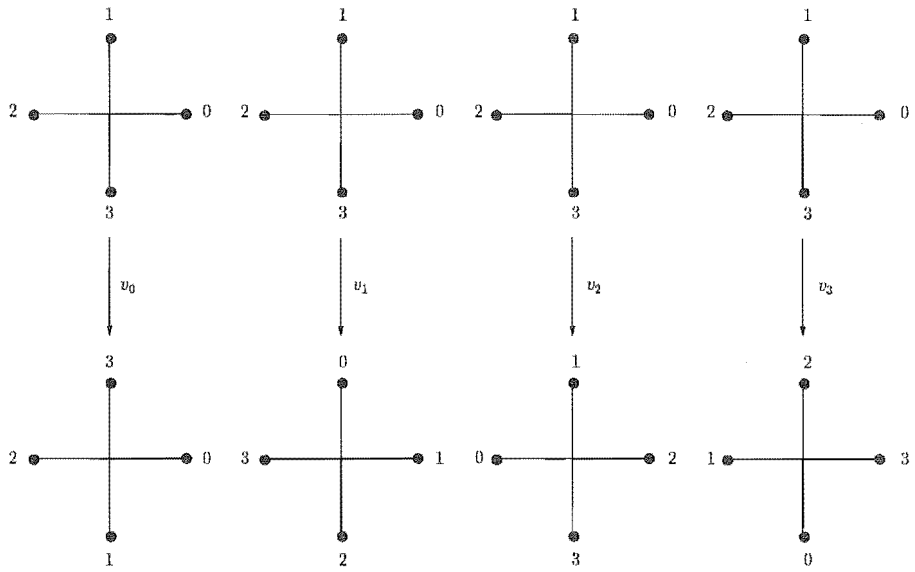


Figure D.3: The four reflection symmetries of a 4-PSK constellation.

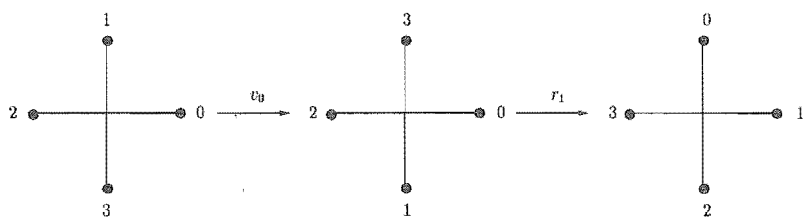


Figure D.4: Example of two operations.

$\circ$	r_0	r_1	r_2	r_3	v_0	v_1	v_2	v_3
r_0	r_0	r_1	r_2	r_3	v_0	v_1	v_2	v_3
r_1	r_1	r_2	r_3	r_0	v_3	v_0	v_1	v_2
r_2	r_2	r_3	r_0	r_1	v_2	v_3	v_0	v_1
r_3	r_3	r_0	r_1	r_2	v_1	v_2	v_3	v_0
v_0	v_0	v_1	v_2	v_3	r_0	r_1	r_2	r_3
v_1	v_1	v_2	v_3	v_0	r_3	r_0	r_1	r_2
v_2	v_2	v_3	v_0	v_1	r_2	r_3	r_0	r_1
v_3	v_3	v_0	v_1	v_2	r_1	r_2	r_3	r_0

Table D.1: Complete 4-PSK symmetry operation table.

G is in G, there exists an identity element, namely r_0, each element has an inverse, and the system is associative. The group $(G, \circ)$ is called the *dihedral* group D_4. In general, a dihedral group $(D_n, \circ)$ consists of all symmetries of a regular polygon with n sides. Note that D_4 is not an abelian group.

D.3 Subgroups.

Definition 43 *If $(G, \circ)$ is a group, and if $H \subseteq G$, then $(H, \circ)$ is called a* subgroup *of $(G, \circ)$ if $(H, \circ)$ is a group. If H is a subgroup of G, we write $H < G$.*

Basically, a subgroup of a group is a subset which forms a group with the *same* operation. For example the set of even integers with the operation + is a subgroup of the set of integers Z. By inspection the subset $\{r_0, r_1, r_2, r_3\}$ of the group D_4 of table D.1 is a group and therefore a subgroup of D_4.

D.4 Cosets.

Consider a group G and a subgroup H of G, $H < G$.

Definition 44 *For $g \in G$, the set $H \circ g = \{h \circ g | h \in H\}$ is called the* right coset of the subgroup H in G determined by g.

Definition 45 *For $g \in G$, the set $H \circ g = \{g \circ h | h \in H\}$ is called the* left coset of the subgroup H in G determined by g.

Note that every element g in G has a right coset $H \circ g$ and a left coset $g \circ H$. The right and left cosets need not equal if the group G is not abelian. It can be shown [60] that two right cosets (the same result holds for left cosets) $H \circ g$ and $H \circ g_1$ are either disjoint or equal. There can be no overlap between two right cosets of a subgroup in a group unless the cosets are the same. It can also be shown that G is the union of all right or left cosets. For example consider the group $G = D_4$, and the subgroup $H = \{r_0, v_1\}$. The right cosets are

$$H \circ r_0 = H \circ v_1 = \{r_0, v_1\} \tag{D.1}$$

$$H \circ r_1 = H \circ v_2 = \{r_1, v_2\} \tag{D.2}$$

$$H \circ r_2 = H \circ v_3 = \{r_2, v_3\} \tag{D.3}$$

$$H \circ r_3 = H \circ v_0 = \{r_3, v_0\} \tag{D.4}$$

while the left cosets are

$$r_0 \circ H = v_1 \circ H = \{r_0, v_1\} \tag{D.5}$$

$$r_1 \circ H = v_0 \circ H = \{r_1, v_0\} \tag{D.6}$$

$$r_2 \circ H = v_3 \circ H = \{r_2, v_3\} \tag{D.7}$$

$$r_3 \circ H = v_2 \circ H = \{r_3, v_2\} \tag{D.8}$$

Note that the union of the left or right cosets equals G and the left and right cosets do not necessarily equal.

D.5 Normal subgroups.

Definition 46 *If G is a group and N is a subgroup of G such that for every $g \in G$, $N \circ g = g \circ N$, we call N a* normal subgroup *of G and write $N \lhd G$.*

This definition states that the left and right cosets of the subgroup N determined by any element of G must be equal. In the previous example we showed that the left and right cosets of the subgroup $H = \{r_0, v_1\}$ do not equal and therefore H is not a normal subgroup of $G = D_4$. If instead we consider the subgroup $N = \{r_0, r_2\}$

of G its right cosets are

$$N \circ r_0 = N \circ r_2 = \{r_0, r_2\} \tag{D.9}$$

$$N \circ r_1 = N \circ r_3 = \{r_1, r_3\} \tag{D.10}$$

$$N \circ v_0 = N \circ v_2 = \{v_0, v_2\} \tag{D.11}$$

$$N \circ v_1 = N \circ v_3 = \{v_1, v_3\} \tag{D.12}$$

while the left cosets are

$$r_0 \circ N = r_2 \circ N = \{r_0, r_2\} \tag{D.13}$$

$$r_1 \circ N = r_3 \circ N = \{r_1, r_3\} \tag{D.14}$$

$$v_0 \circ N = v_2 \circ N = \{v_0, v_2\} \tag{D.15}$$

$$v_1 \circ N = v_3 \circ N = \{v_1, v_3\} \tag{D.16}$$

Clearly the left and right cosets are equal and N is a normal subgroup of G.

D.6 Quotient groups.

Definition 47 *If $N \lhd G$, then let $G|N$ = the set of all cosets of N in G.*

From the previous example $N = \{r_0, r_2\}$, a normal subgroup of G and $G|N$ is

$$G|N = \{\{r_0, r_2\}, \{r_1, r_3\}, \{v_0, v_2\}, \{v_1, v_3\}\} \tag{D.17}$$

the set of cosets. In general the order of $G|N$ is $|G|/|N|$.

Definition 48 *If G is a group and $N \lhd G$, then the system $(G|N, \circ)$ where $\circ$ is defined by*

$$(N \circ g_1) \circ (N \circ g_2) = N \circ (g_1 \circ g_2) \tag{D.18}$$

is a group, called the quotient group.

The proof is straightforward [60]. From the previous example we have constructed the operation table of the quotient group $G|N$, for $N = \{r_0, r_2\}$, as shown in table D.6.

$\circ$	$\{r_0, r_2\}$	$\{r_1, r_3\}$	$\{v_0, v_2\}$	$\{v_1, v_3\}$
$\{r_0, r_2\}$	$\{r_0, r_2\}$	$\{r_1, r_3\}$	$\{v_0, v_2\}$	$\{v_1, v_3\}$
$\{r_1, r_3\}$	$\{r_1, r_3\}$	$\{r_0, r_2\}$	$\{v_1, v_3\}$	$\{v_0, v_2\}$
$\{v_0, v_2\}$	$\{v_0, v_2\}$	$\{v_1, v_3\}$	$\{r_0, r_2\}$	$\{r_1, r_3\}$
$\{v_1, v_3\}$	$\{v_1, v_3\}$	$\{v_0, v_2\}$	$\{r_1, r_3\}$	$\{r_0, r_2\}$

D.7 Isomorphisms.

Definition 49 *Let G and H be any groups and let $\alpha : G \to H$ be a bijection (1-1 and onto mapping) between the elements of G and the elements of H such that for all x, y in G*

$$\alpha(x \circ y) = \alpha(x) \circ \alpha(y) \tag{D.19}$$

Then α is called an isomorphism *of G onto H and G and H are said to be* isomorphic.

Appendix E

Code generators.

This appendix lists the geometrically uniform code generators of chapter 4.

1 bit/symbol 1×4-PSK	
v	generators
1	v_3, r_2
2	r_2, v_3, r_2
3	r_2, v_3, r_2, r_2
4	r_2, v_1, v_3, v_3, r_2
5	$v_1, v_3, r_2, v_3, r_2, r_2$
6	$r_2, v_1, v_3, r_2, r_2, v_3, r_2$
7	$r_2, v_3, v_1, v_3, r_2, r_2, v_3, r_2$
8	$r_2, v_3, r_0, r_2, r_2, v_1, r_2, v_3, r_2$

Table E.1: Generators of 1×4-PSK codes transmitting 1 bit/symbol.

1 bit/symbol 1 × 8-PSK	
v	generators
1	r_4, v_3
2	r_4, v_3, r_4
3	v_3, r_4, v_5, r_4
4	r_4, v_5, v_3, r_4, v_3
5	$r_4, v_5, v_3, v_5, r_4, v_3$
6	$v_5, v_3, r_4, v_5, v_3, v_5, r_4$
7	$v_3, v_1, r_4, v_5, v_3, r_4, v_5, r_4$
8	$r_4, v_3, v_5, r_4, v_1, v_3, r_4, v_5, r_4$

Table E.2: Generators of 1 × 8-PSK codes transmitting 1 bit/symbol.

1 bit/symbol 2 × 4-PSK	
v	generators
1	$(r_2, r_2); (v_1, v_3), (v_1, r_2)$
2	$(r_2, v_3), (r_2, v_1); (v_1, v_1), (v_1, r_2)$
3	$(r_2, r_2), (r_2, v_1); (v_1, v_3), (v_1, r_2), (v_1, r_2)$
4	$(v_3, v_3), (r_2, v_3), (r_2, v_1); (v_1, r_2), (v_1, v_1), (v_1, r_2)$
5	$(r_2, r_2), (r_2, v_3), (r_2, v_1); (v_1, v_3), (v_1, r_2), (r_0, v_1), (v_1, v_3)$
6	$(v_1, r_2), (r_2, v_1), (r_2, v_3), (r_2, v_1); (v_3, v_1), (v_1, v_3), (v_1, r_2), (v_1, v_3)$
7	$(v_1, v_3), (r_2, r_2), (v_3, v_1), (r_2, r_2); (v_3, r_2), (r_0, r_2), (v_1, r_2), (r_0, r_2), (v_1, v_3)$
8	$(r_2, v_3), (v_1, r_0), (r_2, v_3), (r_2, r_2), (r_2, r_2); (v_1, r_2), (v_3, r_2), (v_1, r_2), (r_0, r_2), (v_1, v_1)$

Table E.3: Generators of 2 × 4-PSK codes transmitting 1 bit/symbol.

1 bit/symbol 3×4-PSK	
v	Generators
1	$(r_2, v_3, v_3); (v_1, v_1, r_2); (v_1, v_3, v_1), (v_1, v_3, r_0)$
2	$(r_2, r_2, r_2); (v_1, v_3, v_3), (v_1, r_2, r_2); (r_0, r_2, v_1), (r_0, r_2, v_3)$
3	$(r_2, r_2, r_0), (r_2, r_2, r_2); (v_1, v_3, r_2), (v_1, r_2, v_3); (v_1, r_2, v_1), (r_0, v_3, v_1)$
4	$(v_3, v_3, v_1), (r_2, r_2, v_3); (v_1, r_2, v_3), (v_1, v_3, r_2); (r_0, v_3, r_2), (v_1, v_3, r_0), (v_1, r_2, r_0)$
5	$(r_2, r_2, v_3), (r_2, r_2, r_2); (v_1, v_1, r_0), (v_1, v_3, r_2), (v_1, r_2, v_3); (v_1, v_3, v_1), (r_0, r_2, v_1), (r_0, v_3, v_3)$
6	$(r_2, r_2, v_3), (r_2, r_2, r_2), (r_2, r_2, r_2); (v_1, v_3, v_1), (v_1, v_3, v_3), (v_1, r_2, r_2); (r_0, r_2, r_2), (v_1, r_2, v_1), (r_0, v_3, v_1)$
7	$(r_2, v_3, v_3), (r_2, r_2, r_2), (r_2, r_2, r_2); (v_1, v_1, v_1), (v_1, v_3, v_3), (v_1, r_2, v_3); (v_1, v_3, r_2), (v_1, v_3, v_1), (r_0, v_1, v_1), (r_0, v_3, r_2)$
8	$(v_3, v_3, v_3), (r_2, r_2, r_2), (r_2, r_2, r_2); (v_1, r_2, r_2), (r_0, r_0, r_0), (v_1, r_2, r_2), (v_1, r_2, r_2); (r_0, v_1, r_2), (v_1, r_2, v_3), (r_0, r_2, v_1), (r_0, v_1, v_3)$

Table E.4: Generators of 3×4-PSK codes transmitting 1 bit/symbol.

2 bits/symbol 1×8-PSK	
v	generators
1	$r_4; r_4, v_3$
2	$v_5, v_7; v_1, r_4$
3	$r_4, v_3; v_3, v_1, r_4$
4	$v_1, r_4, r_4; v_5, v_3, v_1$
5	$v_5, v_3, v_3; v_1, r_0, r_4, r_4$
6	$r_4, r_4, v_7, v_3; v_3, v_1, r_4, v_5$
7	$r_4, r_4, v_1, r_4; v_3, v_1, v_3, v_3, v_1$
8	$v_5, v_3, v_1, v_1, r_4; v_1, v_7, v_5, v_5, v_3$

Table E.5: Generators of 1×8-PSK codes transmitting 2 bits/symbol.

2 bits/symbol 1 × 16-PSK	
v	generators
1	$r_8; v_7, r_4$
2	$v_{13}, r_8; v_5, v_3$
3	$v_9, r_8; v_1, v_5, v_3$
4	$r_8, v_3, v_5; v_5, v_{11}, v_{13}$
5	$r_8, v_9, r_8; v_3, v_5, v_5, v_7$
6	$r_8, v_3, v_{13}, v_{11}; v_5, v_{11}, v_5, v_3$
7	$r_8, v_9, v_{11}, r_8; v_3, v_5, v_3, v_7, v_5$
8	$v_{13}, v_{11}, r_8, v_{13}, v_{11}; v_5, v_3, v_7, v_5, v_3$

Table E.6: Generators of 1 × 16-PSK codes transmitting 2 bits/symbol.

2 bits/symbol 2 × 8-PSK	
v	generators
1	$(r_4, r_4); (r_6, r_6); (r_1, r_5); (r_6, r_2), (v_2, v_1)$
2	$(r_4, r_4); (r_6, r_2); (r_0, r_4), (v_1, v_7); (v_1, v_5), (r_4, r_0)$
3	$(r_4, r_4); (v_3, v_1), (v_3, r_4); (v_1, v_7), (r_0, r_4); (r_4, r_0), (v_1, v_5)$
4	$(r_4, r_4), (v_7, v_7); (v_3, v_5), (r_4, r_4); (v_1, v_3), (r_0, r_4); (r_4, r_0), (v_1, v_5)$
5	$(r_4, r_4), (v_7, v_3); (v_3, v_3), (r_4, r_4); (v_1, v_1), (v_1, v_5); (r_4, r_0), (v_1, v_7), (r_4, r_0)$
6	$(r_4, r_4), (v_2, v_7); (v_3, v_6), (r_4, r_4); (v_1, v_4), (r_0, r_4), (v_4, v_1); (v_2, v_1), (v_4, v_1), (v_1, v_6)$
7	$(r_4, v_5), (r_4, r_4); (r_0, r_4), (r_0, r_4), (v_3, v_3); (v_1, r_4), (v_3, r_0), (v_1, v_5); (v_1, v_3), (v_1, v_7), (r_4, r_0)$
8	$(r_4, r_4), (r_4, r_0), (v_7, v_7); (v_3, v_3), (v_3, v_6), (r_4, r_0); (r_0, r_4), (v_1, v_2), (r_4, r_4); (v_1, v_1), (v_1, v_6), (v_1, v_1)$

Table E.7: 2 × 8-PSK codes transmitting 2 bit/symbol.

2 bits/symbol 2×16-PSK	
v	generators
1	$(r_8, r_8); (r_{12}, r_{12}); (r_6, r_{14}); (v_{13}, v_{15}), (v_{13}, v_7)$
2	$(r_8, r_8); (r_4, r_4); (r_2, r_2), (r_2, r_{10}); (v_1, v_{13}), (v_1, v_{15})$
3	$(r_8, r_8); (r_{12}, r_4), (v_{13}, v_{15}); (r_0, r_8), (r_{12}, r_4); (v_1, v_7), (r_0, r_8)$
4	$(v_9, v_5), (v_15, v_{13}); (v_1, v_{13}), (v_7, v_5); (v_5, v_9), (v_3, v_1); (v_3, v_3), (v_1, v_7)$
5	$(v_{15}, v_9), (r_8, r_8); (v_7, v_1), (v_5, v_{13}); (v_3, v_5), (v_1, v_9); (v_1, v_{15}), (v_3, v_{11}), (v_3, v_{11})$
6	$(v_{13}, v_{13}), (r_8, r_8); (v_5, r_8), (v_5, v_3); (r_0, v_5), (v_1, v_7), (v_1, v_7); (v_1, r_8), (v_3, v_{13}), (v_3, v_5)$
7	$(r_8, v_9), (r_8, r_8); (r_0, r_8), (v_3, r_8), (v_7, v_{15}); (v_5, v_{13}), (r_0, r_8), (v_3, v_3); (r_0, v_5), (v_7, v_5), (v_1, v_{13})$
8	$(r_8, v_{15}), (r_8, r_8), (r_8, r_8); (v_5, r_8), (v_7, v_3), (v_7, v_{13}); (r_0, r_8), (v_3, v_7), (v_3, v_9); (v_5, v_{11}), (v_1, v_9), (v_1, v_7)$

Table E.8: 2×16-PSK codes transmitting 2 bits/symbol.

References

[1] J.B. Anderson. "Limited search trellis decoding of convolutional codes". *IEEE Trans. Inform. Theory,* vol. IT-35, no. 5, pp. 944-955, Sep. 1989.

[2] S. Benedetto, R. Garello, M. Mondin, and G. Montorsi. "Geometrically uniform TCM codes over groups based on L×MPSK constellations". *IEEE Trans. Inform. Theory,* vol. IT-40, no. 1, pp. 137-152, Jan. 1994.

[3] S. Benedetto, R. Garello, M. Mondin, and G. Montorsi. "Geometrically uniform partitions of L×MPSK constellations and related binary trellis codes". *IEEE Trans. Inform. Theory,* vol. IT-39, no. 6, pp. 1773-1798, Nov. 1993.

[4] M. Bertelsmeier and G. Komp. "Trellis-coded 8-PSK with embedded QPSK". *IEEE J. Selec. Areas Commun.* vol. 7, no. 9, pp. 1296-1306, Dec. 1989.

[5] E. Biglieri and G. Caire. "Symmetry properties of multilevel coded modulation". *IEEE Trans. Inform. Theory,* vol. IT-40, no. 5, pp. 1630-1632, Sep. 1994.

[6] E. Biglieri, D. Divsalar, P.J. McLane, and M.K. Simon. *"Introduction to Trellis-Coded Modulation with Applications".* New York: Macmillan Publishing Company, 1991.

[7] R.E. Blahut. *"Theory and practice of error control codes".* Addison-Wesley, 1983.

[8] A. Brajal and A. Chouly. "Optimal trellis-codes 8-PSK and 4-AM modulations for the Rayleigh channel". *Supercomm/ICC,* vol. 1, pp. 28-33, 1994.

[9] G. Butler. *"Fundamental Algorithms for Permutation Groups".* Springer-Verlag, New York, 1991.

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

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

[12] A.R. Calderbank and N.J.A. Sloane. "New trellis codes based on lattices and cosets". *IEEE Trans. Inform. Theory,* vol. IT-33, no. 2, pp. 177-195, Mar. 1987.

[13] J.K. Cavers and P. Ho. "Analysis of the error performance of trellis-coded modulations in Rayleigh-fading channels". *IEEE Trans. Commun.*, vol. 40, no. 1, pp. 74-83, Jan. 1992.

[14] M. Cedervall and R. Johannesson. "A fast algorithm for computing distance spectrum of convolutional codes". *IEEE Trans. Inform. Theory,* vol. IT-35, no. 6, pp. 1146-1159, Nov. 1989.

[15] K.Y. Chan and A. Bateman. "The performance of reference-based M-ary PSK with trellis coded moduation in Rayleigh fading". *IEEE Trans. Veh. Technol.*, vol. 41, no. 2, pp. 190-198, May, 1992.

[16] D. Chase. "A class of algorithms for decoding block codes with channel measurement information". *IEEE Trans. Inform. Theory,* vol. IT-18, no. 1, pp. 170-182, Jan. 1972.

[17] J.H. Conway and N.J.A. Sloane. *"Sphere packings, lattices, and groups".* Springer-Verlag, NY, 1988.

[18] R.C. Dingman and D.P. Taylor. "A new upper bound on trellis code error performance on Rayleigh flat-fading channels". Master's thesis, McMaster University, 1992.

[19] D. Divsalar and M.K. Simon. "Trellis coded modulation for 4800-9600 bits/s transmission over a fading mobile satellite channel". *IEEE J. Select. Areas Commun.*, vol. 5, no. 2. pp. 162-175, Feb. 1987.

[20] D. Divsalar and M.K. Simon. "Maximum-likelihood differential detection of uncoded and trellis coded amplitude phase modulation over AWGN and fading

channels - metrics and performance". *IEEE Trans. Commun.*, vol. 42, no. 1, pp. 76-89, Jan. 1994.

[21] D. Divsalar and M.K. Simon. "Multiple Symbol Differential Detection of MPSK". *IEEE Trans. Commun.*, vol. 38, pp. 300-308, Mar. 1990.

[22] D. Divsalar and M.K. Simon. "The design of trellis coded MPSK for fading channels: Performance Criteria". *IEEE Trans. Commun.*, vol. COM-36, no. 9, pp. 1004-1012, Sep. 1988.

[23] D. Divsalar and M.K. Simon. "The design of trellis coded MPSK modulation for fading channels: Set partitioning for optimum code design". *IEEE Trans. Commun*, vol. COM-36, no. 9, pp. 1013-1021, Sep. 1988.

[24] D. Divsalar, M.K. Simon, and M. Shahshahani. "The performance of trellis-code MDPSK with multiple symbol detection". *IEEE Trans. Commun.*, vol. 38, no. 9, pp. 1391-1403, Sep. 1990.

[25] F. Edbauer. "Performance of interleaved trellis-coded differential 8-PSK modulation over fading channels". *IEEE J. Select. Areas Commun.*, vol. 7, no. 9, pp. 1340-1346, Dec. 1989.

[26] M.V. Eyuboglu and G.D. Forney Jr. "Trellis precoding: Combined coding, precoding and shaping for intersymbol interference channels". *IEEE Trans. Inform. Theory*, vol. IT-38, no. 2, pp. 301-314, Mar. 1992.

[27] S. A. Fechtel and H. Meyr. "matched filter bound for trellis-coded transmission over frequency-selective fading channels with diversity". *European Trans. Telecomm.*, vol. 4, no. 3, pp. 343-354, May-Jun. 1993.

[28] T.R. Fischer, M.W. Marcelin, and M. Wang. "Trellis codes vector quantization". *IEEE Trans. Inform. Theory*, vol. IT-37, no. 6, pp. 1551-1566, Nov. 1991.

[29] A. Gersho and V.B. Lawrence. "Multidimensional Signal Constellations for Voiceband Data Transmission". *IEEE J. Select. Areas Commun.*, vol. 2, no. 5. pp. 687-702, Sep. 1984.

[30] R.W. Hamming. "Error detecting and error correcting codes". *Bell Syst. Tech. J.,* vol. 42, pp. 79-94, Apr. 1950.

[31] S. Haykin. *"Digital Communications"*. Wiley, 1988.

[32] P. Ho and D. Fung. "Error Performace of Multiple Symbol Differential Detection of PSK Signals Transmitted over Correlated Rayleigh Fading Channels". *IEEE Trans. Commun.,* vol.. 40, no. 10, pp. 1566-1569, Oct. 1992.

[33] J. Huang. "Performance analysis of multiple trellis coded MDPSK in an AWGN channel". *IEEE Pacific Rim conference on commun., comp., and sig. proc. pp. 83-86,* May 1991.

[34] D.A. Huffman. "A method for the contruction of minimum-redundancy codes". *Proc. IRE,* vol. 40, pp. 1098-1101, Sep. 1952.

[35] H. Imai and S. Hirakawa. "A new multilevel coding method using error-correcting codes". *IEEE Trans. Inform. Theory,* vol. IT-23, no. 3, pp. 371-377, May 1977.

[36] G.D. Forney Jr. "Convolutional codes II: Maximum-likelihood decoding". *Inform. Contr.,* vol. 25, pp. 222-266, 1974.

[37] G.D. Forney Jr. "Convolutional codes III: Sequential decoding". *Inform. Contr.,* vol. 25, pp. 267-297, 1974.

[38] G.D. Forney Jr. "The Viterbi algorithm". *Proc. of the IEEE,* vol. 61, pp. 268-278, Mar. 1973.

[39] G.D. Forney Jr. "Trellis shaping". *IEEE Trans. Inform. Theory,* vol. IT-38, no. 2, pp. 281-300, Mar. 1992.

[40] G.D. Forney Jr. "Convolutional codes I: algebraic structure". *IEEE Trans. Inform. Theory,* vol. IT-16, no. 6, pp. 720-738, Nov. 1970.

[41] G.D. Forney Jr. "Geometrically Uniform Codes". *IEEE Trans. Inform. Theory,* vol. IT-37, no. 5, pp. 1241-1260, Sep. 1991.

[42] G.D. Forney Jr. and M.D. Trott. "The dynamics of group codes: State spaces, trellis diagrams and canonical encoders". *IEEE Trans. Inform. Theory,* 1993.

[43] W.C. Jakes Jr. *"Microwave Mobile Communications"*. New York: Wiley, 1974.

[44] T. Kasami, T. Takata, T. Fujiwara, and S. Lin. "on multilevel block modulation codes". *IEEE Trans. Inform. Theory,* vol. IT-37, no. 4, pp. 965-975, Jul.. 1991.

[45] E. Kreyzig. *"Advanced Engineering Mathematics"*. New York: Wiley, 6th edition, 1988.

[46] W.C.Y. Lee. "Estimate of channel capacity in Rayleigh fading environment". *IEEE Trans. on Veh. Technol.,* vol. 39, no. 3, pp. 187-189, Aug. 1990.

[47] S. Lin and D.J. Costello. *"Error Control Coding"*. Englewood Cliffs, NJ: Prentice-Hall, 1983.

[48] F.J. MacWilliams and N.J.A. Sloane. *"The Theory of Error-Correcting Codes, Part I"*. North-Holland, 1977.

[49] M.G. Mulligan and S.G. Wilson. "An improved algorithm for evaluating trellis phase codes". *IEEE Trans. Inform. Theory,* vol. IT-30, no. 6, pp. 846-851, Nov. 1984.

[50] B. Noble and J.W. Daniel. *"Applied Linear Algebra"*. Prentice-Hall, 3rd edition, 1988.

[51] A. Papoulis. *"Probability, Random Variables and Stochastic Processes"*. McGraw-Hill, 1965.

[52] S.S. Periyalwar and S.M. Fleisher. "A modified design of trellis-coded MPSK for the fading channel". *IEEE Trans. Commun.,* vol. 41, no. 6, pp. 874-881, Jun. 1993.

[53] S.S. Pietrobon. *"Trellis coding with multi-dimensional signal sets and rotationally invariant trellis codes"*. PhD thesis, University of Notre Dame, Dec, 1990.

[54] S.S. Pietrobon, R.H. Deng, A. LaFanechere, G. Ungerboeck, and D.J. Costello. "Trellis-coded multidimensional phase modulation". *IEEE Trans. Inform. Theory,* vol. IT-36, no. 1, pp. 63-89, Jan. 1990.

[55] J.E. Porath. "Converting convolutional codes from feedback to feedforward form". Technical report, School of Electrical and Computer Engineering, Chalmers University of Technology, 1987.

[56] G.J. Pottie and D.P. Taylor. "A comparison of reduced complexity decoding algorithms for trellis codes". *IEEE J. Select. Areas Commun.*, vol. 7, no. 9, pp. 1369-1380, Dec. 1989.

[57] G.J. Pottie and D.P. Taylor. "Multilevel codes based on partitioning". *IEEE Trans. Inform. Theory,* vol. IT-35, no. 1, pp. 87-98, Jan. 1990.

[58] E.J. Rossin, N.T. Sindhushayana, and C.D. Heegard. "Trellis group codes for the Gaussian channel". *IEEE Trans. Inform. Theory,* vol. IT-41, no. 5, pp. 1217-1245, Sep. 1995.

[59] M. Rouanne and D.J. Costello. "A lower bound on the minimum Euclidean distance of trellis-coded modulation". *IEEE Trans. Inform. Theory,* vol. IT-34, no. 2, pp. 223-236, Mar. 1988.

[60] R. Sandler and L. S. Foster. *"Modern Algebra"*. Harper and Row, 1978.

[61] S.L. Sayegh. "A class of optimum block codes in signal space". *IEEE Trans. Commun.,* vol. 34, pp. 1043-1045, Oct. 1986.

[62] C. Schlegel and D.J. Costello Jr. "Bandwidth efficient coding for fading channels: Code construction and performance analysis". *IEEE J. Selec. Areas Commun.,* vol. 7, no. 9, pp. 1356-1368, Dec. 1989.

[63] M. Schwartz, W.R. Bennet, and S. Stein. *"Communication systems and techniques"*. McGraw-Hill, 1966.

[64] N. Seshadri and C-E. W. Sundberg. "Multi-level coded modulations for the Rayleigh fading channels". *Conf. Rec. Globecom 91,* pp. 47-51, Phoenix, Arizona, Dec. 1991.

[65] N. Seshadri and C.W. Sundberg. "Multilevel trellis coded modulations for the Rayleigh fading channel". *IEEE Trans. Commun.,* vol. 41, no. 9, pp. 1300-1310, Sep. 1993.

[66] C.E. Shannon. "A mathematical theory of communication". *Bell Syst. Tech. J.*, vol. 27, pp. 379-423 and pp. 623-656, 1948.

[67] M.K. Simon and D. Divsalar. "Multiple symbol partially coherent detection of MPSK". *IEEE Trans. Commun.*, vol. 42, no. 2/3/4, pp. 430-439, Feb./Mar./Apr. 1994.

[68] M.K. Simon and D. Divsalar. "The performance of trellis codes multilevel DPSK on a fading mobile satellite channel". *IEEE Trans. Commun.*, vol. 42, no. 2/3/4, pp. 430-439, Feb./Mar./Apr. 1994.

[69] D. Slepian. "Group codes for the Gaussian channel". *Bell Syst. Tech. J.*, vol. 47, pp. 575-602, Apr. 1968.

[70] S.B. Slimane and T. Le-Ngoc. "A tight upper bound on the error probability of coded modulation schemes in Rayleigh fading channels". *Conf. rec. PIMRC*, pp. 249-253, Sep. 1993.

[71] T. Takata, S. Ujita, and S. Lin. "Multistage decoding of multilevel block M-PSK modulation codes and its performance analysis". *IEEE Trans. Inform. Theory*, vol. IT-39, no. 4, pp. 1204-1218, Jul. 1993.

[72] R.M. Tanner. "Algebraic contruction of large euclidean distance combined coding/demodulation systems". Univ. Calif., Santa Cruz, Tech. Rep. UCSC-CRL-87-7, Jun. 1987.

[73] G. Ungerboeck. "Trellis-coded modulation with redundant signal sets part I: Introduction". *IEEE Commun. Mag.*, vol. 25, pp. 5-11, Feb. 1987.

[74] G. Ungerboeck. "Trellis-coded modulation with redundant signal sets, part II: State of the art". *IEEE Commun. Mag.*, vol. 25, no. 2, Feb. 1987.

[75] G. Ungerboeck. "Channel coding with multilevel/phase signals". *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 55-67, Jan. 1982.

[76] B. Vucetic. "Bandwidth efficient concatenated coding schemes for fading channels". *IEEE Trans. Commun.*, vol. 41, no. 1, pp. 50-61, Jan. 1993.

[77] L.F. Wei. "Trellis-coded modulation with multi-dimensional constellations". *IEEE Trans. Inform. Theory,* vol. IT-33, no. 4, pp. 483-501, Jul. 1987.

[78] L.F. Wei. "Coded M-DPSK with built-in time diversity for fading channels". *IEEE Trans. Inform. Theory,* vol. IT-39, no. 6, pp. 1820-1839, Nov. 1993.

[79] J. Wu and D.J. Costello Jr. "New multi-level codes over GF(q)". *IEEE Trans. Inform. Theory,* vol. IT-38, no. 3, pp. 933-939, May 1992.

[80] J. Wu and S. Lin. "Multilevel trellis MPSK modulation codes for the Rayleigh fading channel". *IEEE Trans. Commun.,* vol. 41, no. 9, pp. 1311-1318, Sep. 1993.

[81] K. Yu and P. Ho. "An improved differential detector for coded PSK modulations in Rayleigh fast-fading channels". *Singapore ICCS/ISITA pp. 1146-1150,* 1992.

[82] E. Zehavi. "8-psk trellis codes for a Rayleigh channel". *IEEE Trans. Commun.,* vol. 40, no. 5, pp. 873-884, May 1992.

[83] E. Zehavi and J.K. Wolf. "On the performance evaluation of trellis codes". *IEEE Trans. Inform. Theory,* vol. IT-33, no. 2, pp. 196-202, Mar. 1987.