Module–Phase–Codes with Non–Coherent Detection and Reduced–Complexity Decoding

by

Raymond Knopp

B. Eng.

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of Master of Engineering

Department of Electrical Engineering
McGill University
Montréal, Canada

September, 1993

Module–Phase Codes with Non–Coherent Detection

Abstract

This thesis considers M–ary phase coding for the non–coherent AWGN channel. More precisely, we develop block–coded MPSK modulation schemes specifically for non–coherent block detection which significantly surpass the performance of ideal uncoded *coherent* MPSK. A class of block codes which are well–matched to MPSK modulation, called *module–phase codes*, is presented. The algebraic framework used for defining these codes relies on elements of module theory which are discussed along with a method for constructing such codes for non–coherent detection. It is shown that differential encoding, when considered on a block basis, may be viewed as a specific code from a particular class of module–phase codes. Two classes of more powerful codes which achieve significant coding gain with respect to coherent detection of uncoded MPSK are presented. In the first class of module–phase codes, the coding gain is achieved at the expense of bandwidth expansion. In the second class, however, the coding gain is achieved at the expense of signal constellation expansion without expanding bandwidth. A reduced–complexity/sub–optimal decoding strategy based on a modification of *information set decoding* is described. Its performance is analysed through the use of computer simulations for various different codes. Finally, we address the performance of these codes combined with the reduced–complexity decoding method over correlated Rayleigh fading channels.

Sommaire

Cette thèse porte sur le codage en phase pour le canal non–cohérent à bruit blanc gaussien additif. Plus précisément, on développe des stratégies de modulation MPSK codées en bloc, conçues expressément pour la détection non–cohérente de bloc, qui dépassent considérablement la performance de MPSK non–codée avec la détection cohérente idéale. Une catégorie de codes bloc, nommée *codes module–phase*, qui va de paire avec la modulation MPSK, est introduite. La structure algébrique utilisée pour décrire ces codes s'appuie sur des éléments de la théorie des modules, qui seront expliqués de même qu'une méthode de construction dans le but de la détection non–cohérente. Il est ainsi démontré que le codage différentiel, considéré bloc par bloc, pourrait être vu comme un exemple particulier d'un groupe spécial de codes module–phase. Deux groupes de codes plus performants qui atteignent des gains de codage considérables comparés à la détection cohérente de MPSK non–codée sont présentés. Le gain de codage des codes du premier groupe repose sur l'agrandissement de la bande passante; ceux du deuxième groupe, cependant, réalisent leurs gains de codage par l'agrandissement de l'ordre de la modulation. Une stratégie de décodage sous–optimale à complexité réduite basée sur une modification de *information set decoding* est présentée. Son analyse pour différents codes est accomplie à l'aide de simulations par ordinateur. Finalement, on considère la performance de ces codes avec la stratégie de décodage à complexité réduite sur des canaux à évanouissement corrélé rayleigh.

Acknowledgements

First of all I must acknowledge the careful guidance and support I received from my thesis supervisor, Dr. Harry Leib, without which the completion of this work would have proven to be an impossible task. I would also like to acknowledge the financial support provided by the National Science and Engineering Research Council of Canada (NSERC) for my Master's studies.

I would like to thank my friends in the TSP lab who were always there to lend a helping hand and who had to put up with my sometimes excessive use of the lab's computers. Finally, I must thank my father who was extremely supportive throughout my studies and without whom I would never have been able to reach this point.

Contents

1 Introduction 1

2 Non–Coherent Detection of Phase–Modulated Signals 4

2.1 Phase Shift Keying (PSK) 4

2.2 Non–Coherent Block Detection of MPSK 7

2.2.1 Maximum Likelihood Detector 7

2.2.2 Performance of the Maximum Likelihood Detector 10

2.3 Non–Coherent Detection of Coded MPSK 12

2.3.1 Multiple–Symbol Differential Detection of MPSK 12

2.3.2 Reduced–complexity receiver structures 15

2.3.3 Error–Control Coding 18

2.3.4 Error–Control in Non–Coherent Systems 21

3 A Class of Codes for Non–Coherent Detection of MPSK 24

3.1 The ring $\mathcal{Z}_M$ and MPSK 24

3.2 Module–Phase Codes 26

3.3 A Method for Building Codes 29

3.3.1 Code Design by Exclusion of Unwanted Vectors 30

3.3.2 Selection of the first vector, $\mathbf{h}_0$ 34

3.3.3 Code Rate Improvement and Differential Encoding 36

3.3.4 Design of $\mathbf{h}_1$ to Yield Coding Gain 39

3.4 Results from code searches 41

3.5 Coding Without Bandwidth Expansion 48

4 Efficient Decoding of Module–Phase Codes with Non–Coherent Detection 54

4.1 A General Two–Stage Decoding Strategy 55

4.2 Two–Stage Non–Coherent Decoding of Module–Phase Codes 56

4.2.1 Generalized Differential Detection 57

4.2.2 Information Set Decoding 59

4.2.3 Modifying the basic scheme for non–coherent detection 61

4.3 Searching for codes better suited to Information Set Decoding 63

4.4 Decoding of Various Module–Phase–Codes 64

4.4.1 Computer Simulations 65

4.4.2 Bandwidth Expanding Codes in $\mathcal{Z}_4$ 66

4.4.3 Bandwidth Expanding Codes in $\mathcal{Z}_8$ 69

4.4.4 Bandwidth Efficient Codes 72

4.5 Performance over Correlated Rayleigh Fading Channels 75

4.5.1 The Correlated Rayleigh Fading Model 76

4.5.2 Error performance 77

5 Conclusion **85**

A Computer Searches and Code Descriptions **87**

A.1 Computer Searches 87

A.1.1 Searching for **h**–vectors 87

A.1.2 Searching for $\mathbf{G}_c$ 88

A.2 Code Descriptions for Bandwidth–Expanding Codes 88

A.3 Code Descriptions for Bandwidth–Efficient Codes 99

List of Tables

2.1 SNR degradation in dB of optimal non–coherent block detection of MPSK (from [3]) 15

2.2 Performance comparison of reduced complexity strategies 19

2.3 Extended basis set for generating an orthogonal code of length N=8 . . 22

2.4 Extended basis set for generating an orthogonal code of length N=8 . . 22

2.5 Asymptotic Performance of Rhodes' scheme for different values of N . . 23

3.1 Worst Vectors after applying $\mathbf{h}_0$ 41

3.2 Bandwidth Expanding codes in $\mathcal{Z}_4$ 44

3.3 Bandwidth Expanding codes in $\mathcal{Z}_8$ 45

3.4 Bandwidth efficient codes 49

4.1 Information sets and generator matrices for decoding a (7,4) code in $\mathcal{Z}_4$ 68

4.2 Information sets and generator matrices for decoding a (7,5) code 71

4.3 Information sets and generator matrices for decoding a (8,4) code in $\mathcal{Z}_4$ 73

4.4 Information sets for decoding a (14,7) code in $\mathcal{Z}_4$ 75

List of Figures

2.1 MPSK Signal Constellations for three values of M 5

2.2 Basic Receiver Structure 9

2.3 Complete Receiver Structure 14

2.4 Samejima's L stage detector 17

3.1 Possible correlation values for an uncoded system in $\mathcal{Z}_8^6$ 31

3.2 Possible correlation values for a coded system in $\mathcal{Z}_8^6$ 32

3.3 Worst Case and next to worst case correlation values 35

3.4 Advantage of codeword overlapping 39

3.5 Gray coding for $\mathcal{Z}_4$ and $\mathcal{Z}_8$ 43

3.6 Performance of some simple QPSK codes 46

3.7 Performance of some more powerful QPSK codes 46

3.8 Performance of some simple 8-PSK codes 47

3.9 Performance of some more powerful 8-PSK codes 47

3.10 Performance of some simple BPSK equivalent codes 51

3.11 Performance of some more powerful BPSK equivalent codes 52

3.12 Performance of some BPSK equivalent codes in higher order rings 52

3.13 Performance of some QPSK equivalent codes . . . . . 53.

4.1 A (6,5) Code in $\mathcal{Z}_4$ 67

4.2 A (7,4) Code in $\mathcal{Z}_4$ 69

4.3 A (5,4) Code in $\mathcal{Z}_8$ 70

4.4 A (7,5) Code in $\mathcal{Z}_8$ 72

4.5 An (8,4) Code in $\mathcal{Z}_4$ 74

4.6 A (14,7) Code in $\mathcal{Z}_4$ 76

4.7 Power spectrum of the land–mobile fading model 78

4.8 Union bound on the performance of an (8,4) BPSK equivalent code in $\mathcal{Z}_4$ for fade rates of $f_DT = .1, .01$ and $.001$ 79

4.9 Comparison of ideal and simulation autocorrelation functions 80

4.10 Magnitude and phase of the fading process over many symbols ($f_DT = .10$) 81

4.11 Magnitude and phase of the fading process over few symbols ($f_DT = .10$) 82

4.12 Simulation results for an (8,4) code in $\mathcal{Z}_4$ ($f_DT = .01$) 83

4.13 Simulation results for a (14,7) code in $\mathcal{Z}_4$ ($f_DT = .01$) 84

4.14 Simulation results for an (8,4) code in $\mathcal{Z}_4$ ($f_DT = .1$) 84

Chapter 1

Introduction

Recently, there has been increased interest in non–coherent detection schemes with improved performance over differentially–coherent systems [1],[2], [3]. The merit of these detection techniques is that they do not require carrier phase tracking, while exhibiting only a very small SNR degradation with respect to coherent detection. It seems that these robust detection schemes could be very attractive for wireless communications over channels where carrier phase tracking is very difficult to achieve. In this thesis we consider the problem of channel coding for *M–ary phase shift keying (MPSK)* with non–coherent block detection, the goal being to design codes which achieve significant performance improvements over uncoded coherent detection of MPSK. Integrating error control with modulation and coherent detection has been considered extensively in the last 15 years, [6], [7]. The corresponding problem with non–coherent detection, however, has received little attention. Here we will specifically address the problem of block–coded modulation primarily for non–coherent AWGN(additive white gaussian noise) channels.

Chapter 2 begins with the definition of MPSK with various detection techniques followed by a detailed review of non–coherent block detection, which is of primary interest in this thesis. The form of the maximum–likelihood (ML) decoder as well as its

performance are given. In addition, a distance measure suitable for non–coherent block detection is defined, as it will be the main criterion used in the design and comparison of codes in the subsequent chapters. We then present a comparison of reduced–complexity block receiver structures for differentially–encoded MPSK, followed by a survey of previous work on error–control coding combined with non–coherent detection.

Chapters 3 and 4 constitute the main contribution of this work. In Chapter 3, we present an algebraic framework for a class of linear block codes, called *module–phase codes*, which are well–matched to MPSK modulation. These codes are very similar to traditional linear block codes, except that they are defined over rings rather than fields. A systematic technique for building these codes for non–coherent block detection is then introduced. It is shown that traditional differential–encoding may be cast into this algebraic framework as an example of a simple code which can significantly improve performance over a differentially–coherent system, when the detection is performed on a block basis. Results from computer searches for more powerful codes which achieve significant coding gain over uncoded coherent MPSK systems are presented. Two types of codes are considered:

1. Codes which expand bandwidth but do not expand the signal constellation
2. Codes which expand the signal constellation but do not expand bandwidth

Using union–bounding techniques, several examples of bit error–rate performance curves are given, so as to show the performance of these codes at lower signal–to–noise ratios.

The design of reduced–complexity/sub–optimal decoding strategies is addressed in Chapter 4. The proposed method is a modification of *information set decoding* which was first introduced by Prange for decoding cyclic binary block codes in [8]. Additionally, it makes use of ideas very similar to those used by Wilson *et al.* in [2], where reduced–complexity algorithms are presented for non–coherent block detection of differentially–encoded MPSK. It is shown through the use of computer simulations that decoding

complexity may be significantly reduced compared to a brute–force maximum–likelihood decoder, without sacrificing much in terms of performance. Finally, we examine the performance of some codes combined with the reduced–complexity decoding method over correlated Rayleigh fading channels.

Chapter 2

Non–Coherent Detection of Phase–Modulated Signals

2.1 Phase Shift Keying (PSK)

Phase Shift Keying (PSK) is a well-known digital modulation format which uses the phase of a modulated signal to convey information. More precisely, the information is conveyed via integers modulo–M which are then mapped into the signal space as M distinct phases of a carrier waveform. More often than not, the signal constellation is symmetric; that is, the M possible phases are equally spaced by $2\pi/M$ radians. We will be concerned solely with the symmetric case and will assume throughout this work that the mapping from the integers to the signal space points maintains numerical order in a counterclockwise fashion. The mapping for MPSK, $F_{MPSK} : \mathcal{Z}_M \rightarrow \mathcal{C}$, may therefore be expressed as

$$F_{MPSK}(\alpha) = \exp\left[j\left(\frac{2\pi}{M}\right)\alpha\right], \quad \alpha \in \mathcal{Z}_M \tag{2.1}$$

This situation is depicted in Fig.2.1 for three values for M.

We have, therefore, that the transmitted waveform for a single MPSK symbol of

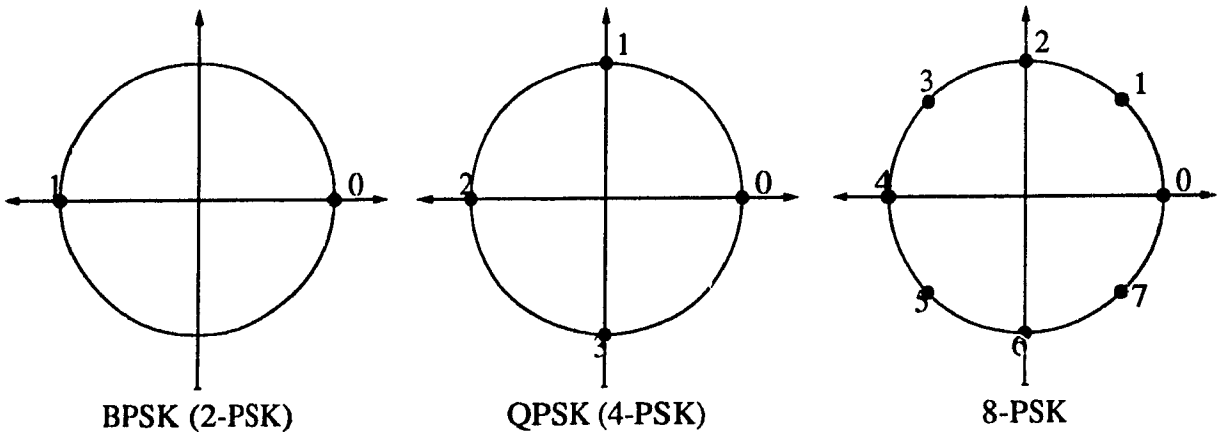

Figure 2.1: MPSK Signal Constellations for three values of M

duration T is represented by

$$s_m(t) = \mathrm{Re}\left\{\exp \jmath \left(2\pi f_c t + \frac{2\pi}{M} m\right)\right\} \quad m = 0, \ldots, M-1, \quad 0 \leq t \leq T \tag{2.2}$$

where f_c is the carrier frequency. Using the complex envelope notation (CE) and assuming transmission over an additive white Gaussian noise (AWGN) channel, we have that the signal upon reception is given by

$$\tilde{r}(t) = e^{\jmath\left(\frac{2\pi}{M} m + \phi\right)} \varrho(t) + \tilde{n}(t) \quad m = 0, \ldots, M-1, \quad 0 \leq t \leq T \tag{2.3}$$

where $\varrho(t)$ is defined as:

$$\varrho(t) = \begin{cases} 1, & \text{if } 0 \leq t \leq T; \\ 0, & \text{otherwise,} \end{cases} \tag{2.4}$$

ϕ is an unknown phase-shift induced by the channel, and the noise $\tilde{n}(t)$ is a complex white Gaussian process with zero mean and two-sided power spectral density N_0. The receiver must somehow extract the information, m, from this signal.

A receiver is said to be *coherent* if, by some means, it estimates the channel phase shift, ϕ, and uses this information in the detection process. If, however, the receiver treats ϕ as a nuisance parameter and and ignores it in detecting the received signal, it is said to be a *non-coherent* receiver. The main disadvantage attributed to a coherent receiver is due to the implementation complexity of estimating ϕ, which in

some cases can be significant. Moreover, there exist systems where fast carrier phase tracking is needed if coherent detection is to be used (TDMA and frequency hopping systems, for instance) which may prove to be impossible. In such instances, some form of non–coherent detection is required.

In comparing the two types of receivers, it should be noted that the reduced complexity associated with a non–coherent receiver does not come without some performance penalty. Typically, non–coherent systems exhibit some degradation in signal–to–noise ratio (SNR) at a given bit error rate when compared to ideal coherent systems. For the case of *uncoded* MPSK, there is no choice but to use a coherent receiver, since, as will be shown analytically later in this chapter, non–coherent detection cannot be used. Indeed, the focus of this work will be to consider methods for coding MPSK in order to perform non–coherent detection and, at the same time, outperform uncoded coherent detection.

By far the simplest method to avoid the need for a coherent receiver for detecting MPSK is to encode the information symbols differentially. More precisely, the information is not extracted from each symbol itself, but rather from the phase difference between adjacent symbols, which is impervious to any unknown phase. We may express the i^{th} transmitted phase as

$$\phi_i = \phi_{i-1} + \theta_i \tag{2.5}$$

where θ_i is the i^{th} information phase. This technique is known as *Differential Phase Shift Keying* (MDPSK). Although the detection of MDPSK is non–coherent, since no attempt is made at estimating the channel phase shift, it is usually referred to as being *differentially–coherent*. The reason for this is because of the fact that symbol decisions are made using the previous symbol as a phase reference, albeit a noisy one causing a degradation with respect to purely coherent MPSK. We will soon see that there is a close relationship between differentially–coherent detection and true non–coherent detection of MPSK, in the conventional sense. This type of detection provides a very practical alternative to coherent detection, and is often the modulation scheme chosen in many

practical situations, most notably for fading channels or short burst communications. The price paid for the reduced complexity of this scheme when compared with coherent detection can, however, be quite severe. While the degradation in SNR for BPSK (or 2–PSK) is negligible, which makes binary DPSK very popular, the degradation associated with larger constellations ($M \geq 4$) approaches 3dB and cannot be ignored when trying to avoid the complexity involved in estimating the carrier phase.

2.2 Non–Coherent Block Detection of MPSK

In this section the framework for detection of MPSK symbol blocks will be presented. Let us consider detecting blocks of N consecutive MPSK symbols. We may view the transmitted information as vectors of the form

$$\mathbf{c}_i = \begin{pmatrix} c_{i0} & c_{i1} & \cdots & c_{i(N-1)} \end{pmatrix}, \quad c_{ij} \in (0, \cdots, M-1). \tag{2.6}$$

If, in the detector, we use the duration of an entire block as the observation interval, the baseband equivalent of the received signal, $\tilde{r}(t)$, when $\mathbf{c}_i$ was transmitted is given by

$$\tilde{r}(t) = \sum_{l=0}^{N-1} e^{\jmath\phi} \exp\left[\jmath\left(\tfrac{2\pi}{M}\right) c_{il}\right] \varrho(t - lT) + \tilde{n}(t), \quad 0 \leq t \leq NT, \tag{2.7}$$

where $\varrho(t)$ is the baseband pulse shape as defined in (2.1).

2.2.1 Maximum Likelihood Detector

The maximum likelihood (ML) detector examines the likelihood functional, $p(\tilde{\mathbf{r}}(\mathbf{t})|\mathbf{c_m})$, over all possible $\mathbf{c_m}$, where $\mathbf{c_m}$ is the m^{th} possible transmitted vector, $0 \leq m \leq |\mathcal{V}|$, with $|\mathcal{V}|$ being the set of possible transmitted vectors. It then chooses the one which maximizes it as the most likely transmitted vector. Considering the unknown phase ϕ as a nuisance parameter, which is modelled as a random variable, we have that this

likelihood functional is given by

$$p(\tilde{r}(t)|\mathbf{c_m}) = \int_{-\pi}^{\pi} p_{\tilde{R}(t)|\mathbf{c_m},\Phi}(\tilde{r}(t)|\mathbf{c_m},\phi)p_{\Phi|\mathbf{c_m}}(\phi|\mathbf{c_m})d\phi, \tag{2.8}$$

where $p_{\tilde{R}(t)|\mathbf{c_m},\Phi}(\tilde{r}(t)|\mathbf{c_m},\phi)$ the probability distribution of the received signal assuming the transmitted codeword,$\mathbf{c_m}$, and the random phase, Φ are known. The quantity $p_{\Phi|\mathbf{c_m}}(\phi|\mathbf{c_m}$ is the probability distribution of Φ assuming $\mathbf{c_m}$ is known. Since the phase is independent of the transmitted vector, we have that

$$p_{\Phi|\mathbf{C_m}}(\phi|\mathbf{c_m}) = p_{\Phi}(\phi). \tag{2.9}$$

The absence of any information on the reference phase is described by a uniform distribution for Φ over $(\pi,\pi]$. Using this probability density, it can be shown [17, p.204] that the ML detector computes $|\mathcal{V}|$ decision variables according to the following rule,

$$U_m = \left|\int_0^{NT} \tilde{r}(t)s_m^*(t)dt\right| \quad m = 0,\cdots,|\mathcal{V}|-1, \tag{2.10}$$

where $s_m^*(t)$ is the baseband equivalent of the m^{th} transmitted signal given by

$$s_m^*(t) = \sum_{l=0}^{N-1} \exp\left[\jmath\left(\tfrac{2\pi}{M}\right)c_{ml}\right]\varrho(t-lT) \quad m = 0,\cdots,|\mathcal{V}|-1. \tag{2.11}$$

Inserting (2.11) into (2.10) and using (2.4) yields the following alternate expression for U_m

$$\begin{aligned} U_m &= \left|\sum_{l=0}^{N-1}\left[\int_{lT}^{(l+1)T}\tilde{r}(t)dt\right]f_{ml}^*\right| \\ &= \left|\sum_{l=0}^{N-1} y_l f_{ml}^*\right|, \end{aligned} \tag{2.12}$$

where y_l is the single symbol correlation of the l^{th} symbol in the received signal given by

$$y_l = \int_{lT}^{(l+1)T}\tilde{r}(t)dt = Te^{\jmath\phi}f_{il} + n_l \quad l = 0,\cdots,N-1,, \tag{2.13}$$

and f_{il} is the baseband equivalent of the l^{th} MPSK symbol given by $f_{il} = \exp\left[\jmath\left(\frac{2\pi}{M}\right)c_{il}\right]$ and n_l is a complex gaussian random variable with mean zero and variance N_0.

By forming the vector, $\mathbf{y} = (y_0 \quad y_2 \quad \cdots \quad y_{N-1})$, composed of the N single symbol correlations associated with the received signal, we may express (2.13) in vector notation as

$$\mathbf{y} = Te^{\mathcal{J}\phi}\mathbf{f_i} + \mathbf{n} \tag{2.14}$$

where $\mathbf{f_i} = (f_{i0} \quad \cdots \quad f_{i(N-1)})$ and $\mathbf{n} = (n_0 \quad \cdots \quad n_{(N-1)})$. The decision rule may, therefore, be expressed as

$$\max_{m=1,2,\cdots,|\mathcal{V}|} |\mathbf{f}_m\mathbf{y}^*|, \tag{2.15}$$

where $\mathbf{f}_m\mathbf{y}^*$ denotes the inner product between the vectors $\mathbf{f}_m$ and $\mathbf{y}$. This rule is simply an envelope detector which is essentially identical to the one used for non–coherent demodulation for M-ary signaling, however this is a discrete correlation over the entire received vector. The basic receiver structure using single symbol correlations is shown in Figure 2.2.

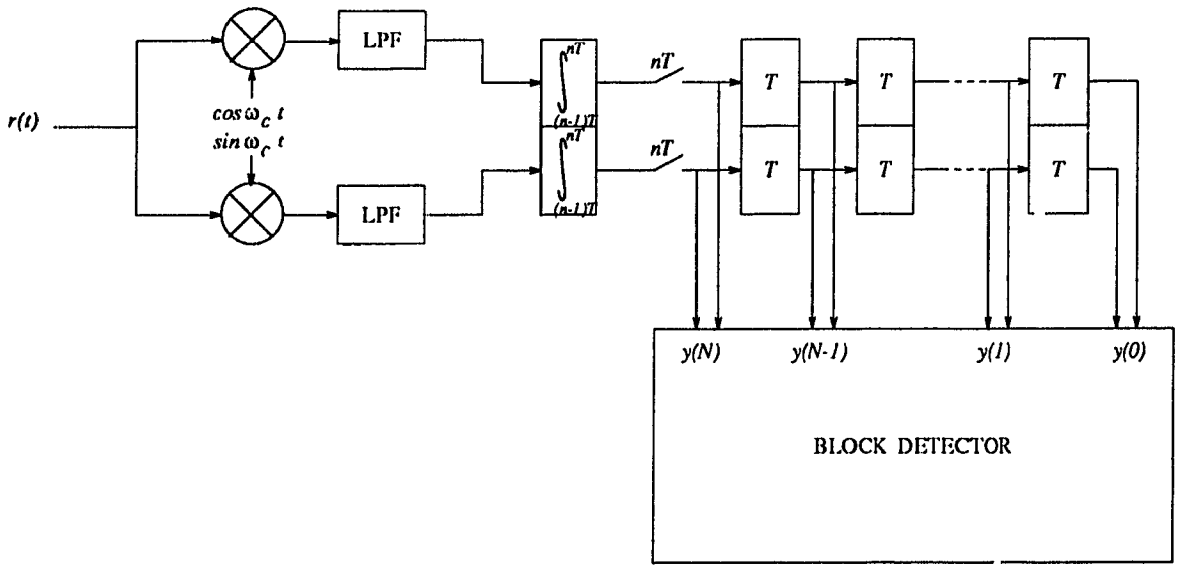

Figure 2.2: Basic Receiver Structure

The decision rule in (2.15), unfortunately, does not shed any light on the problem of how the maximization is to be carried out. A brute force approach would be to search through $\mathcal{V}$ one vector at a time. This method, however, has a complexity which will

usually increase exponentially with the block size N. It would be practical, therefore, to reduce this complexity by performing some sort of reduced search through $\mathcal{V}$, possibly with a minimal loss in performance. At the same time, imposing some structure on $\mathcal{V}$ should help in this regard.

2.2.2 Performance of the Maximum Likelihood Detector

Let us assume that the i^{th} vector was transmitted. We have, therefore, that the received vector is

$$\mathbf{y} = Te^{\jmath\phi}\mathbf{f}_i + \mathbf{n} \tag{2.16}$$

and that the decision rule may be expressed as

$$\max_{m=1,2,\cdots,|\mathcal{V}|} |TNe^{-\jmath\phi}\rho_{mi} + N_m| = U_m \tag{2.17}$$

where ρ_{mi} is the normalized complex correlation between the i^{th} and m^{th} transmitted vectors given by

$$\rho_{mi} = \frac{1}{N}\mathbf{f}_m\mathbf{f}_i^* = (1/N)\sum_{k=1}^{N}\exp\left[\jmath\left(\frac{2\pi}{M}\right)(c_{mk} - c_{ik})\right] \tag{2.18}$$

and N_m is a complex gaussian random variable with mean zero and variance $\sigma_N^2 = NN_0$ given by

$$N_m = \mathbf{f}_m\mathbf{n}^*. \tag{2.19}$$

If we separate N_m into its real and imaginary components which are independent and identically distributed

$$N_m = V_m + \jmath W_m \tag{2.20}$$

we have the decision variable, U_m, given by

$$\begin{aligned} U_m &= |V_m + TN\rho_{mi}\cos\phi + \jmath(W_m - TN\rho_{mi}\sin\phi)| \\ &= \sqrt{(V_m + TN\rho_{mi}\cos\phi)^2 + (W_m - TN\rho_{mi}\sin\phi)^2} \end{aligned}$$

This decision variable has a Rice distribution given by

$$p_{U_m}(u_m) = \frac{u_m}{\sigma_N^2} \exp\left[\frac{-(u_m^2 + T^2N^2\rho_{mi}^2)}{2\sigma_N^2}\right] \mathrm{I}_0\left(\frac{u_m TN\rho_{mi}}{\sigma_N^2}\right) \quad u_m \geq 0 \tag{2.21}$$

In order to determine the probability of error we follow a union bounding approach which indicates that the probability

$$P(m,i) = \Pr[U_i \leq U_m | \ \mathbf{f_i} \quad \text{transmitted}], \quad m = 1, 2, \cdots, |\mathcal{V}|, m \neq i \tag{2.22}$$

must be determined. Clearly, this may be expressed equivalently as

$$P(m,i) = \Pr[U_i^2 \leq U_m^2 | \ \mathbf{f_i} \quad \text{transmitted}], \quad m = 1, 2, \cdots, |\mathcal{V}|, m \neq i \tag{2.23}$$

which is shown in [17, p. 207] to be

$$P(m,i) = \mathrm{Q}(a,b) - \frac{1}{2} e^{-(a^2+b^2)/2} \mathrm{I}_0(ab) \tag{2.24}$$

where $\mathrm{Q}(a,b)$ is the Marcum Q-function and

$$\begin{aligned} a &= \sqrt{N\frac{\gamma}{2}\left[1 - \sqrt{1 - |\rho_{mi}|^2}\right]} \\ b &= \sqrt{N\frac{\gamma}{2}\left[1 + \sqrt{1 - |\rho_{mi}|^2}\right]} \end{aligned}$$

with γ being the SNR per symbol. For high SNR and $\rho_{mi} > 0$ this expression may be approximated by [3]

$$P(m,i) \approx \left[\frac{1 + |\rho_{mi}|}{2|\rho_{mi}|}\right]^{\frac{1}{2}} \mathrm{Q}\left[\sqrt{N\gamma(1 - |\rho_{mi}|)}\right] \tag{2.25}$$

The union bound for the overall probability of error associated with the i^{th} transmitted vector (for high SNR) may therefore be expressed as

$$P_e(i) \leq \sum_{\substack{m=1 \\ m \neq i}}^{|\mathcal{V}|} P(m,i) \tag{2.26}$$

It is useful to define a distance metric to express (2.25) in a manner analogous to the corresponding performance measure for coherent detection. The metric is termed the *non-coherent distance* between the m^{th} and i^{th} transmitted vectors and is given by

$$d_{NC}^2(m,i) = N(1 - |\rho_{mi}|) \tag{2.27}$$

so that (2.25) may be expressed as

$$P(m,i) \approx \left[\frac{N-d^2_{NC}(m,i)/2}{N-d^2_{NC}(m,i)}\right]^{\frac{1}{2}} \mathrm{Q}\left[\sqrt{\gamma d^2_{NC}(m,i)}\right] \tag{2.28}$$

The performance measure is therefore determined by the minimum d^2_{NC} over all pairs of transmitted vectors, as this is the dominant term in the union bound of (2.26). Equivalently, we may consider the maximum correlation magnitude as the performance measure.

It was previously mentioned that for uncoded MPSK it is impossible to perform non-coherent detection. This can be shown by computing the maximum correlation magnitude in this instance. For uncoded MPSK, the set $\mathcal{V}$ consists simply of all the possible length N vectors whose components are chosen from the integers modulo-M. If we examine the two vectors, $\mathbf{c}_a$ and $\mathbf{c}_b = \mathbf{c}_a + (\alpha \quad \alpha \quad \cdots \quad \alpha)$, we have that $|\rho_{ab}| = 1$ which forces $P(a,b) = 1/2$. This shows that the uncoded system cannot deal with phase ambiguities. Consequently, the system is rendered useless, which makes some form of coding indispensable, in order for true non-coherent detection to be possible.

2.3 Non-Coherent Detection of Coded MPSK

2.3.1 Multiple-Symbol Differential Detection of MPSK

As was already mentioned, the simplest way to code MPSK for non-coherent detection is differential encoding. We may consider, however, to implement a receiver based on the block detection process of section 2.2. This type of receiver and its associated performance are explored in [1, 3]. The results of these works will be summarized using the framework presented in 2.2.

We may view differential encoding as a means to remove the phase ambiguities which corrupt a non-coherent system. To see this, let us consider the detection of blocks

composed of N consecutive differentially encoded MPSK symbols. If the N information symbols which are to be transmitted are denoted by

$$s_i(n), \quad 1 \leq n \leq N, \quad 0 \leq i \leq 2^N - 1, \tag{2.29}$$

we have that after differential encoding, the transmitted vector $v_i(n)$ (before modulation) is given by

$$v_i(n) = \left(v_i(0) + \sum_{k=1}^{n} s_i(k) \right) \bmod M \quad 1 \leq n \leq N, \tag{2.30}$$

where $v_i(0)$ is the last symbol of the previously transmitted vector. By expanding out the dot product in (2.15) for diffentially–encoded blocks, the decision rule may be expressed as

$$\max_i \left| \exp\left\{ j \left(\frac{2\pi}{M}\right) v_i(0) \right\} y^*(0) + \sum_{n=1}^{N} y^*(n) \exp\left\{ j \left(\frac{2\pi}{M}\right) \left[v_i(0) + \sum_{k=1}^{n} s_i(k) \right] \right\} \right|.$$

Since the term $e^{-j\left(\frac{2\pi}{M}\right)v_i(0)}$ can be factored out from each term in the expansion, the decision rule becomes

$$\max_i \left| y^*(0) + \sum_{n=1}^{N} y^*(n) \exp\left\{ j \left(\frac{2\pi}{M}\right) \left[\sum_{k=1}^{n} s_i(k) \right] \right\} \right|. \tag{2.31}$$

The first term in (2.31) clearly shows that the each vector in $\mathcal{V}$ begins with a zero and, therefore, we see that the set $\mathcal{V}$ in (2.15) is simply all the 2^N vectors of length $N+1$ which begin with a zero. We have, therefore, that

$$\forall \mathbf{v} \in \mathcal{V} \quad , \quad \mathbf{v} + (\alpha \quad \alpha \quad \cdots \quad \alpha) \notin \mathcal{V}, \tag{2.32}$$

since $\mathbf{v} + (\alpha \quad \alpha \quad \cdots \quad \alpha)$ does not begin with a zero. This means that by differentially encoding the information vector, $s_i(n)$, we can assure that phase ambiguities are removed.

The decision rule in (2.31) yields the complete receiver structure shown in Fig.2.3. A modification to 2.31 yields an equivalent structure with comparable complexity[3]. Divsalar *et al.* [1] also consider a serial implementation of the receiver.

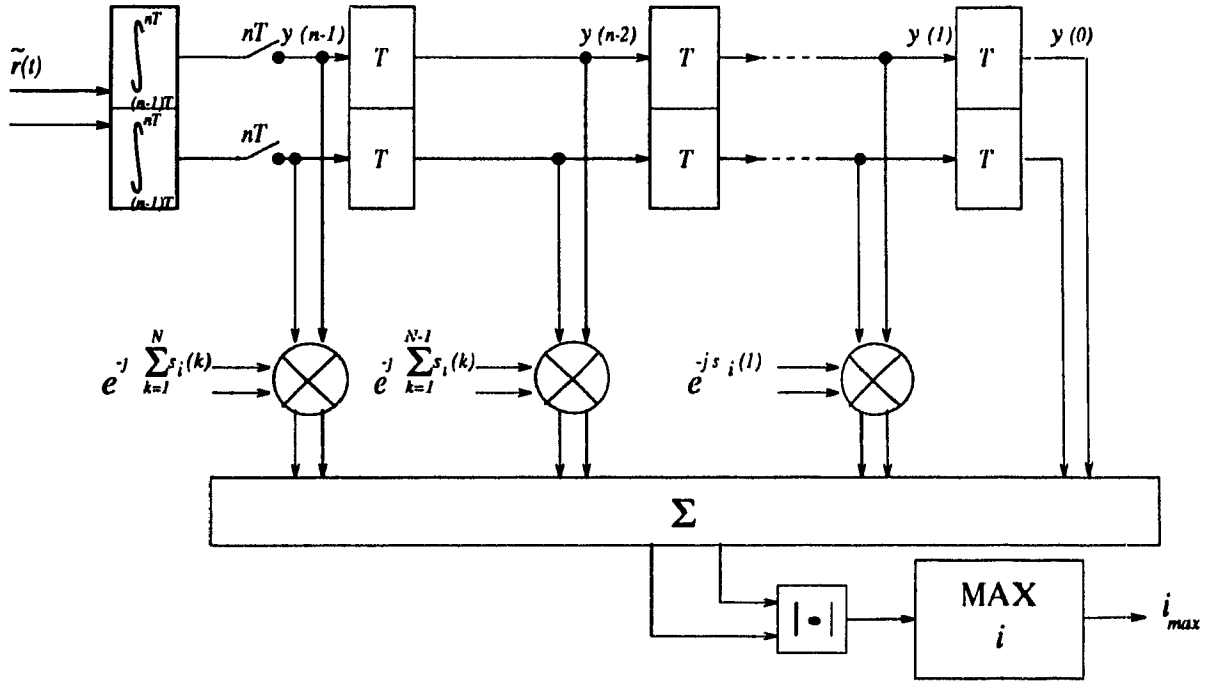

Figure 2.3: Complete Receiver Structure

As was alluded to earlier, it can be shown that the decision rule (2.31) for the case $N = 1$ is completely equivalent to that of conventional DPSK detection. Expressing (2.31) for $N = 1$ we have

$$\max_{\mathbf{i}} \left| y^*(0) + y^*(1) \exp\left\{ -j \left(\frac{2\pi}{M}\right) s_i(1) \right\} \right| \tag{2.33}$$

which may be expressed equivalently as

$$\max_{\mathbf{i}} \left\{ |y(0)| + |y(1)| + 2|y(0)||y(1)| \cos\left[\angle y(1) - \angle y(0) - \left(\frac{2\pi}{M}\right) s_i(1) \right] \right\}, \tag{2.34}$$

and reduces to

$$\max_{\mathbf{i}} \cos\left[\angle y(1) - \angle y(0) - \left(\frac{2\pi}{M}\right) s_i(1) \right], \tag{2.35}$$

or,

$$\min_{\mathbf{i}} \left| \angle y(1) - \angle y(0) - \left(\frac{2\pi}{M}\right) s_i(1) \right| \bmod 2\pi, \tag{2.36}$$

which is the classic differentially–coherent receiver. We may say, therefore, that differentially–coherent detection is in fact non–coherent, even in the conventional sense. The performance of this scheme approaches that of coherent MPSK as the block length, N is

increased. Explicitly, the asymptotic degradation in SNR, in terms of the block length N, was shown in [3] to be

$$D(N) = \frac{N + 1 + \sqrt{N^2 + 1 + 2N \cos\frac{2\pi}{M}}}{2N}. \tag{2.37}$$

Performance values for various block lengths for QPSK and 8–PSK are shown in Table 2.1(taken from [3]). It is seen that simply adding one additional symbol to the observation interval greatly improves performance over differentially-coherent detection ($N = 1$).

M	N					
	1	2	3	4	5	6
4	2.3	1.2	0.77	0.57	0.45	0.37
8	2.8	1.6	1.1	0.86	0.70	0.58

Table 2.1: SNR degradation in dB of optimal non-coherent block detection of MPSK (from [3])

2.3.2 Reduced–complexity receiver structures

The receivers of [1, 3] perform maximum likelihood detection in a brute force fashion. That is, they search exhaustively through all possible transmitted vectors for the one which maximizes the decision rule in (2.15). This method, of course, has exponential complexity ($O(M^N)$), which may be undesirable for large N. It is worthwhile, therefore, to consider receiver structures with reduced complexity, which may suffer minimal performance loss in comparison to the maximum likelihood receiver. This type of receiver is considered in [2]. Two methods are proposed in this work, both of which attempt to significantly reduce the search space of the maximum likelihood decoder without sacrificing too much in terms of performance. This reduction in search time is achieved by making premature decisions on some of the symbols in the block using single symbol differential detection.

In the first method, named the 2^L algorithm, decisions are made on each symbol in the block using single symbol differential detection. The $N-L$ most reliable symbols are then fixed at these decisions and the set of 2^L candidate vectors is formed by filling in all possible combinations of best and second best choices in the remaining L positions. Maximum–likelihood detection is then performed on this set of candidates via (2.15), resulting in a complexity of $O(2^L)$. It was found by computer simulation that L could be chosen to be less than K, thereby greatly reducing the search complexity, without significantly affecting performance. Even for the case $L = 1$ noticeable improvement is achieved over ordinary differential detection.

The second method, denoted the $N+1$ algorithm, also performs single symbol differential detection on the entire block, but reduces the search space in a different fashion. This algorithm has complexity $O(N)$ and adopts the philosophy that at most a single error occurs in each block if single symbol differential detection is used. Using this assumption, $N+1$ candidate vectors are formed by placing a second best choice in one position and fixing the remaining $N-1$ positions at their best choice. The performance of this method for reasonably small N is comparable to that of optimal block detection, and yet is achieved with a significant reduction in complexity.

In [12] Samejima *et al.* explore another block detection strategy which exploits the inherent coding of MDPSK in order to perform non–redundant error correction. Although their method is rather different than the non–coherent block detection schemes just presented, it can be considered as being a reduced–complexity block detection scheme since it is quite simple to implement. This work considers using L detectors in parallel, each of which performs the difference between the i^{th} and $(i-k)^{th}$ symbols $(k = 1, \cdots, L)$ as shown in Fig.2.4. It is shown that this may be viewed as a rate $1/L$ convolutional encoder, whose corresponding code is capable of correcting $L-1$ errors. Using this realization an improved receiver is developed which significantly bridges the gap between differentially–coherent MPSK and coherent MPSK. The receiver makes use

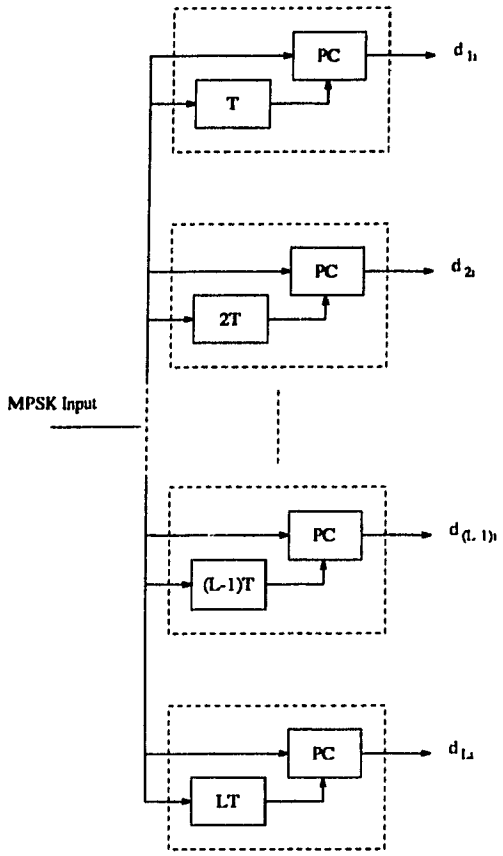

Figure 2.4: Samejima's L stage detector

of a syndrome decoding circuit, based on the parity–check matrix of the code corresponding to the L parallel detectors, in order to correct errors in the block. The performance of this technique is determined analytically by evaluating the occurrence probabilities of the most likely error patterns at the output of the L stage detector. In order to verify the correctness of the analytical results an experimental circuit was tested for QPSK and was shown to perform very close to theoretical predictions. For QPSK and using $L = 2$ it is shown that a 1.2dB improvement over differentially–coherent detection can be obtained. By choosing $L = 3$ this may be increased to 1.7dB. The remainder of the work considers three important practical issues and how they effect the performance of the system, namely the effects of receiving filter bandwidth, carrier frequency offset and an unbalanced receiver.

The results of the two reduced–complexity schemes is summarized in Table 2.2, where the degradation with respect to coherent detection of the various schemes is presented. In order to avoid confusion, we have used N instead of L to indicate the order of the Samejima *et al.* receiver. Wilson *et al.* only consider the decoding of 8–PSK symbol blocks, since a much larger reduction in complexity is attainable, in comparison with QPSK, by using the 2^L or $N+1$ algorithms instead of a ML decoder. Samejima *et al.*, however, only consider their receiver structure for QPSK, and, therefore, a fair comparison of the two strategies cannot be made. We may, however, compare both to optimal non–coherent block detection of differentially–encoded MPSK, which is shown in the last column of Table 2.2. We see that the 2^L method attains optimal performance, even for $L < N$, which allows for a significant reduction in complexity without suffering a performance penalty. The $N+1$ method, on the other hand, comes close to optimal performance for moderate block sizes, whereas for longer block sizes, in this case $N = 10$, there is a significant performance penalty. Samejima's method, at least for the small block sizes considered, performs almost as well as the optimal non–coherent receiver, although it is based on an entirely different premise. More precisely, they look at demodulation as a decoding problem, which, ten years ago was a pioneering approach to what is well–known and accepted today.

2.3.3 Error–Control Coding

We have seen that differential encoding, a rather simplistic form of coding, can be used in non–coherent systems with a minimum performance penalty with respect to coherent detection, when the block length is increased. We would therefore like to consider more powerful coding techniques so as even to surpass the performance of uncoded coherent detection.

Performance enhancement via error–control coding over the gaussian channel is achieved by adding some sort of redundant information to the signal. This redundancy is

Method	M	N	L	Degradation(dB)	Optimal(dB)
2^L	8	5	1	1.6	$0.8(P_b = 10^{-5})$
2^L	8	5	3	0.8	$0.8(P_b = 10^{-5})$
2^L	8	5	5	0.8	$0.8(P_b = 10^{-5})$
$N+1$	8	3	-	1.4	$1.0(P_b = 10^{-5})$
$N+1$	8	5	-	1.0	$0.8(P_b = 10^{-5})$
$N+1$	8	10	-	1.0	0.4 (SNR$\rightarrow \infty$)
Samejima	4	2	-	1.3	1.2 (SNR$\rightarrow \infty$)
Samejima	4	3	-	0.8	$0.8(P_b = 10^{-6})$

Table 2.2: Performance comparison of reduced complexity strategies

exploited during the decoding process so as to correct errors introduced by the channel. The two types of error-control codes which are most widely used in practice are *block codes* and *convolutional codes*. In this work we will consider only the former, and how they may be used effectively with non-coherent detection of MPSK modulation.

In mathematical terms, a block code is simply a mapping from a K-dimensional space onto an N-dimensional space; that is, vectors of length K over some specific symbol alphabet, are transformed into vectors of length N, with the redundancy being reflected in the $N - K$ additional symbols. The ratio $R_c = K/N$ is known as the code rate, and reflects the code's redundancy. The codewords are usually selected such that they are as far apart as possible, according to some distance measure, or *metric*. More precisely, codes are constructed so that the minimum distance between all pairs of codewords is maximized. A block code is said to be *linear*, in the strictest sense, if the codewords form an algebraic group. Very often, however, additional structural constraints are placed on the codes, more often than not to facilitate construction and to allow for efficient decoding strategies. For example, many of the existing block codes are subspaces of some abstract vector space.

Traditionally, algebraic coding techniques considered coding and modulation as

distinct entities. They main problem with this approach is matching the distance measure used in the design of the code with that of the channel. This is important because distances in the code space are not necessarily preserved after modulation, which can be considered as an abstract mapping from the code space onto the signal space. Modulation can be interpreted, therefore, as a function which warps the distance profile of a code. Consequently, a code with excellent distance properties in the code space may be completely useless when combined with certain modulation formats. There are, of course, certain algebraic metrics which are useful with some modulation formats. For example, the *hamming distance*, which is used extensively for binary codes and occasionally non–binary codes as well, is only useful on a coherent Gaussian channel for binary and ternary modulation [5]. The *Lee metric*, on the other hand, is suitable for M–ary phase modulated signals over the coherent Gaussian channel. These two metrics are useful in these instances, because they are closely related to *euclidean distance* in the signal space, the distance measure associated with perfectly coherent detection. The hamming distance in these two instances is linearly related to euclidean distance, whereas the Lee–metric is a close approximation for phase–modulated signals [4].

In order to alleviate the problem of preserving distances after modulation, it was later suggested that considering coding and modulation as a combined entity may yield very fruitful performance rewards [9]. The rational behind this suggestion is that by using the channel's distance measure, or the distance metric in the signal space, we can find a code which maximizes the performance for a given modulation format directly. Ungerboeck's *trellis–coded modulation* [6] revolutionized this realization, and as a result many practical systems have emerged using his techniques. Given the success of this technique it would be natural to attempt to design codes according to the distance measure for non–coherent block detection of MPSK given in (2.27), which will be the subject of the next chapter.

2.3.4 Error–Control in Non–Coherent Systems

In comparison with coherent systems, very little attention has been given to the coding problem for non–coherent systems *per se*. Nevertheless, some methods exist for coding of MPSK for use with differentially–coherent detection. In [10], Nakamura develops a class of linear codes over the ring of integers modulo M for use with MDPSK channels, which is the natural choice for a symbol alphabet to use in conjunction with MPSK. These are cyclic codes designed for the Lee metric, and their construction is somewhat reminiscent to that of the well–known BCH codes. Codes of various rates for $M = 4$ and $M = 8$ are presented which are capable of correcting all single and double Lee errors. This work only considers the problem of constructing codes for the Lee metric and presents the associated algebraic concepts required. Unfortunately, no mention is made concerning the performance of the codes over gaussian channels and of decoding strategies. Although incomplete, this work does shed light on the problem of finding an appropriate algebraic structure for coding MPSK, namely codes defined over the ring of integers modulo M.

Rhodes considers binary block codes for use with binary DPSK modulation in [11]. These codes are made up of $N = 2^n$ orthogonal codewords of length N, n being the number of information bits, and are a generalization of the single symbol($N = 1$) case. The N codewords are formed using all possible linear combinations of a basis set of n codewords plus the all–zero codeword. Tables 2.3 and 2.4 show the construction of the codes for the case $N = 8$ and are taken from [11]. The idea of differentially–coherent detection is naturally extended to differential detection between code blocks. The performance of these codes is determined analytically and results are presented for both coherent and non–coherent binary DPSK and show that significant coding gain may be obtained with reasonably small block lengths. These gains are summarized for non–coherent detection in Table 2.5. The main disadvantage of these codes is that in order to achieve significant coding gain, the code rate must be quite low, which implies

Basis Number	Codeword
0	0 0 0 0 0 0 0 0
$2^0 = 1$	0 1 0 1 0 1 0 1
$2^1 = 2$	0 0 1 1 0 0 1 1
$2^2 = 4$	0 0 0 0 1 1 1 1

Table 2.3: Extended basis set for generating an orthogonal code of length N=8

Orthogonal Word Number	Terms of Basis Set	Codeword
0	0	0 0 0 0 0 0 0 0
1	1	0 1 0 1 0 1 0 1
2	2	0 0 1 1 0 0 1 1
3	1+2	0 1 1 0 0 1 1 0
4	4	0 0 0 0 1 1 1 1
5	1+4	0 1 0 1 1 0 1 0
6	2+4	0 0 1 1 0 0 1 1
7	1+2+4	0 1 1 0 1 0 0 1

Table 2.4: Extended basis set for generating an orthogonal code of length N=8

that substantial bandwidth expansion is needed. Secondly, the work only addresses the binary case which, although important, is quite limiting. In terms of complexity, this system, in one particular instance ($N = 16$), is roughly comparable to that of a 16-state convolutional code with Viterbi decoding, both of which, using coherent detection, share similar performance characteristics.

In [13], Divsalar *et al.* apply the idea of multiple-symbol detection to trellis-coded MDPSK which, however, only yields marginal improvement at the expense of a significant increase in complexity. Their decoder uses a sub-optimal modification of the Viterbi Algorithm which uses multiple-symbol decisions to make up the path metrics. Two examples of codes are given. The first is a two-state rate 1/2 trellis-coded DQPSK

N	Asymptotic coding gain relative to uncoded DPSK(dB)
4	3.0
8	4.0
16	4.8
32	5.4

Table 2.5: Asymptotic Performance of Rhodes' scheme for different values of N

system. At a bit error rate of 10^{-5} there is an improvement of approximately 0.25dB going from a conventional $N = 2$ receiver to one with $N = 3$. Both simulation and analytical results are given to demonstrate the improvement. The second example is a 16-state, rate 2/3 trellis-coded 8-DPSK system. Using only simulations, it is shown that at a bit error-rate of 10^{-4}, an approximate 0.75dB improvement is attainable by going from a conventional $N = 2$ receiver to an $N = 3$ receiver.

Recently, there have been some simple codes developed for non-coherent block detection of *Minimum Shift Keying (MSK)* [14]. Although this is a modulation scheme rather different than MPSK, the detection process is somewhat similar since the distance metric is identical. This work explores the use of simple binary block codes to significantly improve performance over an uncoded MSK system. The code redundancy is designed so as to combine effectively with the inherent MSK redundancy and to increase the minimum non-coherent distance. The performance enhancement is achieved by determining the most likely error patterns, or equivalently those codewords which have the smallest non-coherent distance from the zero codeword. Once determined, as many of these as possible are excluded from the code by choosing an appropriate parity-check matrix which, in turn, defines the code. These codes expand bandwidth in order to achieve coding gain over an uncoded system.

Chapter 3

A Class of Codes for Non–Coherent Detection of MPSK

3.1 The ring $\mathcal{Z}_M$ and MPSK

As was stated at the beginning of the last chapter, the mapping for MPSK, F_{MPSK} : $\mathcal{Z}_M \rightarrow \mathcal{C}$,

$$F_{MPSK}(\alpha) = \exp\left[j\left(\frac{2\pi}{M}\right)\alpha\right], \quad \alpha \in \mathcal{Z}_M, \tag{3.1}$$

translates the ring of integers modulo–M, $\mathcal{Z}_M$, into M distinct phasors along the unit circle. For the moment, let us consider $\mathcal{Z}_M$ only as an algebraic group under addition modulo–M, and note that the set of MPSK signal points also forms a group under complex multiplication. The important aspect about the mapping $F_{MPSK} : \mathcal{Z}_M \rightarrow \mathcal{C}$, is that it is an *isomorphism.* An isomorphism is defined as follows:

Definition 1 *An* isomorphism *between two groups* $(\mathcal{G}, \cdot)$ *and* $(\mathcal{H}, \circ)$ *is a one-to-one mapping between the elements of* $\mathcal{G}$ *and* $\mathcal{H}$ *such that, if* $a_G \in \mathcal{G} \leftrightarrow a_H \in \mathcal{H}$ *and* $b_G \in \mathcal{G} \leftrightarrow b_H \in \mathcal{H}$ *then* $a_G \cdot b_G \leftrightarrow a_H \circ b_H$

Clearly, by using addition modulo-M as the composition operation for $\mathcal{Z}_M(\mathcal{G})$ and similarly complex multiplication for the set of MPSK signal points($\mathcal{H}$), the mapping $F_{MPSK} : \mathcal{Z}_M \rightarrow \mathcal{C}$ is an isomorphism. This is important since it allows us to use $\mathcal{Z}_M$ and the set of MPSK signal points interchangeably, which simplifies the matter of designing codes specifically for MPSK modulation.

Another aspect concerning a mapping which also plays an important role in the construction of codes for a particular signal set, is whether or not it is *matched* for a particular distance measure,$d(\cdot,\cdot)$ in the signal space. A definition for "matching" is as follows

Definition 2 *A mapping μ from a group $(\mathcal{G},\cdot)$ onto a signal set $\mathcal{S}$ is a* matched mapping *for a particular distance measure $d(\cdot,\cdot)$ if, for all g and g' in $\mathcal{G}$,*

$$d(\mu(g),\mu(g')) = d(\mu(g^{-1}\cdot g'),\mu(\epsilon)), \tag{3.2}$$

where ϵ is the identity zero element of $(\mathcal{G},\cdot)$ and g^{-1} is the inverse of the element g.

This definition is similar to one given in [16], except that we do not make the assumption that the distance measure $d(\cdot,\cdot)$ is a proper metric.

In our case the distance measure is the non–coherent distance $d^2_{NC}(\cdot,\cdot)$ defined in (2.27) of the previous chapter. This distance is defined for blocks of MPSK symbols, and therefore we must consider the extension group of $\mathcal{Z}_M$, denoted $\mathcal{Z}_M^N$. $\mathcal{Z}_M^N$ is simply the set of length N vectors whose components are elements of $\mathcal{Z}_M$. It is clear that $\mathcal{Z}_M^N$ is also a group if we consider componentwise addition as the composition operation. Suppose that we have a collection of vectors $\mathcal{C}$ which form a subgroup of $\mathcal{Z}_M^N$, and consider two members of $\mathcal{C}$, $\mathbf{c_i}$ and $\mathbf{c_j}$. As in the previous chapter, ρ_{ij} is defined as the complex correlation coefficient between the i^{th} and j^{th} transmitted vectors given by

$$\rho_{ij} = \frac{1}{N}\mathbf{f_i}\mathbf{f_j^*} = \frac{1}{N}\sum_{k=1}^{N}\exp\left[\jmath\left(\frac{2\pi}{M}\right)(c_{ij}-c_{jk})\right]. \tag{3.3}$$

Since $\mathbf{c_i}$ and $\mathbf{c_j}$ are elements of a group, $\mathbf{c_i} - \mathbf{c_j}$ must also be an element,$\mathbf{c_q}$, of the same group. We have, therefore, that

$$\rho_{ij} = \rho_{q,0}, \tag{3.4}$$

where 0 corresponds to the all-zero codeword,$\mathbf{c_0}$, or the identity element of $\mathcal{C}$. From this, it is clear that

$$d^2_{NC}(F_{MPSK}(\mathbf{c_i}), F_{MPSK}(\mathbf{c_j})) = d^2_{NC}(F_{MPSK}(\mathbf{c_q}), F_{MPSK}(\mathbf{c_0})), \tag{3.5}$$

where the mapping F_{MPSK} now operates componentwise on a vector in $\mathcal{Z}_M$, and, therefore, the mapping is matched for the non-coherent distance.

3.2 Module–Phase Codes

We now consider an algebraic framework for block-coded M–PSK modulation. First of all, we would like the codes to be subgroups of $\mathcal{Z}_M^N$, so that they are *linear* or *group* codes. A second reason for this restriction relates to the idea of matching. If the codes are subgroups of $\mathcal{Z}_M^N$ we have seen that the mapping from the code vectors to the signal space is matched for the non–coherent distance. This is important because it is related to the distance profile of the code. Since the mapping is matched, we need only determine the distances from each codeword to the all–zero codeword, since the distance between any arbitrary pair of codewords may be computed using (3.5). This assures that the codes are *superlinear*[18].

Up until this point we have considered only the additive properties of $\mathcal{Z}_M$ and it sufficed, therefore, to consider it as a group, and similarly $\mathcal{Z}_M^N$ as its extension group. In fact, since $\mathcal{Z}_M$ is actually a ring, we may look at $\mathcal{Z}_M^N$ in a more flexible way, from the standpoint of coding. It is actually an example of an another algebraic abstraction known as a *module*. A module, simply put, is the generalization of a vector space, where the scalars now belong to an arbitrary ring rather than to a field. The codes we are

considering, called *module–phase codes*, are sub–modules of $\mathcal{Z}_M^N$, which are still groups as before, but have additional properties which will prove most useful both in the definition of codes over $\mathcal{Z}_M$ and in their construction for the non–coherent distance measure, d_{NC}^2.

Sub–modules are analogs of sub–spaces of a vector space, and are themselves modules, hence the name *module–phase codes.* This framework has previously been proposed for block codes over $\mathcal{Z}_M$ for the Hamming metric [19] and Lee–metric [20]. Recalling the properties of a $\mathcal{Z}_M$–module, we have that a code $\mathcal{C}$ over $\mathcal{Z}_M$ is such that

1. $\mathcal{C}$ forms a commutative group under vector addition
2. $\forall \mathbf{x}, \mathbf{y} \in \mathcal{C} \text{and} \lambda, \mu \in \mathcal{Z}_M$ we have
   (a) $\lambda(\mu\mathbf{x}) = (\lambda\mu)\mathbf{x}$
   (b) $1\mathbf{x} = \mathbf{x}$
   (c) $(\lambda + \mu)\mathbf{x} = \lambda\mathbf{x} + \mu\mathbf{x}$
   (d) $\lambda(\mathbf{x} + \mathbf{y}) = \lambda\mathbf{x} + \lambda\mathbf{y}$

The concept of linear independence for modules remains the same as in the case of vectors over fields. The codes are *free* modules which have the property that they may be finitely generated by a set of linearly independent vectors, called generators, $\mathbf{g}_1, \mathbf{g}_2, \cdots, \mathbf{g}_K \in \mathcal{Z}_M^N$, which are said to form a basis for $\mathcal{C}$. We have, therefore, that every codeword $\mathbf{c}_m \in \mathcal{C}$ can be expressed as

$$\mathbf{c}_m = \sum_{i=1}^{K} x_{mi}\mathbf{g}_i, \tag{3.6}$$

where $x_{mi} \in \mathcal{Z}_M$. We can define this equivalently as

$$\mathbf{c}_m = \mathbf{x}_m \begin{pmatrix} \mathbf{g}_1 \\ \mathbf{g}_2 \\ \vdots \\ \mathbf{g}_K \end{pmatrix} = \mathbf{x}_m \mathbf{G}, \tag{3.7}$$

where $\mathbf{G}$ is the $(N-K)\times N$ generator matrix for $\mathcal{C}$ and $\mathbf{x}_m = (\, x_{m1} \quad \cdots \quad x_{mK} \,)$ is the information vector. The K-dimensional information vectors, $\mathbf{x}_m$, belong to the module $\mathcal{Z}_M^K$. Therefore, a module code is the image of an injective homomorphism from $\mathcal{Z}_M^K$ to $\mathcal{Z}_M^N$ determined by the generator matrix $\mathbf{G}$.

Since we are using matrices over rings, it is important to note the similarities and differences with the case of matrices over fields. The only concepts that will be of interest to us are *rank*, *singularity*, and *elementary row operations*. The rank of a matrix is identical to the case over fields, it is simply the maximum number of linearly independent rows. The generator matrices for our codes, for example, have rank K. The idea of singular matrices differs somewhat from the case over fields. In general, a matrix is singular if its determinant is an element which is not invertible. In the case of fields, all elements except zero are invertible, and therefore we need only assure that the determinant is non–zero to assess whether or not a matrix is non–singular. In general, rings have elements other than zero which are not invertible, and are known as *zero divisors*. As is the case over fields, a non–singular matrix is invertible. The concept of elementary row operations on a matrix is also similar, with the added restriction that only multiplication of a row by an invertible element is permitted.

The only restriction that we put on the rows of the generator matrix, $\mathbf{G}$, is that they are linearly independent. It follows, therefore, that there must exist at least one set of K columns from $\mathbf{G}$ such that the square matrix made up of these columns is non–singular. Let us denote this matrix by $\mathbf{Q}$, and define a new generator matrix $\mathbf{G}'$ given by

$$\mathbf{G}' = \mathbf{Q}^{-1}\mathbf{G}. \tag{3.8}$$

Clearly, $\mathbf{G}'$ is a generator matrix for the same code. The only thing that is altered is the mapping from the information vectors $\mathbf{x}_m$ to the codewords in $\mathcal{C}$. What is useful about this new generator matrix, however, is that is always places an exact copy of the information vector in the K positions of each codeword corresponding to the K columns

chosen for $\mathbf{Q}$. Such a matrix is said to be in *systematic form.* In other words, the K columns in $\mathbf{G}'$ that correspond to the K columns chosen for $\mathbf{Q}$ contain the $K \times K$ identity matrix, $\mathbf{I_K}$. Since interchanging columns does not alter the code, we may obtain a new generator matrix by permuting the columns of $\mathbf{G}'$ such that the K information symbols occur in first K positions of every codeword. Using these suitable transformations, we may always express any generator matrix in systematic form as

$$\mathbf{G} = (\,\mathbf{I}_K \;\;|\;\; \mathbf{p}_1^T \;\; \cdots \;\; \mathbf{p}_{(N-K)}^T\,). \tag{3.9}$$

The codes may be equivalently defined as the vectors from $\mathcal{Z}_M^N$ which belong to the kernel of an $(N-K) \times N$–dimensional matrix $\mathbf{H}$ of rank $N-K$. We have, therefore, that

$$\forall \mathbf{c} \in \mathcal{C} \iff \mathbf{c}\mathbf{H}^T = 0. \tag{3.10}$$

Since the generators are themselves codewords, it follows that $\mathbf{G}\mathbf{H}^T = \mathbf{0}$ and the corresponding form for $\mathbf{H}$ is given, therefore, by

$$\mathbf{H} = \left(\begin{array}{c|c} \begin{matrix} -\mathbf{p}_1 \\ \vdots \\ -\mathbf{p}_{(N-K)} \end{matrix} & \mathbf{I}_{N-K} \end{array}\right). \tag{3.11}$$

3.3 A Method for Building Codes

Using the algebraic structure just presented, we present a technique for finding a set of generators that define a submodule of $\mathcal{Z}_M^N$ with reduced maximum correlation magnitude $|\rho_{max}|$. It is this reduction in $|\rho_{max}|$ which leads to improved performance over an uncoded system. In order to quantify the performance enhancement achieved by coding, we will use the measure known as *coding gain.* Coding gain is defined as the difference in decibels(dB) between the signal–to–noise ratios(SNR) of a coded scheme and an uncoded reference scheme operating at the same error probability. For convenience, all of the codes that will be presented in this chapter will be compared to uncoded coherent

MPSK. This is done because the performance of uncoded coherent MPSK is always a lower bound to any of the non-coherent block detection schemes, and therefore if these coded schemes outperform coherent MPSK, they will necessarily outperform any block detection scheme and clearly differentially-coherent detection as well.

We note that the probability of symbol error for uncoded coherent MPSK is given approximately by [2]

$$P_e \simeq 2\mathrm{Q}\left(\sqrt{\gamma(1-\cos\frac{2\pi}{M})}\right). \tag{3.12}$$

Similarly for an (N,K) coded scheme with non-coherent distance d^2_{NC} at high SNR, the probability of codeword error with non-coherent detection is given approximately by

$$P_{codeword} \simeq N_n \left[\frac{N+1-d^2_{NC}/2}{N+1-d^2_{NC}}\right]^{\frac{1}{2}} \mathrm{Q}\left(\sqrt{\left(\frac{K}{N}\right)\gamma d^2_{NC}}\right), \tag{3.13}$$

where N_n is the number of codewords which share the distance, d^2_{NC}. The factor K/N is due to bandwidth expansion, since K information symbols are contained in each codeword of length N. We have, therefore, that the asymptotic coding gain is given by

$$G_c = 10\log\left[\left(\frac{K}{N}\right)\frac{d^2_{NC}}{1-\cos\frac{\pi}{M}}\right]\mathrm{dB}. \tag{3.14}$$

If we look at the set of possible correlations created by the N-tuples $\mathcal{Z}_M^N$, as depicted in Fig.3.1 for $\mathcal{Z}_8^6$, we would like to choose a submodule such that $|\rho_m| \leq r_{max}$, for all m. This amounts to creating a code which selects a set of correlations that lie within a circle of radius r_{max}, and therefore has $d^2_{NCmin} \geq N(1-r_{max})$. The best scenario would be if the set of correlations were very packed, as opposed to being spread throughout the range of possible values. An example is shown in Fig. 3.2, where we show a possible set of correlations for a code which has $d^2_{NCmin} = 2$.

3.3.1 Code Design by Exclusion of Unwanted Vectors

In order to choose submodules of $\mathcal{Z}_M^N$ which limit the correlation magnitude, the vectors with large $|\rho_m|$ in $\mathcal{Z}_M^N$ must be determined. Let us assume that we want to generate

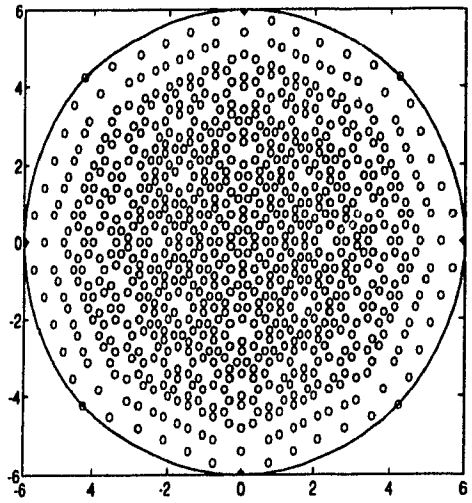

Figure 3.1: Possible correlation values for an uncoded system in $\mathcal{Z}_8^6$

a code which has a maximum correlation magnitude, $|\rho_{\max}|$. The following method attempts to exclude these vectors from the desired code, C. Let us create a list, L_0, which consists of the vectors that are to be excluded from C. These are the vectors with $|\rho_m| > |\rho_{\max}|$. Consider now an N–dimensional vector given by

$$\mathbf{h}_0 = (\, h_{i1} \quad h_{i2} \quad \cdots \quad h_{i(N-1)} \quad 1\,). \tag{3.15}$$

We will choose the initial code, C_0, as the kernel of $\mathbf{h}_0$, which is chosen so as to exclude as many vectors from L_0 as possible. The kernel of $\mathbf{h}_0$ defines a rate $(N-1)/N$ code having a generator matrix given by

$$\mathbf{G}_0 = [\mathbf{I}_{N-1} \mid (\, -h_{01} \quad -h_{02} \quad \cdots \quad -h_{0(N-1)}\,)^T]. \tag{3.16}$$

The matrix $\mathbf{G}_0$ defines an injective homomorphism from $\mathcal{Z}_M^{N-1}$ to $\mathcal{Z}_M^N$ denoted by

$$G_0 \colon \mathcal{Z}_M^{N-1} \xrightarrow{\mathbf{G}_0} \mathcal{Z}_M^N. \tag{3.17}$$

The image of G_0, $\mathrm{im}(G_0)$, is the code generated by $\mathbf{G}_0$, C_0. Let L'_0 be the set of vectors from L_0 which also belong to $\mathrm{im}(\mathbf{G}_0)$, and let L_1 be the set of vectors from $\mathcal{Z}_M^{N-1}$ which

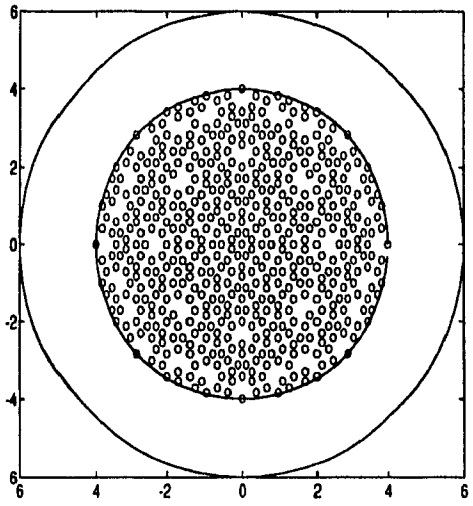

Figure 3.2: Possible correlation values for a coded system in $\mathcal{Z}_8^6$

are mapped by $\mathbf{G}_0$ into L'_0. It is clear that $|L_1| < |L_0|$ because of the construction of $\mathbf{G}_0$. We can now repeat this procedure again with respect to the list(set) L_1. We choose an $(N-1)$–dimensional vector $\mathbf{h}_1 = (\, h_{11} \quad \cdots \quad h_{1(N-2)} \quad 1\,)$ such that as many vectors as possible from L_1 are excluded from its kernel. Let us consider the generator matrix associated with $\mathbf{h}_1$, defined by

$$((-h_{11} \quad \cdots \quad -h_{1(N-2)})^T \quad | \quad \mathbf{I}_{N-2}) \tag{3.18}$$

and its associated injective homomorphism

$$G_1 : \mathcal{Z}_M^{N-2} \xrightarrow{\mathbf{G}_1} \mathcal{Z}_M^{N-1} \tag{3.19}$$

defined by

$$\mathbf{c}_i = \mathbf{x}_i \mathbf{G}_1. \tag{3.20}$$

The code $\mathcal{C}_1$ is defined as the image of the composition

$$G_0\, G_1 : \mathcal{Z}_M^{N-2} \xrightarrow{\mathbf{G}_1} \mathcal{Z}_M^{N-1} \xrightarrow{\mathbf{G}_0} \mathcal{Z}_M^{N}. \tag{3.21}$$

Now let L'_1 be the set of vectors from L_1 which also belong to $\mathrm{im}(G_1)$. It is clear that $|L'_1| < |L_1|$. We may therefore continue this procedure until we have excluded all the

required vectors. The procedure ends since $|L_0| > |L_1| > |L_2| \cdots$ (at each stage we must get rid of at least one vector from the list), and $|L_0| < \infty$. The maximum number of iterations is $|L_0|$, but the actual number of iterations is far less due to the fact that far more than one vector is eliminated at each stage. Assuming that we require $N-K$ steps to remove all the vectors, we are left with an (N, K) code having a generator matrix $\mathbf{G}$ given by

$$\mathbf{G} = \mathbf{G}_{N-K-1}\mathbf{G}_{N-K-2}\cdots\mathbf{G}_1\mathbf{G}_0, \tag{3.22}$$

where $\mathbf{G}_i$ is a $(N-1-i) \times (N-i)$ matrix. It should be noted that the performance (ie. the minimum-weight codeword) after each step does not necessarily improve, since we are simply trying to remove as many vectors as possible at each step. After all the vectors are removed, however, we are guaranteed an improvement in performance that is dictated by the weight of the codeword(s) lying just outside L_0. The factorization in (3.22) may be viewed as a concatenated coding system which uses K coding stages rather than the conventional two-stage approach [15].

It should be pointed out that the construction method can be modified somewhat to allow for more efficient searching methods. Provided that all the $\mathbf{h}$-vectors and corresponding generator matrices are chosen in systematic form, we may search for the $\mathbf{h}$-vectors in any order. For instance, suppose that we wanted to create a code with specific N and K. We could begin searching for each $\mathbf{h}$-vector in the order $\mathbf{h}_0, \mathbf{h}_1, \cdots, \mathbf{h}_{N-K-1}$, as dictated by the construction method with the hope that at the end of the search, we have deleted all the undesired vectors. We may also search in the reverse sense, beginning with $\mathbf{h}_{N-K-1}$. Starting from the initial list L_0, we create a new list, L_0', which is contains all the distinct length $K+1$ vectors coming from the first $K+1$ positions of all the vectors in L_0. We then choose $\mathbf{h}_{N-K-1}$ such that as many of the vectors in L_0' lie outside its kernel. For each vector, $\mathbf{x}$, which is deleted from L_0' we delete those vectors from L_0 whose first $K+1$ positions are identical to $\mathbf{x}$. In general, the number of vectors deleted from L_0 will be greater than those deleted from L_0'. The vectors from L_0 which remain undeleted make up the list L_1. We repeat the same procedure on the list L_1

with the vector $\mathbf{h}_{N-K-2}$, and continue to repeat it until we reach $\mathbf{h}_0$, hopefully having removed all the undesired vectors along the way. In the end, the factorization of the code's generator matrix will, of course, be in the same form as in (3.22). This alternate view of looking at the code construction method will be used to find some of the codes in this chapter, as well as to search for codes tailored specially for the decoding strategy presented in the next chapter.

3.3.2 Selection of the first vector, $\mathbf{h}_0$

Let us consider codes with only one parity symbol. We need only consider, therefore, the construction of $\mathbf{h}_0$, whose main function will be to remove phase ambiguities, which are catastrophic in any non-coherent system. Phase ambiguities are caused by the vectors $(\alpha \quad \alpha \quad \cdots \quad \alpha)$, where $\alpha \in \mathcal{Z}_M, \alpha \neq 0$. It should be clear that if these vectors belong to $\mathcal{C}$ there would be no way of distinguishing any codeword $\mathbf{c}$ from $\mathbf{c} + (\alpha \quad \alpha \quad \cdots \quad \alpha)$ since $|\rho_m|$ is invariant to any phase shift α. For phase vectors having equal components we have $|\rho_m| = 1$, and therefore the function of $\mathbf{h}_0$ is to exclude the vectors having $d^2_{NC} = 0$.

Theorem 1 *In order for the vectors* $(\alpha \quad \alpha \quad \cdots \quad \alpha)$ *to be outside the kernel of* $\mathbf{h}_0$, *where* $\alpha \in \mathcal{Z}_M, \alpha \neq 0$, *the sum of the components of* $\mathbf{h}_0$ *must be an invertible element in* $\mathcal{Z}_M$.

Proof. Let $\lambda = \sum_{i=0}^{N-1} h_{0_i}$ and $\mathbf{v}$ be any of the vectors that cause phase ambiguities which are to lie outside the kernel of $\mathbf{h}_0$. We have, therefore, that

$$\mathbf{v}\mathbf{h}_0^T = \alpha\lambda. \tag{3.23}$$

In order for this expression not to be zero, for all α ($\alpha \neq 0$), which assures that $\mathbf{v}$ lies outside the kernel of $\mathbf{h}_0$, we require that λ be an invertible element. The requirement,

therefore, to assure that phase ambiguities are removed from the code is given by

$$\lambda = \sum_{i=0}^{N-1} h_{0i} = \text{any invertible element in } Z_M \quad Q.E.D. \tag{3.24}$$

We will now show that it is not possible to remove all of the next worst case vectors with $\mathbf{h}_0$. From the definition for $|\rho_m|$ in (2.18) it should be clear that the next worst case vectors are those which contain $N-1$ identical symbols, α, and one of the two symbols nearest to α, either $\alpha+1$ or $\alpha-1$. This is depicted in Figure 3.3, where we show ρ_m composed of $N-1$ phasors pointing in the same direction (corresponding to the phase α) and one pointing upwards or downwards in the next closest direction (corresponding to $\alpha+1$ or $\alpha-1$). The angle $\left(\frac{2\pi}{M}\right)$ corresponds to the symbols 1 and $M-1(-1)$ in Z_M, and therefore the next to worst case vectors have correlation magnitude

$$|\rho_m| = \left|\frac{N-1}{N} + \frac{1}{N}\exp\left(j\frac{2\pi}{M}\right)\right|. \tag{3.25}$$

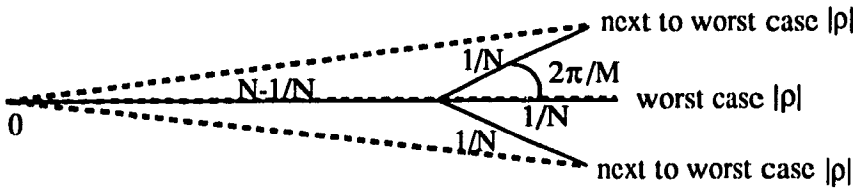

Figure 3.3: Worst Case and next to worst case correlation values

We have, therefore, that phase vectors of the form

$$(\alpha \quad \cdots \quad \alpha \quad \alpha+1 \quad \alpha \quad \cdots \quad \alpha) \tag{3.26}$$

and

$$(\alpha \quad \cdots \quad \alpha \quad \alpha-1 \quad \alpha \quad \cdots \quad \alpha) \tag{3.27}$$

are the sources of the largest correlations $|\rho_m| < 1$. Letting $\mathbf{h}_0$ act on the first of these two vectors yields

$$(\alpha \quad \cdots \quad \alpha+1 \quad \alpha \quad \cdots \quad \alpha)\,\mathbf{h}_0^T = \lambda\alpha + h_{0i}, \tag{3.28}$$

where i is the position of $\alpha + 1$ in the vector. Since $\lambda\alpha$ is a distinct element in $\mathcal{Z}_M$ for every α because λ is not zero divisor, there will be exactly one α satisfying

$$\lambda\alpha + h_{0i} = 0. \tag{3.29}$$

Using this argument for each component of $\mathbf{h}_0$ implies that N such vectors must remain after applying $\mathbf{h}_0$. The situation is identical, of course, for the second vector.

We may therefore conclude that $\mathbf{h}_0$ may be chosen arbitrarily as long as (3.24) is satisfied and that its function is to remove phase ambiguities. This being the case, let us choose

$$\mathbf{h}_0 = (1 \quad 0 \quad \cdots \quad 0). \tag{3.30}$$

This amounts to choosing as codewords, all the vectors from $\mathcal{Z}_M^N$ beginning with a zero. One of the corresponding generator matrices has the form

$$\mathbf{G}_0 = (\mathbf{0}_{N-1}{}^T \quad | \quad \mathbf{I}_{N-1}). \tag{3.31}$$

Another equivalent generator matrix will be considered shortly. The generator matrix $\mathbf{G}_0$ in (3.31) corresponds to a phase ambiguity removing code which is the most inner stage in (3.22).

3.3.3 Code Rate Improvement and Differential Encoding

The choice for $\mathbf{h}_0$ in (3.30) is particularly interesting since it allows an improvement in the rate of the code, as we shall now demonstrate. First of all, since all phase ambiguities are removed by applying $\mathbf{h}_0$, we may choose to transmit a codeword from $\mathcal{C}$, or equivalently, from any of its cosets $\mathcal{C} + (\alpha \quad \alpha \quad \cdots \quad \alpha)$, $\forall\alpha \in \mathcal{Z}_M$, since $|\rho_m|$ is invariant over these cosets. Therefore, if we wish to transmit a certain codeword $\mathbf{c}_i$, we may equivalently transmit $\mathbf{c}_i' = \mathbf{c}_i + c'_{j(N-1)}(1 \quad 1 \quad \cdots \quad 1)$, where $c'_{j(N-1)}$ is the last symbol of the previous codeword, $\mathbf{c}_j'$. With the aforementioned choice for $\mathbf{h}_0$, it is clear that the last symbol of the previous codeword will be identical to the first symbol of

the next codeword, and clearly only one need be transmitted. This overlapping effect therefore increases the rate of the code by $N/(N-1)$.

The other interesting point that arises due to this choice for $\mathbf{h}_0$ combined with the overlapping effect is that this scheme is equivalent to differential encoding. The differential encoding process may be viewed as first multiplying an N-dimensional information vector by a generator matrix of the form

$$\mathbf{G}_d = \begin{pmatrix} 0 & 1 & 1 & \cdots & 1 \\ 0 & 0 & 1 & \cdots & 1 \\ \vdots & \vdots & \vdots & \ddots & 1 \\ 0 & 0 & 0 & \cdots & 1 \end{pmatrix} \tag{3.32}$$

to yield an $N+1$-dimensional codeword. Every symbol of the codeword is then incremented by the value of the last symbol of the previous codeword, then the codeword is overlapped with the previous codeword. These two operations, multiplication by $\mathbf{G}_d$ and codeword overlapping, are equivalent to regular differential encoding. Clearly, $\mathbf{G}_d$ is row-wise equivalent to the matrix in (3.31), and thus the two schemes are equivalent. Therefore, using $\mathbf{h}_0$ alone, we have codes that do not expand bandwidth (ie. $R_c = 1$) with respect to M-PSK, remove phase ambiguities but suffer some degradation with respect to coherent detection. The performance of these codes is equivalent to the non-coherent detection schemes of [1]-[3] where it is shown that the SNR degradation with respect to coherent M-PSK is reduced by increasing the codeword length N.

Using codeword overlapping, we have that the i^{th} transmitted block, $\mathbf{b_i}$, including the last symbol of the previous block, is given by,

$$\mathbf{b_i} = [\, b_{i-1,N} \quad \mathbf{x_i} \,]\, \mathbf{G} \tag{3.33}$$

where $b_{i-1,N}$ is the last symbol of the previously transmitted block, $\mathbf{x_i}$ is the information

vector of the i^{th} codeword, and $\mathbf{G}$ is a $(K+1)\times(N+1)$ matrix given by

$$\mathbf{G}=\begin{bmatrix}0 & 0 & \cdots & 0\\ & \mathbf{G_c} & & \end{bmatrix}\mathbf{G}_0+\underbrace{\begin{bmatrix}1 & 1 & \cdots & 1\\ & \mathbf{0}_{N\times K} & & \end{bmatrix}}_{\mathbf{G}_{over}}=\left(\begin{array}{c|ccc}1 & 1 & 1 & \cdots\ 1\\ \hline 0 & & & \\ 0 & & \mathbf{G_c} & \\ \vdots & & & \\ 0 & & & \end{array}\right). \quad (3.34)$$

The $K\times N$ matrix $\mathbf{G_c}$ is the generator matrix of what we call the *outer code*, that is the one corresponding to the parity-check vectors, $\mathbf{h}_1,\cdots,\mathbf{h}_{N-K}$, which is responsible for achieving coding gain over an uncoded system. The matrix $\mathbf{G}_0$ is a $N\times(N+1)$ generator matrix of a phase ambiguity removing code similar to (3.31), and the $N\times(N+1)$ matrix $\mathbf{G}_{over}$ reflects the overlapping operation.

Although it is not necessary to employ this codeword overlapping technique in order to design codes for non-coherent detection, we maintain that by doing so, the resulting codes will perform at least as well, if not better, than codes that do not use it. To see this, let us assume that we have two (N,K) codes $\mathcal{C}_1$ and $\mathcal{C}_2$. In addition, with $\mathcal{C}_2$ we will employ codeword overlapping. This means that for $\mathcal{C}_2$, the observation interval will be $N+1$ symbols, whereas the one with $\mathcal{C}_1$ will be N symbols. Codeword overlapping assures that each codeword begins with a zero, as this is the function of $\mathbf{h}_0$. This makes the complex correlations of the remaining vectors the same as that of a scheme of length N, except that they are all shifted to the right by one unit, as shown in Fig.3.4(a). In order for a code to have a given d^2_{NC}, we must remove all the vectors with complex correlations lying in the annulus bounded by the circles of radii $N+1$ and $N+1-d^2_{NC}$ for $\mathcal{C}_2$ and N and $N-d^2_{NC}$ for $\mathcal{C}_1$ as shown in Fig.3.4(b). Clearly, far fewer of the vectors need be excluded for the code which uses codeword overlapping which makes the chances of designing more powerful codes much better. What this really implies is that the outer code for a system which employs codeword overlapping has to do much less work to remove the unwanted vectors. Consequently, from this point onward we will assume that codeword overlapping is performed, and we will concentrate only on

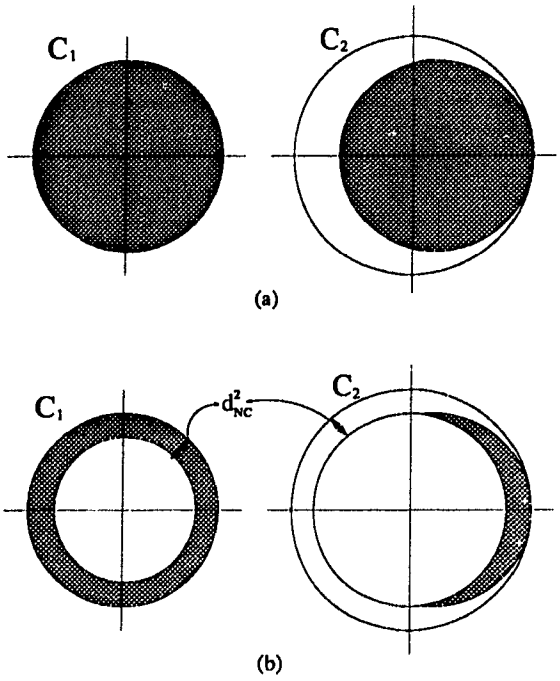

Figure 3.4: Advantage of codeword overlapping

the design of the outer code.

3.3.4 Design of $\mathbf{h}_1$ to Yield Coding Gain

We now consider $\mathbf{h}_1$ in order to devise some codes which exhibit minimal bandwidth expansion ($R_c = (N-1)/N$) but achieve coding gain with respect to coherent M–PSK. The form of $\mathbf{h}_1$ is somewhat dependent on the desired codeword length, and therefore as an example we will consider building a code with $N = 9$ in $\mathcal{Z}_8$, so that the rate of the code is 8/9. After applying $\mathbf{h}_0$ the remaining worst vectors are shown in Table 3.1, which corresponds to the list L_0' as previously defined (note that the vectors are actually of length $N + 1 = 10$ since we perform codeword overlapping.)

Recalling that $\mathbf{h}_1$ acts on the module $\mathcal{Z}_M^N$, we must focus our attention on the symbols in positions 1 through 9 which make up the vector from $\mathcal{Z}_M^N$. The collection of vectors composed of the symbols from these positions correspond to the list L_1. In

order for as many vectors from L_1 as possible to lie outside the kernel of $\mathbf{h}_1$, we have the following constraints on the form of $\mathbf{h}_1$:

1. $\sum_{i=0}^{8} h_{1i} \neq 0$
2. $h_{1i} \neq 0, 0 \leq i \leq 8$
3. if $h_{1i} = \alpha, h_{1j} \neq -\alpha \;\; 0 \leq j \leq 8$
4. no sum of any 8 h_{1i} can be zero.

Each of the four constraints on $\mathbf{h}_1$ applies to a particular set of vectors from Table 3.1, and assures that none in that corresponding set belong to its kernel. The first constraint applies to the vectors in set 1 and is essentially the same as the condition imposed on the form for $\mathbf{h}_0$ (3.24) since the two vectors are strings of one symbol; however, the requirement for the sum is that it not be zero (the strings are made up only of 1's or 7's). The second constraint, for the vectors in set 2, is required since the vectors are permutations of the vectors having only a single non–zero component (1 or 7) and therefore $\mathbf{h}_1$ may not have a zero in any position. The third constraint simply requires that no two components of $\mathbf{h}_1$ may be additive inverses of each other, since set 3 contains all permutations of the vectors with only two non–zero components which are identical (1 or 7). The fourth constraint similar to the first since the vectors in set 4 are permutations of the vectors with a single 0 and eight 1's or 7's. The last set of vectors (set no. 5) in the table cannot be entirely removed with $\mathbf{h}_1$ since it would require that no symbol in $\mathbf{h}_1$ is repeated, which is impossible for a vector in $\mathcal{Z}_8^9$. One possible vector which satisfies all four constraints is

$$\mathbf{h}_1 = (1 \quad 1 \quad \cdots \quad 1 \quad 2 \quad 3) \tag{3.35}$$

and results in a code with $d_{NC}^2 = .586$. In establishing the performance of this code with respect to coherent 8–PSK, we use the relation in (3.14) which yields a coding gain of

$$G_c = 10 \log \left[\frac{8}{9} .586/(1 - \cos \frac{\pi}{4}) \right] = 2.50 \text{dB}. \tag{3.36}$$

Set no	Vector										ρ_m
1	0	1	1	1	1	1	1	1	1	1	9733
	0	7	7	7	7	7	7	7	7	7	9733
2	0	7	0	0	0	0	0	0	0	0	9733
	0	0	7	0	0	0	0	0	0	0	9733
											9733
	0	0	0	0	0	0	0	0	7	0	9733
	0	0	0	0	0	0	0	0	0	7	9733
	0	1	0	0	0	0	0	0	0	0	9733
	0	0	1	0	0	0	0	0	0	0	9733
											9733
	0	0	0	0	0	0	0	0	1	0	9733
	0	0	0	0	0	0	0	0	0	1	9733
3	0	1	1	0	0	0	0	0	0	0	9520
	0	1	0	1	0	0	0	0	0	0	9520
											9520
	0	0	0	0	0	0	0	1	0	1	9520
	0	0	0	0	0	0	0	0	1	1	9520
	0	7	7	0	0	0	0	0	0	0	9520
	0	7	0	7	0	0	0	0	0	0	9520
											9520
	0	0	0	0	0	0	0	7	0	7	9520
	0	0	0	0	0	0	0	0	7	7	9520
4	0	0	1	1	1	1	1	1	1	1	9520
	0	1	0	1	1	1	1	1	1	1	9520
											9520
	0	1	1	1	1	1	1	1	0	1	9520
	0	1	1	1	1	1	1	1	1	0	9520
	0	0	7	7	7	7	7	7	7	7	9520
	0	7	0	7	7	7	7	7	7	7	9520
											9520
	0	7	7	7	7	7	7	7	0	7	9520
	0	7	7	7	7	7	7	7	7	0	9520
5	0	1	7	0	0	0	0	0	0	0	9414
	0	1	0	7	0	0	0	0	0	0	9414
											9414
	0	0	0	0	0	0	0	7	0	1	9414
	0	0	0	0	0	0	0	0	7	1	9411

Table 3.1: Worst Vectors after applying $\mathbf{h}_0$

3.4 Results from code searches

In order to find codes with more parity symbols which achieve higher coding gains over uncoded coherent M-PSK, a computer program was devised to carry out the previously outlined procedure. Details on the computer searches can be found in Appendix A.

Before discussing the results of the searches, we present a general union bound on the bit error probability for these codes. Since the codes are linear, each transmitted codeword has the same probability of error associated with it, and we may consider only the zero codeword as being transmitted. We have, therefore, that the probability of

codeword error is bounded by

$$P_{codeword} \leq \sum_{m=1}^{|\mathcal{C}|-1} P(\mathbf{c}_0 \rightarrow \mathbf{c}_m). \tag{3.37}$$

If we retain only the dominant terms in (3.37), which correspond to the codewords having minimum non–coherent distance, and use the actual pairwise error probability in (2.24), we obtain the following bound on $P_{codeword}$ for high SNR

$$P_{codeword} \leq N_n P(\mathbf{c}_0 \rightarrow \mathbf{c}_{neighbour}) \tag{3.38}$$

The codeword $\mathbf{c}_{neighbour}$ is one which has a distance d^2_{NC}, a so-called *nearest–neighbour*, and the factor N_n is the number of such codewords. We would like to express $P_{codeword}$ in terms of γ_b, the signal-to–noise ratio per transmitted bit. Since there are $K \log_2 M$ information bits in the N transmitted symbols, we have that

$$\gamma = \frac{K}{N} (\log_2 M) \gamma_b. \tag{3.39}$$

In order to obtain an expression for the bit error probability, P_b, we determine the average number of bit errors per codeword among the set of codewords with minimum d^2_{NC}, $\overline{e_b}$, so that

$$P_b = \frac{\overline{e_b}}{K \log_2 M} P_{codeword}. \tag{3.40}$$

In determining $\overline{e_b}$, we will assume that the bit representations of the symbols in $\mathcal{Z}_M$ are *Gray coded.* This assures that adjacent symbols differ only in one bit position, which should make $\overline{e_b}$ smaller. Gray coding for $\mathcal{Z}_4$ and $\mathcal{Z}_8$ are shown in Fig. 3.5.

Tables 3.2 and 3.3 present the asymptotic coding gain over uncoded coherent M–PSK for bandwidth–expanding codes in $\mathcal{Z}_4$ and $\mathcal{Z}_8$ that were found. The number of nearest neighbours, N_b, and the average number of bits in error per block, $\overline{e_b}$, are also shown. Upon first glance, it may seem that some of these codes achieve only marginal coding gain. We must note, however, that these gains are expressed with respect to coherent detection which is a rather ambitious reference. An additional 2.7dB for 8–PSK and 2.3dB for QPSK must be added to realize the performance improvement over

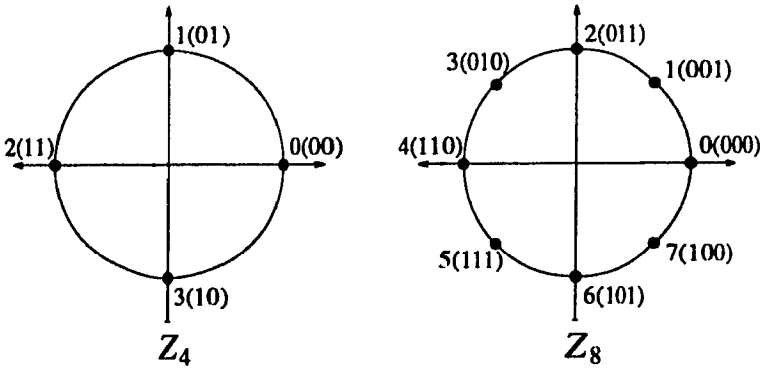

Figure 3.5: Gray coding for $\mathcal{Z}_4$ and $\mathcal{Z}_8$

an uncoded scheme with differentially–coherent detection. For each code length N, the values of K were chosen so that the resulting code is more bandwidth efficient than the next lower order uncoded modulation format (ie. the QPSK codes all convey more information than uncoded BPSK, and the 8–PSK all convey more information than uncoded QPSK.) We should also note that many of the codes with one parity symbol, the $(N, N-1)$ codes, are simply parity–check codes in $\mathcal{Z}_M$. These codes are characterized by a parity check matrix

$$\mathbf{H} = (1 \quad 1 \quad \cdots \quad 1), \tag{3.41}$$

which means that the components of every valid codeword must sum to zero modulo–M, which is a parity–check in $\mathcal{Z}_M$.

We now present the bit–error rate performance of several bandwidth–expanding codes in $\mathcal{Z}_4$ and $\mathcal{Z}_8$. We have found that many of the codes have weight distributions with a significant number of codewords near the minimum distance, and therefore a nearest neighbour approximation to the union–bound is not sufficient even at fairly high SNR. The following curves all use several of the minimum weights to more accurately describe the bit–error rate.

In Fig.3.6 we present the probability of bit error for three codes in $\mathcal{Z}_4$, each with a single parity check bit. All three codes have a fairly large number of nearest neighbours, which results in a performance which only reaches asymptotic values for very

N	K	d^2_{NC}	N_n	$\overline{\epsilon_b}$	Gain (Coh. QPSK)	h-vectors
3	2	2.00	12	2.000	1.25 dB	$\mathbf{h}_1 = (1\ 1\ 1)$
6	5	2.00	61	2.623	2.22 dB	$\mathbf{h}_1 = (1\ 1\ 1\ 1\ 1\ 1)$
7	6	2.00	56	3.000	2.34 dB	$\mathbf{h}_1 = (1\ 1\ 1\ 1\ 1\ 1\ 1)$
7	4	3.53	44	3.364	3.04 dB	$\mathbf{h}_1 = (3\ 2\ 1\ 2\ 0\ 0\ 1)$ $\mathbf{h}_2 = (3\ 2\ 1\ 0\ 0\ 1)$ $\mathbf{h}_3 = (3\ 1\ 1\ 1\ 1)$
9	5	4.00	17	2.118	3.47 dB	$\mathbf{h}_1 = (3\ 0\ 1\ 2\ 2\ 0\ 0\ 0\ 1)$ $\mathbf{h}_2 = (1\ 3\ 2\ 0\ 0\ 0\ 0\ 1)$ $\mathbf{h}_3 = (3\ 1\ 1\ 0\ 1\ 0\ 1)$ $\mathbf{h}_4 = (1\ 1\ 1\ 1\ 0\ 1)$
10	9	2.00	91	1.978	2.55 dB	$\mathbf{h}_1 = (1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1)$
10	7	2.94	36	3.556	3.13 dB	$\mathbf{h}_1 = (1\ 0\ 1\ 0\ 2\ 1\ 1\ 0\ 0\ 1)$ $\mathbf{h}_2 = (2\ 2\ 1\ 1\ 0\ 0\ 2\ 0\ 1)$ $\mathbf{h}_3 = (1\ 1\ 1\ 1\ 1\ 1\ 0\ 1)$
10	6	3.72	44	4.227	3.49 dB	$\mathbf{h}_1 = (3\ 1\ 3\ 2\ 3\ 2\ 3\ 1\ 3\ 1)$ $\mathbf{h}_2 = (1\ 1\ 1\ 3\ 1\ 1\ 3\ 1\ 1)$ $\mathbf{h}_3 = (3\ 3\ 1\ 2\ 2\ 3\ 2\ 1)$ $\mathbf{h}_4 = (3\ 3\ 3\ 3\ 3\ 1\ \ 1)$
11	10	2.00	132	3.333	2.60 dB	$\mathbf{h}_1 = (1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1)$
11	8	2.95	38	2.842	3.31 dB	$\mathbf{h}_1 = (0\ 1\ 0\ 1\ 0\ 1\ 1\ 1\ 0\ 0\ 1)$ $\mathbf{h}_2 = (3\ 2\ 2\ 1\ 1\ 0\ 2\ 1\ 0\ 1)$ $\mathbf{h}_3 = (1\ 1\ 1\ 1\ 1\ 1\ 0\ 0\ 1)$
11	7	3.52	6	3.667	3.50 dB	$\mathbf{h}_1 = (0\ 0\ 0\ 0\ 2\ 0\ 3\ 3\ 3\ 2\ 1)$ $\mathbf{h}_2 = (2\ 2\ 0\ 2\ 0\ 0\ 3\ 1\ 0\ 1)$ $\mathbf{h}_3 = (1\ 0\ 1\ 3\ 1\ 1\ 0\ 3\ 1)$ $\mathbf{h}_4 = (1\ 1\ 3\ 1\ 1\ 1\ 3\ 1)$
11	6	4.38	14	4.286	3.79 dB	$\mathbf{h}_1 = (2\ 2\ 1\ 1\ 0\ 1\ 0\ 3\ 2\ 1\ 1)$ $\mathbf{h}_2 = (2\ 3\ 0\ 0\ 0\ 1\ 2\ 2\ 2\ 1)$ $\mathbf{h}_3 = (0\ 0\ 3\ 2\ 1\ 3\ 1\ 2\ 1)$ $\mathbf{h}_4 = (3\ 0\ 2\ 3\ 1\ 0\ 0\ 1)$ $\mathbf{h}_5 = (1\ 1\ 1\ 1\ 1\ 3\ 1)$

Table 3.2: Bandwidth Expanding codes in $\mathcal{Z}_4$

high SNR. Even at $P_b = 10^{-7}$ all three have reached only slightly more than half their asymptotic coding gain. Significant improvement, however, is still achieved compared with differentially–coherent detection. In Fig.3.7 the performance of four codes with larger d^2_{NC} is shown. It can be seen that three of the four codes achieve their asymptotic performance much more rapidly than was the case with the simple codes, because of the smaller number of nearest neighbours. Figs. 3.8 and 3.9 present similar results for codes in $\mathcal{Z}_8$. With these codes we must be a little more careful since the more powerful ones approach the bandwidth efficiency of QPSK, that is to say their throughput is only slightly higher than uncoded QPSK. Although they achieve significant coding gain over uncoded coherent 8–PSK, they have an error performance quite close to uncoded coherent QPSK because of the approximate 3.6dB gap between coherent QPSK and 8–PSK. Of course, comparing these codes in $\mathcal{Z}_8$ with both differentially–coherent QPSK

N	K	d^2_{NC}	N_n	$\overline{e_b}$	Gain Coh 8-PSK	h-vectors
5	4	.586	20	1.600	2.04 dB	$\mathbf{h}_1 = (1\ 1\ 1\ 1\ 1)$
6	5	.586	30	1.667	2.22 dB	$\mathbf{h}_1 = (1\ 1\ 1\ 1\ 1\ 1)$
7	6	.586	56	3.000	2.34 dB	$\mathbf{h}_1 = (1\ 1\ 1\ 1\ 1\ 1\ 1)$
7	5	.844	12	2.667	3.13 dB	$\mathbf{h}_1 = (0\ 3\ 2\ 1\ 2\ 0\ 1)$ $\mathbf{h}_2 = (2\ 1\ 1\ 1\ 0\ 1)$
8	7	.586	42	2.000	2.43 dB	$\mathbf{h}_1 = (3\ 1\ 1\ 1\ 1\ 1\ 1\ 1)$
8	6	.848	16	2.750	3.34 dB	$\mathbf{h}_1 = (0\ 3\ 2\ 1\ 3\ 2\ 0\ 1)$ $\mathbf{h}_2 = (2\ 1\ 1\ 1\ 0\ 0\ 1)$
9	8	.586	42	2.000	2.50dB	$\mathbf{h}_1 = (3\ 2\ 1\ 1\ 1\ 1\ 1\ 1\ 1)$
9	7	.851	32	3.062	3.54dB	$\mathbf{h}_1 = (5\ 4\ 3\ 2\ 1\ 3\ 2\ 0\ 1)$ $\mathbf{h}_2 = (1\ 1\ 1\ 1\ 1\ 0\ 0\ 1)$
10	9	.586	90	1.800	2.55dB	$\mathbf{h}_1 = (1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1)$
10	8	.773	10	5.200	3.25dB	$\mathbf{h}_1 = (2\ 1\ 0\ 1\ 0\ 3\ 2\ 1\ 0\ 1)$ $\mathbf{h}_2 = (5\ 5\ 5\ 4\ 4\ 1\ 1\ 1\ 1)$
10	7	1.07	18	2.778	4.06dB	$\mathbf{h}_1 = (0\ 0\ 0\ 0\ 2\ 0\ 2\ 0\ 0\ 1)$ $\mathbf{h}_2 = (3\ 6\ 4\ 2\ 5\ 3\ 1\ 0\ 1)$ $\mathbf{h}_2 = (2\ 1\ 1\ 1\ 0\ 0\ 0\ 1)$

Table 3.3: Bandwidth Expanding codes in $\mathcal{Z}_8$

and 8–PSK still results in significant performance improvements.

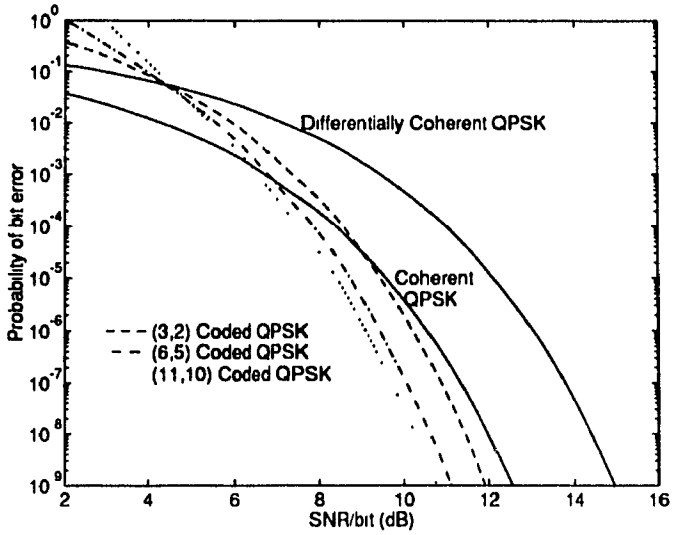

Figure 3.6: Performance of some simple QPSK codes

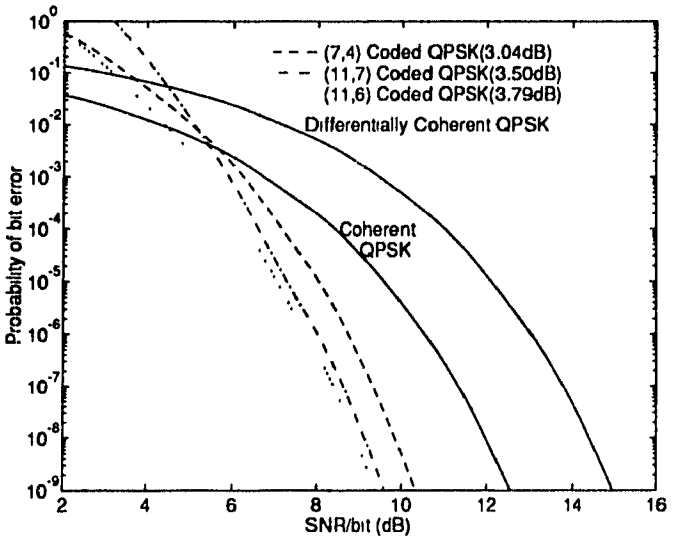

Figure 3.7: Performance of some more powerful QPSK codes

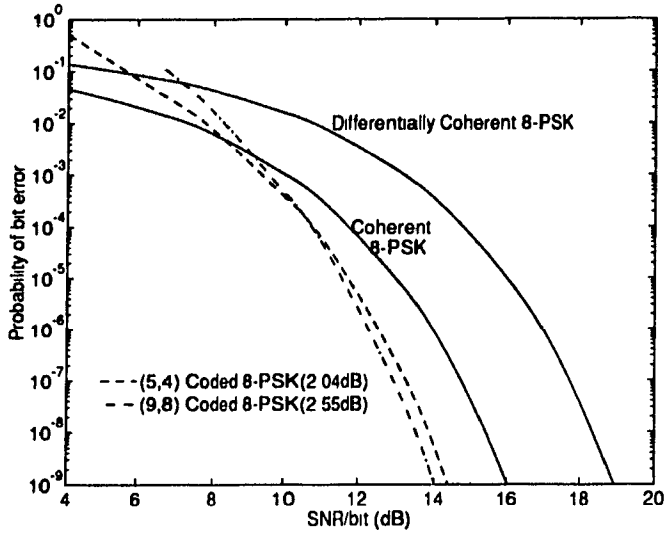

Figure 3.8: Performance of some simple 8–PSK codes

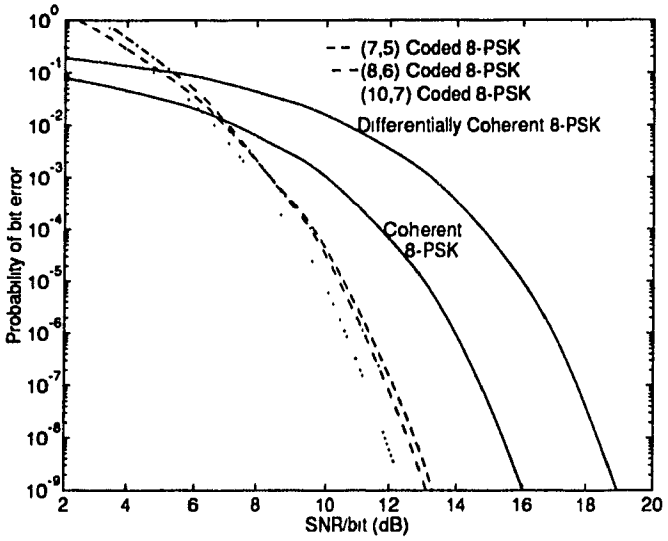

Figure 3.9: Performance of some more powerful 8–PSK codes

3.5 Coding Without Bandwidth Expansion

In this section we present some codes which do not expand bandwidth with respect to some uncoded reference scheme. Let us assume that we are using a module-phase code generated by the $(K \times N)$ matrix $\mathbf{G}$ which can be represented by the homomorphism $G: \mathcal{Z}_M^K \xrightarrow{\mathbf{G}} \mathcal{Z}_M^N$. We will also assume that symbols are transmitted at a rate R[symbols/sec]. The information per transmitted symbol is $(K/N)\log_2 M$[bits/symbol]. We have, therefore, that the information rate, R_I is related to the symbol rate, R, via: $R_I = R(K/N)\log_2 M$[bits/sec]. Suppose that we have another code $\mathbf{G}'$: $\mathcal{Z}_{M'}^{K'} \xrightarrow{G'} \mathcal{Z}_{M'}^{N'}$ such that $R_I' = R'(K'/N')\log_2 M'$. If the information rates of the two coded systems are to be the same, we require that

$$R(K/N)\log_2 M = R'(K'/N')\log_2 M'. \tag{3.42}$$

If our reference scheme $\mathbf{G}'$ is an uncoded scheme, then $K'/N' = 1$ and $R(K/N)\log_2 M = R'\log_2 M'$. In the previous section, the codes were compared with uncoded schemes with the same M-PSK constellation. Thus, $M = M'$ and the bandwidth expansion was

$$R/R' = N/K. \tag{3.43}$$

Since we do not want to expand bandwidth, we are forced to impose the constraint that $R = R'$. This shows that we must have a constellation expansion since $\log_2 M/\log_2 M' = N/K$. The code rate (K/N) must therefore satisfy

$$K/N = \log_2 M'/\log_2 M \tag{3.44}$$

in order not to expand bandwidth with respect to uncoded M'-PSK.

Table 3.4 lists some bandwidth efficient codes and their performance with respect to some uncoded coherent scheme. All were found by computer search, the details of which can be found in Appendix A. Some of BPSK equivalent codes (ie. codes which do not expand bandwidth with respect to uncoded BPSK), have been found by exhaustive

N	K	Ring	d^2_{NC}	N_n	$\overline{c_b}$	Gain	$\mathbf{G_C}$ or h-vectors
6*	3	Z_4	3.396	14	2.429	2.30dB(BPSK)	$\mathbf{G_C} = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 2 \\ 0 & 1 & 0 & 0 & 2 & 1 \\ 0 & 0 & 1 & 2 & 0 & 1 \end{pmatrix}$
8	4	Z_4	4.000	28	3.000	3.01dB(BPSK)	$\mathbf{h}_1 = (1\ 1\ 1\ 2\ 0\ 0\ 0\ 1)$ $\mathbf{h}_2 = (3\ 1\ 1\ 0\ 0\ 0\ 1)$ $\mathbf{h}_3 = (1\ 1\ 2\ 0\ 0\ 1)$ $\mathbf{h}_4 = (3\ 3\ 1\ 1\ 1)$
10	5	Z_4	4.597	18	3.778	3.61dB(BPSK)	$\mathbf{h}_1 = (2\ 1\ 1\ 0\ 3\ 3\ 2\ 0\ 1\ 1)$ $\mathbf{h}_2 = (0\ 3\ 3\ 0\ 1\ 2\ 0\ 1\ 1)$ $\mathbf{h}_3 = (2\ 0\ 3\ 2\ 0\ 0\ 1\ 1)$ $\mathbf{h}_4 = (1\ 0\ 0\ 2\ 1\ 2\ 1)$ $\mathbf{h}_5 = (2\ 1\ 1\ 1\ 2\ 1)$
12	6	Z_4	4.938	26	3.154	3.93dB(BPSK)	$\mathbf{h}_1 = (0\ 0\ 0\ 0\ 0\ 2\ 0\ 0\ 0\ 0\ 0\ 1)$ $\mathbf{h}_2 = (2\ 3\ 1\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 1)$ $\mathbf{h}_3 = (1\ 3\ 1\ 3\ 1\ 0\ 0\ 0\ 0\ 1)$ $\mathbf{h}_4 = (3\ 1\ 0\ 2\ 2\ 0\ 0\ 0\ 1)$ $\mathbf{h}_5 = (3\ 2\ 2\ 1\ 0\ 0\ 0\ 1)$ $\mathbf{h}_6 = (1\ 1\ 1\ 1\ 1\ 1\ 1)$
14	7	Z_4	5.566	8	5.750	4.45dB(BPSK)	$\mathbf{h}_1 = (1\ 3\ 0\ 2\ 0\ 1\ 1\ 1\ 0\ 3\ 2\ 1\ 0\ 1)$ $\mathbf{h}_2 = (2\ 0\ 2\ 2\ 0\ 1\ 0\ 0\ 0\ 3\ 0\ 2\ 1)$ $\mathbf{h}_3 = (1\ 2\ 3\ 1\ 0\ 3\ 2\ 1\ 1\ 0\ 1\ 1)$ $\mathbf{h}_4 = (0\ 0\ 0\ 0\ 0\ 2\ 0\ 1\ 1\ 0\ 1\ 1)$ $\mathbf{h}_5 = (2\ 3\ 2\ 1\ 2\ 0\ 3\ 3\ 1\ 1)$ $\mathbf{h}_6 = (1\ 0\ 0\ 1\ 3\ 2\ 3\ 3\ 1)$ $\mathbf{h}_7 = (1\ 1\ 1\ 1\ 0\ 1\ 1\ 1)$
6*	2	Z_8	4.000	42	2.857	3.01dB(BPSK)	$\mathbf{G_C} = \begin{pmatrix} 1 & 0 & 0 & 3 & 5 & 7 \\ 0 & 1 & 3 & 0 & 5 & 7 \end{pmatrix}$
9	6	Z_8	1.286	10	4.200	1.09dB(QPSK)	$\mathbf{h}_1 = (7\ 5\ 2\ 4\ 6\ 1\ 2\ 5\ 1)$ $\mathbf{h}_2 = (2\ 4\ 7\ 7\ 3\ 4\ 5\ 1)$ $\mathbf{h}_3 = (6\ 6\ 6\ 6\ 3\ 6\ 1)$
12	8	Z_8	1.494	12	5.333	1.74dB(QPSK)	$\mathbf{h}_1 = (6\ 1\ 3\ 6\ 2\ 6\ 7\ 5\ 0\ 7\ 7\ 1)$ $\mathbf{h}_2 = (7\ 4\ 6\ 6\ 1\ 6\ 4\ 5\ 6\ 5\ 1)$ $\mathbf{h}_3 = (5\ 1\ 2\ 2\ 7\ 5\ 2\ 3\ 2\ 1)$ $\mathbf{h}_4 = (5\ 1\ 1\ 1\ 4\ 0\ 1\ 4\ 1)$
15	10	Z_8	1.699	6	5.333	2.30dB(QPSK)	$\mathbf{h}_1 = (0\ 6\ 4\ 4\ 2\ 2\ 1\ 6\ 7\ 0\ 7\ 2\ 1\ 3\ 1)$ $\mathbf{h}_2 = (5\ 5\ 0\ 4\ 5\ 3\ 7\ 4\ 4\ 6\ 5\ 7\ 2\ 1)$ $\mathbf{h}_3 = (5\ 4\ 3\ 4\ 1\ 7\ 5\ 4\ 5\ 6\ 6\ 3\ 1)$ $\mathbf{h}_4 = (4\ 5\ 5\ 2\ 2\ 0\ 0\ 0\ 6\ 4\ 4\ 1)$ $\mathbf{h}_5 = (1\ 4\ 5\ 0\ 5\ 0\ 1\ 4\ 1\ 1\ 1)$
4*	1	Z_{16}	2.631	2	1.000	1.19dB(BPSK)	$\mathbf{G_C} = (1\ 2\ 5\ 8)$
5*	1	Z_{32}	3.172	2	2.000	2.00dB(BPSK)	$\mathbf{G_C} = (1\ 3\ 5\ 16\ 25)$
6**	1	Z_{64}	3.539	2	3.000	2.48dB(BPSK)	$\mathbf{G_C} = (1\ 48\ 8\ 27\ 42\ 52)$
7**	1	Z_{128}	3.787	2	3.000	2.77dB(BPSK)	$\mathbf{G_C} = (1\ 105\ 41\ 78\ 93\ 96\ 98)$
8**	1	Z_{256}	4.015	2	5.000	3.03dB(BPSK)	$\mathbf{G_C} = (1\ 190\ 188\ 26\ 19\ 153\ 143\ 100)$

* indicates an exhaustive search

** indicates an incomplete random search

Table 3.4: Bandwidth efficient codes

computer searches for optimal generator matrices and are marked with an asterisk. These codes are interesting because of their simplicity in terms of the small number of codewords. Those marked with a double asterisk were found by an incomplete random search, and are therefore not necessarily optimum. The remaining codes were found using the method of section 3.3. It should be noted that these bandwidth efficient codes use codeword overlapping, as was the case for the bandwidth expanding codes of the

previous section. We have, therefore, that blocks are transmitted as in (3.33) and $\mathbf{G_c}$ is such that (3.44) is satisfied.

We note that some of the BPSK codes use rings having orders larger than four. Most traditional coded-modulation schemes double the size of the modulation format, since it was shown in [6] that increasing the modulation order by a factor of more than two does not significantly improve performance. This does not mean that codes in these larger constellations have no purpose. In our case, for instance, the $(8,4)$ code in $\mathcal{Z}_4$, achieving an asymptotic coding gain of 3dB over uncoded coherent BPSK, has 256 codewords, whereas an asymptotically equivalent code in $\mathcal{Z}_8$ has 64 codewords. Clearly, from a complexity standpoint the code in $\mathcal{Z}_8$ is preferable. With coherent detection, however, the penalty for using higher order modulation formats is quite severe, since carrier phase tracking becomes more difficult as the modulation size grows, and may also result in some performance degradation. With non-coherent detection, this problem is non-existent, since carrier phase tracking is not performed. We may, therefore, consider codes in higher order rings, even from a practical standpoint.

We now present the performance of some of the bandwidth efficient codes. We must note that, as in the previous section, several of the codewords with small d^2_{NC} are used and not only the nearest neighbours. Fig.3.10 shows the performance of three very simple BPSK equivalent codes. They are simple in the sense that they have a small number of codewords (≤ 64). The most powerful of the three, is really only powerful in the asymptotic sense because the number of nearest neighbours is almost as large as the total number of codewords, and consequently it attains its asymptotic coding gain very slowly. In Fig.3.11 the performance of three more powerful codes, all in $\mathcal{Z}_4$ is shown. They all achieve significant improvement over uncoded BPSK even at moderate P_b. The final set of BPSK equivalent codes is shown in Fig.3.12. They use increasingly higher order rings to achieve coding gain, which, with non-coherent detection, poses no practical disadvantage, as was mentioned earlier. Although these codes all have

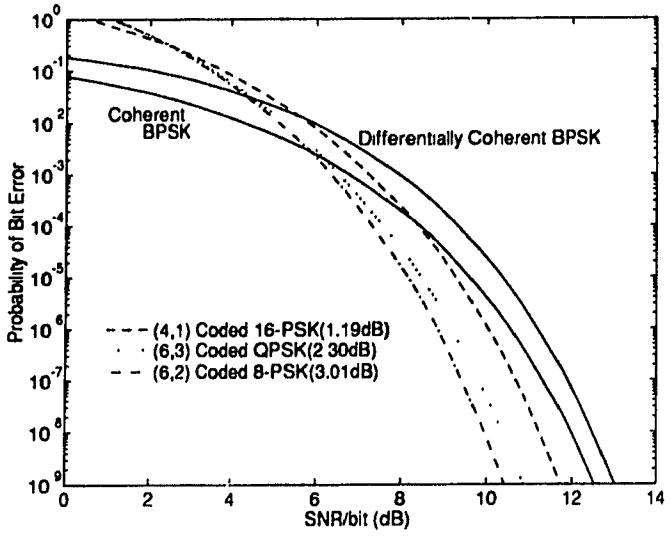

Figure 3.10: Performance of some simple BPSK equivalent codes

small number of nearest neighbours (two in each case), they all have many codewords with distances very near the minimum distance. They do not, therefore, achieve their asymptotic performance as quickly as one would expect looking only at the nearest neighbours. Fig.3.13 shows the performance of three QPSK equivalent codes. Clearly the $(15, 10)$ code is the best since it has very good asymptotic performance and a small number of nearest neighbours. The main drawback with these three codes is the large number of codewords; in the case of the (15,10) code, it has 2^{30}. Obviously, such large codes cannot be decoded using a brute force ML decoder, necessitating some sort of reduced complexity decoding strategy in order for them to be practical.

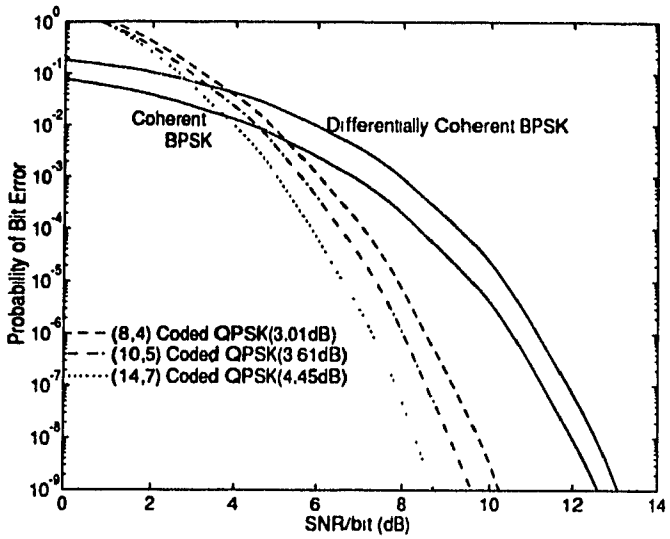

Figure 3.11: Performance of some more powerful BPSK equivalent codes

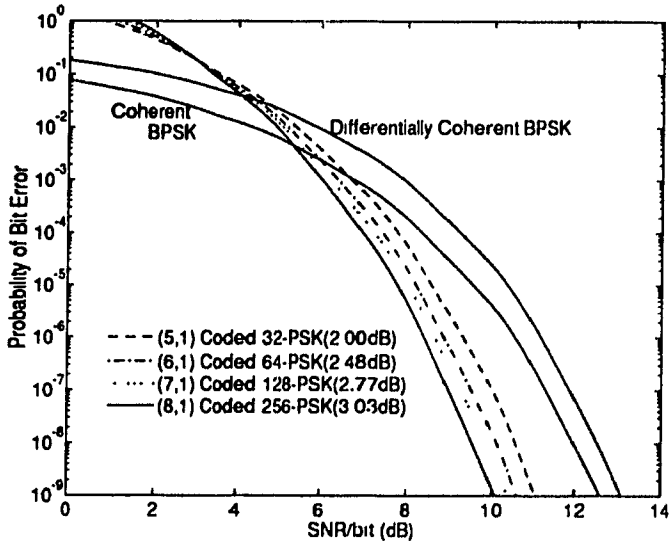

Figure 3.12: Performance of some BPSK equivalent codes in higher order rings

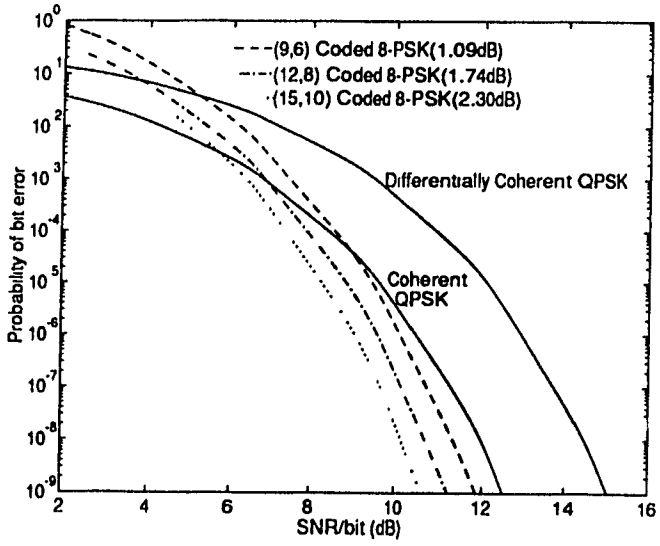

Figure 3.13: Performance of some QPSK equivalent codes

Chapter 4

Efficient Decoding of Module–Phase Codes with Non–Coherent Detection

While the ML decoder is optimal, the brute force decoding strategy suggested by (2.15) requires M^K comparisons and is therefore computationally inefficient. Moreover, despite the moderate block lengths of the codes we are considering, it is far too impractical for most of them. For this reason, the focus of this section will be on reduced-complexity decoding strategies which attempt to make some the codes more practical from an implementation standpoint. In some cases, the reduction in complexity comes at the expense of some performance. Another shortcoming of the brute force decoding strategy is that it performs a completely unstructured search through the code, neglecting its inherent algebraic structure. We would like to use this added structure to our advantage in the design of more efficient decoders.

4.1 A General Two–Stage Decoding Strategy

Let us consider a general framework for a two–stage decoding strategy. During the first stage the decoder performs hard decisions on each symbol of the received codeword and, based on these decisions, creates a list of postulated transmitted codewords, $\mathcal{P}$. In order to create $\mathcal{P}$, the decoder will use knowledge of both the noise characteristics of the channel and the algebraic structure of the code. In the second stage it performs ML decoding on $\mathcal{P}$, in order to find the most–likely transmitted codeword among the set of postulates.

We may express the probability of error for such a two–stage scheme as follows,

$$P_E \leq P_E(\text{stage 1}) + (1 - P_E(\text{stage 1}))P_E(ML) \tag{4.1}$$

where $P_E(\text{stage 1})$ is the probability of making an error during the first stage, and $P_E(ML)$ is the probability of error of the ML decoder. Clearly, the decoder will always make an error in the first stage of decoding if the transmitted codeword is not among the set of postulates, and therefore the probability of making an error during the first stage of decoding is the probability of this event. In terms of performance, the decoder will be effective compared with the ML decoder, if $P_E(\text{stage 1})$ is smaller than $P_E(ML)$. As far as complexity is concerned, the decoder will be effective if $|\mathcal{P}| \ll M^K$. These two initial observations reveal that this type of decoder offers a performance versus complexity tradeoff. More precisely, it would seem that if we were willing to sacrifice some complexity by increasing $|\mathcal{P}|$ we could reduce $P_E(\text{stage 1})$ and vice versa.

The reduced–complexity receiver structures for non–coherent detection of MDPSK in [2] may be cast into this framework. It must be pointed out that these techniques were meant for uncoded systems which afford less flexibility in the design of the receiver. These schemes use single–symbol differential detection to perform hard decisions on the received sequence. The set of postulates $\mathcal{P}$ is created by adding a subset of the most likely channel error patterns to the hard decisions, based on the reliability of the hard

decisions. Maximum-likelihood detection is then performed on the set of postulates. Using this as a starting point, we will generalize this approach for the coded case.

4.2 Two-Stage Non-Coherent Decoding of Module-Phase Codes

The first stage of the general decoder presented in the previous section requires that hard decisions be performed on each symbol of the received block. These hard decisions must be performed in the presence of an unknown phase offset induced by the channel. Let us assume that the transmitted codeword belongs to an (N, K) code in $\mathcal{Z}_M$ with generator matrix, $\mathbf{G}_c$, and is transmitted using codeword overlapping, as described in Chapter 3. The decoder must process the single symbol correlations, y_j. We may express the correlations corresponding to the block $\mathbf{b_i}$, which were defined in (2.13), as

$$y_{ij} = T \exp \jmath \left(\phi + \left(\frac{2\pi}{M} \right) b_{ij} \right) + n_{ij} \tag{4.2}$$

where ϕ is the unknown phase offset induced by the channel and n_j is a complex gaussian random variable with mean zero and variance N_0. The term $\left(\frac{2\pi}{M}\right) b_{ij}$ is an element of the set $\left(\frac{2\pi}{M}\right) \mathcal{Z}_M = \left\{0, \left(\frac{2\pi}{M}\right), \left(\frac{2\pi}{M}\right) 2, \cdots, \left(\frac{2\pi}{M}\right)(M-1)\right\}$, whose elements belong to the ring of real numbers modulo-2π, $\mathcal{R}_{2\pi}$. In order to obtain hard decisions on the received block we must compute the phase of each of the y_{ij}, each of which may be expressed as

$$\theta_{ij} = \left(\frac{2\pi}{M} \right) b_{ij} + \phi + \eta_{ij} \tag{4.3}$$

where η_{ij} is a random phase shift induced by n_{ij}. We should note that $\phi, \eta_{ij} \in \mathcal{R}_{2\pi}$. Let $\mathbf{\Theta_i}$ be the $(N+1)$-dimensional vector

$$\begin{aligned} \mathbf{\Theta_i} &= [\theta_{i0}, \theta_{i1}, \cdots, \theta_{iN}] = \left(\frac{2\pi}{M} \right) \mathbf{b_i} + \phi \mathbf{1} + \mathbf{N_i} \\ &= \left(\frac{2\pi}{M} \right) [b_{i-1,N} \quad \mathbf{x_i}] \mathbf{G} + \phi \mathbf{1} + \mathbf{N_i} \end{aligned} \tag{4.4}$$

which is processed in the first decoding stage, where $\mathbf{1}$ is an all–one vector, and $\mathbf{N_i} = [\eta_{i0}, \eta_{i1}, \cdots, \eta_{iN}]$.

4.2.1 Generalized Differential Detection

Let us define an $(N+1) \times N$ *generalized differential decoding matrix* $\mathbf{Q}$ over $\mathcal{R}_{2\pi}$, which is a phase–offset annihilator when it operates on $\mathbf{\Theta_i}$. Assuming that $\mathbf{Q}$ will operate on the right of $\mathbf{\Theta_i}$, we require, therefore, that

$$\phi \mathbf{1Q} = \mathbf{0}, \quad \phi \in \mathcal{R}_{2\pi}. \tag{4.5}$$

This condition implies that every column of $\mathbf{Q}$ must sum to zero. Let us now examine the effect of $\mathbf{Q}$ on the entire block,$\mathbf{\Theta_i}$. Looking at the matrix product, $\left[\left(\frac{2\pi}{M}\right)\mathbf{G}\right]\mathbf{Q}$ we have

$$\left[\left(\frac{2\pi}{M}\right)\mathbf{G}\right]\mathbf{Q} = \left(\frac{2\pi}{M}\right)\left(\begin{array}{c|ccc} 1 & 1 & 1 & \cdots & 1 \\ \hline 0 & & & & \\ 0 & & \mathbf{G_c} & & \\ \vdots & & & & \\ 0 & & & & \end{array}\right)\mathbf{Q} = \begin{pmatrix} 0 & 0 & \cdots & \cdots & 0 \\ \left[\left(\frac{2\pi}{M}\right)\mathbf{G_c}\right]\mathbf{Q_s} & & & & \end{pmatrix} \tag{4.6}$$

where $\mathbf{Q_s}$ is the $N \times N$ matrix formed by removing the first row from $\mathbf{Q}$. We have, therefore, that

$$\begin{aligned} \mathbf{\Theta_i Q} &= \left[\left(\frac{2\pi}{M}\right)\mathbf{x_i G_c}\right]\mathbf{Q_s N_i Q} \\ &= \mathbf{b_i}' + \mathbf{N_i}' \end{aligned} \tag{4.7}$$

Since we will be performing hard decisions on the vector $\mathbf{\Theta_i Q}$, which corresponds to quantizing the components of $\mathbf{\Theta_i Q}$ to the closest elements in $\left(\frac{2\pi}{M}\right)\mathcal{Z}_M$, we require that

$$b'_{ij} \in \left(\frac{2\pi}{M}\right)\mathcal{Z}_M \quad , 0 \le j \le N. \tag{4.8}$$

This is because, in the absence of the noise term $\mathbf{N_i}'$, the result of performing hard decisions on $\mathbf{\Theta_i Q}$ must be $\mathbf{b_i}'$. If this is the case, we can surely recover the transmitted codeword, $\mathbf{x_i G_c}$, correctly, by applying $\mathbf{Q_s}^{-1}$ to the hard decision vector. We must assure, therefore, that $\mathbf{Q_s}$ is invertible.

An important aspect to consider when selecting $\mathbf{Q}$ is its effect on the noise vector $\mathbf{N_i}$. We would like to find a form for $\mathbf{Q}$ which minimizes the noise variance in each component of $\mathbf{N_i}'$. Since the components of $\mathbf{N_i}$ are all independent and identically distributed, the number of times any one of these components is summed to form $\mathbf{N_i Q}$ should be as small as possible. This assures that the effect that any one component of $\mathbf{N_i}$ has on the entire vector $\mathbf{N_i Q}$ is minimal. It is reasonable, therefore, to minimize the number of non–zero elements in each column of $\mathbf{Q}$. We should also assure that $|Q_{i,j}| \leq 1$ so that the noise variance is never amplified. It should be clear that the $|Q_{i,j}|$'s cannot be greater than zero and strictly less than one, since (4.8) would, in general, be violated. The components of $\mathbf{Q}$ must, therefore, only assume the three values -1,0 or 1, and in order to satisfy (4.5), there should be only a single (1,-1) pair in each column. This implies that each component of $\mathbf{N_i Q}$ is a difference of two components of $\mathbf{N_i}$.

The form that we will use for $\mathbf{Q}$ which is, in fact, a matrix representation for ordinary differential detection is given by

$$\mathbf{Q} = \begin{pmatrix} -1 & 0 & \cdots & 0 & 0 & 0 \\ 1 & -1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & -1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 1 & -1 \\ 0 & 0 & \cdots & 0 & 0 & 1 \end{pmatrix}. \tag{4.9}$$

This is, of course, not the only possible good choice for $\mathbf{Q}$. In general, all matrices which have $N-1$ rows with two non–zero elements and two rows with a single non–zero element which satisfy the other requirements will be equivalent since all the components of $\mathbf{N_i}$ are independent.

Let us denote the vector made up of the hard decisions by $\mathbf{r}_H$, which is given by

$$\mathbf{r}_H = \left(\frac{2\pi}{M}\right)\mathbf{x_i}\mathbf{G_c}\mathbf{Q_s} + \mathbf{e}_H, \tag{4.10}$$

where $\mathbf{e}_H$ is the error–pattern resulting from making hard decisions on the received block. Let $\mathbf{r}$ be the vector from $\mathcal{Z}_M$ which corresponds to $\mathbf{r}_H$, $\mathbf{r} = \mathbf{x_i}\mathbf{G_c}\mathbf{Q_s} + \mathbf{e}$. We have, therefore, that $\mathbf{r}$ is the sum of a codeword from the code generated by $\mathbf{G_c}\mathbf{Q_s}$ and an error pattern induced by the channel.

4.2.2 Information Set Decoding

The main function of the first stage of the decoder is to create a set of postulate codewords $\mathcal{P}$. We would like to create $\mathcal{P}$ such that at least one of the postulates is the transmitted codeword, which means that we must somehow cancel or *cover* the error pattern $\mathbf{e}$ in (4.10). A method for doing this, known as *Information Set Decoding* was first used by Prange in [8] for decoding binary cyclic block codes. This method has also been used for *soft–decision* decoding of binary block codes [15, p.102], which is exactly what we intend to do for module–phase codes. The main feature of the method is that it exploits properties of vector spaces in order to cover a specific number of error patterns. We will exploit similar properties for the more general case of modules in order to decode the codes of Chapter 3.

Let us assume that we have a received vector, $\mathbf{r}$ given by

$$\mathbf{r} = \mathbf{x_i}\mathbf{G} + \mathbf{e} \tag{4.11}$$

where $\mathbf{G}$ is the generator matrix for an arbitrary (N, K) code in $\mathcal{Z}_M$. It should be clear that if the information positions of the received codeword are error–free, we can reconstruct the entire codeword correctly simply by applying $\mathbf{G}$ to the information vector. The key property of linear codes which is exploited by this technique is that for a given code there may be several sets of K symbols, called *information sets*, which can be used

to generate the code. By using several information sets we may cover, in the best case, any error pattern made up of $N-K$ or fewer errors. This is because, in the best case, any of the $\binom{N}{K}$ information sets can be used to generate the code, and there will always be at least one which is error-free for any error pattern having $N-K$ or fewer errors. In reality, we cannot expect that any information set can be used to generate the code, since this would require that all $K \times K$ sub-matrices of $\mathbf{G}$ be non-singular. This is especially true for matrices over rings due to the larger number of zero divisors. Because of this we call information set *useful* if it can be used to generate the entire code. More precisely, we have the following definition

Definition 3 *An information set, denoted by the vector* $\mathbf{l}$ *whose* K *components are the locations of information positions, is* useful *if* $\mathbf{G}$ *can be manipulated via elementary row operations to form a matrix* $\mathbf{G_l}$*, such that the columns of* $\mathbf{G_l}$ *corresponding to* $\mathbf{l}$ *form the* $K \times K$ *identity matrix.*

If we define a matrix $\mathbf{U_l}$ which is the $K \times K$ matrix formed by the columns of $\mathbf{G}$ which correspond to $\mathbf{l}$, we have that $\mathbf{l}$ is useful if $\mathbf{U_l}$ is non-singular in $\mathcal{Z}_M$. If this is the case, we have that the generator matrix corresponding to $\mathbf{l}$ is given by

$$\mathbf{G_l} = \mathbf{U_l^{-1}G}. \tag{4.12}$$

In the design of the decoder, we determine the maximum number of errors, t, that we wish to cover. Assuming that this can be done for the code we are using, we choose the minimum number of information sets needed to cover up to t errors, and compute the appropriate generator matrix for each information set. This can be done simply by using a greedy algorithm which forms of list of the error positions and chooses useful information sets one at a time, so that a maximum number of the error patterns are deleted from the list at each step. This is continued until all the error patterns are covered. The decoding strategy is quite simple once these have been determined, and is as follows:

1. For each information set, $\mathbf{l}$
2. Create the information vector, $\mathbf{x_l}$ made up of the components of $\mathbf{r}$ corresponding to $\mathbf{l}$
3. Form the codeword $\mathbf{c_l} = \mathbf{x_l G_l}$
4. Add $\mathbf{c_l}$ to $\mathcal{P}$
5. If there are information sets remaining go back to 1
6. Choose the codeword in $\mathcal{P}$ closest to the received block according to some distance criterion as the decoder output

The complexity of this strategy, in terms of number of comparisons, is proportional to the number of information sets, N_t, needed to cover any t–error pattern.

4.2.3 Modifying the basic scheme for non–coherent detection

The received vector, $\mathbf{r}$, in the case of non–coherent detection is the result of making hard decisions on the differentially–detected block, $\Theta_\mathbf{i}\mathbf{Q}$ and is given in (4.10). In this case, the generator matrix which must be used in the algorithm is not $\mathbf{G_c}$, the generator matrix of the code, but rather $\mathbf{G_c Q_s}$. In addition, the postulated codeword that is added to $\mathcal{P}$ must be post–multiplied by $\mathbf{Q_s^{-1}}$ to reverse the effects of $\mathbf{Q}$ on the received block. For the general form for $\mathbf{Q}$ in (4.9), we have that $\mathbf{Q_s^{-1}}$ is given by

$$\mathbf{Q_s}^{-1} = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 & 1 \\ 0 & 1 & 1 & \cdots & 1 & 1 \\ 0 & 0 & 1 & \cdots & 1 & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 1 & 1 \\ 0 & 0 & \cdots & 0 & 0 & 1 \end{pmatrix} \tag{4.13}$$

The basic assumption in the information set decoding algorithm is that there is at least one information set that is error-free. Unfortunately, because we are using M-ary modulation combined with the fact that we employ differential detection to perform hard decisions in the first stage of decoding, this is a fairly unrealistic assumption. A more reasonable one, in this case, would be that at least one information set contains only ***small*** errors. By small, we mean that no symbol in the information set differs from the actual symbol in the transmitted codeword by more than one element. We assume, therefore, that each symbol in the information set is either the actual transmitted symbol, s, or one of the two symbols closest to s, $s+1$ or $s-1$. In order to cover these information set errors, we will use the 2^L algorithm proposed by Wilson *et al.* in [2] (see chapter 2), on each information set. This method will cover any small errors in the information set, while the more serious errors will be trapped in the parity set, as in normal information set decoding.

In the design of the modified decoder, we first must decide on the number of information set errors that are to be covered, L ($L \leq K$). We must then determine the maximum number of serious errors, t, that we wish to cover. As before, we assume that this can be done for the code we are using and we choose the minimum number of information sets needed to cover up to t errors, and compute the appropriate generator matrix for each information set. The modified decoding strategy is as follows:

1. For each symbol in the block, $\Theta\mathbf{Q}$, determine the best and second best hard decisions and the reliabilities of the best decisions
2. For each information set, l
3. Determine the $K-L$ most reliable symbols and set those positions to best decision in the vector $\mathbf{x}_l$,
4. For each of the 2^L possible choices for the remaining positions in $\mathbf{x}_l$, j
5. Form the codeword $\mathbf{c}_{lj} = \mathbf{x}_l\mathbf{G}_l$

6. Add $\mathbf{c}_{\mathbf{lj}}\mathbf{Q}_{\mathbf{s}}^{-1}$ to $\mathcal{P}$

7. If there are additional possible information set vectors return to 4

8. If there are information sets remaining go back to 2

9. Choose the codeword in $\mathcal{P}$ closest in terms of non-coherent distance to the received block $\mathbf{y}$ as the decoder output

The complexity of the this combined scheme, in terms of the number of comparisons, is proportional to $N_t 2^L$. In order to reduce the 2^L factor in the complexity of this scheme, we may use only a subset of the 2^L possible information set choices. For instance, we may choose only those vectors which contain less than a certain number of second best choices. If we were to choose only those vectors which have at most a single second best choice, it would the same as Wilson's $N+1$ algorithm in [2], and would make the complexity $N_t(K+1)$. This reduction in complexity would have to be weighed with a possible loss in performance.

4.3 Searching for codes better suited to Information Set Decoding

The only criterion that was used in the code search described in the previous chapter was to maximize the minimum non-coherent distance, d_{NC}^2. As soon as a satisfactory code was found, the search was terminated. As far as the brute force ML decoder is concerned, these codes are adequate. This may or may not be the case when we use an information set decoder. More precisely, because of performance requirements, we may require to cover a certain number of error patterns which cannot be covered by the generator matrix of the code that was found. For this reason, it is possible that the occasion will arise where we must search for equivalent codes, in terms of distance

properties, which are better suited for information set decoding. By this we mean that a code which has a larger number of useful information sets is more effective for this type of decoding.

A simple modification of the search presented in the last chapter can be performed to find more amenable codes. Let us assume that we want to search for a code equivalent to one which was already found which has $N-K$ parity-check vectors $\mathbf{h_i}, 1 \leq i \leq N-K$. We leave any $N-K-1$ of these vectors as they are. Let us call the parity-check vector which we singled out $\mathbf{h_s}(0)$, $1 \leq s \leq N-K$ which is the vector we will use in the search. Using the original list of vectors which had to be excluded from the code, L_0, we retain the vectors which remain after applying the $N-K-1$ parity check vectors, and call this smaller list L_s. Create the list L'_s by retaining the $K+1-s$ leading positions of each vector in L_s. It should be clear, that no vector in L'_s is in the kernel of $\mathbf{h_s}(0)$. Starting from the initial value $\mathbf{h_s}(0)$ we continue searching for new $\mathbf{h_s}(k)$ such that no vector in L'_s is in its kernel, which means that $\mathbf{h_s}(k)$ combined with the other $\mathbf{h_i}$ define an equivalent code. At the same time we determine the useful information sets for the matrix

$$\mathbf{G} = \mathbf{G_{N-K}G_{N-K-1}}\cdots\mathbf{G_s(k)}\cdots\mathbf{G_1G_0Q_s} \tag{4.14}$$

where $\mathbf{G_s}(k)$ is the generator matrix corresponding to the parity-check vector, $\mathbf{h_s}(k)$. As soon as we find a code which has the desired characteristics, the search is terminated. If no such code was found, we may try to repeat the procedure with a different search vector $\mathbf{h_r}(0)$, $1 \leq r \leq N-K, r \neq s$.

4.4 Decoding of Various Module–Phase–Codes

In this section we will consider various examples of codes, and how they may be efficiently decoded, both in terms of complexity and performance, using the modified information set decoding algorithm consider in the previous section. We will consider three sets of

codes

1. bandwidth expanding codes for 8-PSK
2. bandwidth expanding codes for QPSK
3. bandwidth efficient codes equivalent to uncoded BPSK

and present results through the use of computer simulations.

4.4.1 Computer Simulations

Simulation software for the modified information set decoding algorithm over a non–coherent channel was written in C. The program requires an input file to characterize the coded system which contains the following parameters:

1. The code parameters, N,K and M
2. The number of information sets to be used, N_t
3. Each information set followed by its corresponding generator matrix

The software simulates a complex baseband MPSK system by transmitting the zero codeword across an additive white gaussian noise channel, while accumulating the number of bit errors of the decoded output, assuming Gray Coding is used. Complex gaussian noise is simulated using a uniform pseudo–random sequence generator and an appropriate transformation. The user supplies a range of SNR values and the number of experiments to be performed for each noise power level, along with the number of trials to be performed per experiment. The error statistics of the decoded output, for a given SNR, consist of the average value for the probability of error over the collection of experiments, and an estimate of the standard deviation.

The bit error-rates of the reduced-complexity decoding strategies are compared, in each case, with the union bound for a ML decoder, an uncoded coherent PSK system and an uncoded differentially-coherent system. The simulation curves are plotted using a cubic-spline interpolation of simulation points spaced 1dB apart. The points are averages over ten experiments, with the number of trials per experiments chosen so as to have small standard deviation (shown in the figures as vertical bars at each simulation point).

4.4.2 Bandwidth Expanding Codes in $\mathcal{Z}_4$

A (6,5) Code with $d^2_{NC} = 2.00$

This is a simple parity-check code in $\mathcal{Z}_4$ which has an asymptotic coding gain of 2.22dB over uncoded coherent QPSK. It has the following generator matrix (see Appendix A)

$$\mathbf{G_c} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 3 \\ 0 & 1 & 0 & 0 & 0 & 3 \\ 0 & 0 & 1 & 0 & 0 & 3 \\ 0 & 0 & 0 & 1 & 0 & 3 \\ 0 & 0 & 0 & 0 & 1 & 3 \end{pmatrix}. \tag{4.15}$$

We have, therefore, that the generator matrix used in the decoder is $\mathbf{G_{dec}} = \mathbf{G_c Q_s}$, which when expressed in systematic form is given by

$$\mathbf{G_{dec}} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 2 \\ 0 & 1 & 0 & 0 & 0 & 3 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 2 \end{pmatrix}. \tag{4.16}$$

For this code we will use only one information set, and therefore only attempt to cover small errors in the information set. We have chosen to simulate two decoders, one with $L = K$, and the other with $L = K - 1$. The results of the simulations are shown in Fig.4.1. We see that the reduced-complexity strategy with $L = K$ performs

noticeably better than the union bound for a ML decoder, which means that $P_E(\text{stage1})$, the probability that the transmitted codeword is not in the set of postulates, is inferior to $P_E(ML)$. The $L = K$ decoder requires 32 comparisons, compared with the 1024 of a ML decoder. At $P_b = 10^{-5}$ we attain a coding gain of approximately 1dB over coherent QPSK, and close to 3.5dB over differentially-coherent QPSK. The $L = K - 1$ decoder requires half the number of comparisons (16) but suffers from a noticeable performance degradation (.25dB at $P_b = 10^{-5}$).

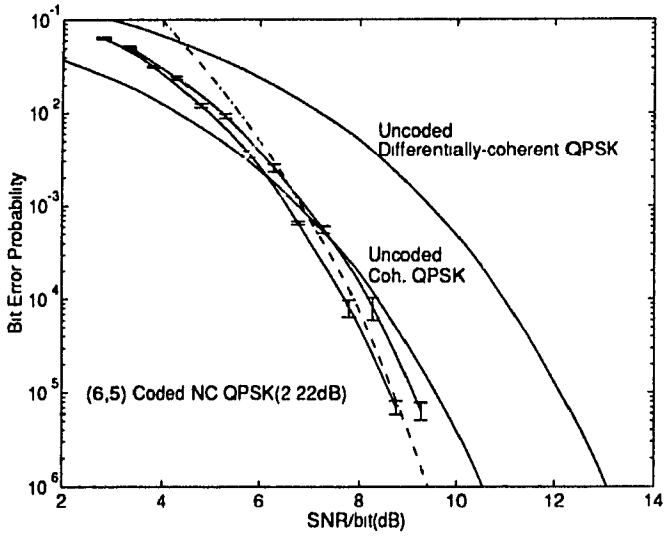

Figure 4.1: A (6,5) Code in $\mathcal{Z}_4$

A (7,4) Code with $d^2_{NC} = 3.53$

This is a more powerful code which has an asymptotic coding gain of 3.04dB over uncoded coherent QPSK. Consequently, we must cover more errors during the first stage of decoding. The generator matrix for the code from in Appendix A is capable of covering single errors, which we have found to be insufficient. It was therefore necessary to search for a code which is equivalent, in terms of distance properties, and has a larger number of useful information sets. Using the extended search method of section 4.3

resulted in an equivalent code which is capable of covering up to two errors using seven information sets. The generator matrix after performing an extended search is

$$\mathbf{G_c} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 3 & 1 \\ 0 & 1 & 0 & 0 & 2 & 0 & 3 \\ 0 & 0 & 1 & 0 & 2 & 3 & 2 \\ 0 & 0 & 0 & 1 & 0 & 2 & 1 \end{pmatrix}. \tag{4.17}$$

The seven information sets and their corresponding generator matrices used by the decoder are given in Table 4.1. It should be clear that all seven matrices are row-wise equivalent to $\mathbf{G_c Q_s}$.

Information Set	Generator Matrix	Information Set	Generator Matrix
0 1 2 3	$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 3 & 3 \\ 0 & 1 & 0 & 0 & 3 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 3 & 2 \\ 0 & 0 & 0 & 1 & 3 & 2 & 3 \end{pmatrix}$	0 1 4 5	$\begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 3 \\ 0 & 0 & 2 & 1 & 1 & 0 & 3 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 \end{pmatrix}$
0 2 5 6	$\begin{pmatrix} 1 & 1 & 0 & 0 & 3 & 0 & 0 \\ 0 & 3 & 1 & 1 & 1 & 0 & 0 \\ 0 & 3 & 0 & 3 & 2 & 1 & 0 \\ 0 & 2 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}$	0 3 4 6	$\begin{pmatrix} 1 & 3 & 3 & 0 & 0 & 3 & 0 \\ 0 & 1 & 2 & 1 & 0 & 1 & 0 \\ 0 & 2 & 3 & 0 & 1 & 3 & 0 \\ 0 & 3 & 3 & 0 & 0 & 0 & 1 \end{pmatrix}$
1 2 4 6	$\begin{pmatrix} 2 & 1 & 0 & 3 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 3 & 1 & 1 & 0 \\ 3 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}$	1 3 5 6	$\begin{pmatrix} 1 & 1 & 0 & 0 & 3 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 2 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 3 & 0 & 3 & 0 & 1 \end{pmatrix}$
2 3 4 5	$\begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 3 \\ 1 & 3 & 0 & 1 & 0 & 0 & 1 \\ 3 & 3 & 0 & 0 & 1 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}$		

Table 4.1: Information sets and generator matrices for decoding a (7,4) code in $\mathcal{Z}_4$

The simulation results for this decoder are shown in Fig.4.2. As for the last code, two decoders were simulated, one having $L = K$ and the other having $L = K - 1$. In both cases, the curves are below the union bound for a ML decoder down to $P_b = 10^{-5}$. The $L = K$ decoder requires 112 comparisons compared to 256 for a ML decoder, whereas the $L = K - 1$ decoder requires 56. There is, however, a slight degradation ($\approx$ .2dB) as a result of this reduction in complexity. For the $L = K$ decoder a coding gain of approximately 2dB is attained over uncoded coherent QPSK at $P_b = 10^{-5}$, and approximately 4.2dB over uncoded differentially-coherent QPSK.

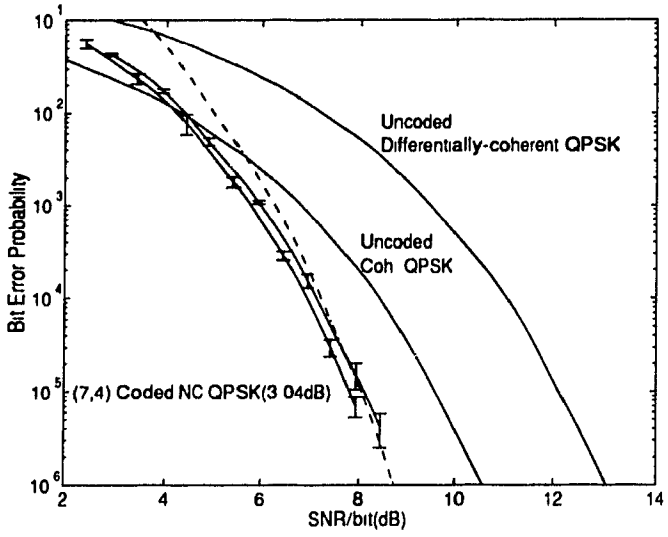

Figure 4.2: A (7,4) Code in $\mathcal{Z}_4$

4.4.3 Bandwidth Expanding Codes in $\mathcal{Z}_8$

A (5,4) Code with $d^2_{NC} = .586$

This is a simple parity-check code in $\mathcal{Z}_8$ which has an asymptotic coding gain of 2.07dB over uncoded coherent 8–PSK. It has the following generator matrix (see Appendix A)

$$\mathbf{G_c} = \begin{pmatrix} 1 & 0 & 0 & 0 & 7 \\ 0 & 1 & 0 & 0 & 7 \\ 0 & 0 & 1 & 0 & 7 \\ 0 & 0 & 0 & 1 & 7 \end{pmatrix}. \tag{4.18}$$

We have, therefore, that the generator matrix used in the decoder, $\mathbf{G_{dec}} = \mathbf{G_c Q_s}$, is given in systematic form by

$$\mathbf{G_{dec}} = \begin{pmatrix} 1 & 0 & 0 & 0 & 3 \\ 0 & 1 & 0 & 0 & 4 \\ 0 & 0 & 1 & 0 & 5 \\ 0 & 0 & 0 & 1 & 6 \end{pmatrix}. \tag{4.19}$$

As was the case for the parity-check code in $\mathcal{Z}_4$ we will use only one information set. We will also choose $L = K = 4$ so that no hard decisions are performed on the

information set. The results of the simulation are shown in Fig.4.3. We see that the reduced complexity strategy performs quite close to the union bound for a ML decoder down to $P_b = 10^{-5}$. Its' complexity is also significantly lower, requiring 16 comparisons instead of 4096. Even at fairly low error rates, we can achieve marginal coding gain over coherent detection (1dB for $P_b = 10^{-4}$). Compared with differentially-coherent detection, we achieve close to 4.5dB at $P_b = 10^{-4}$ with very little complexity.

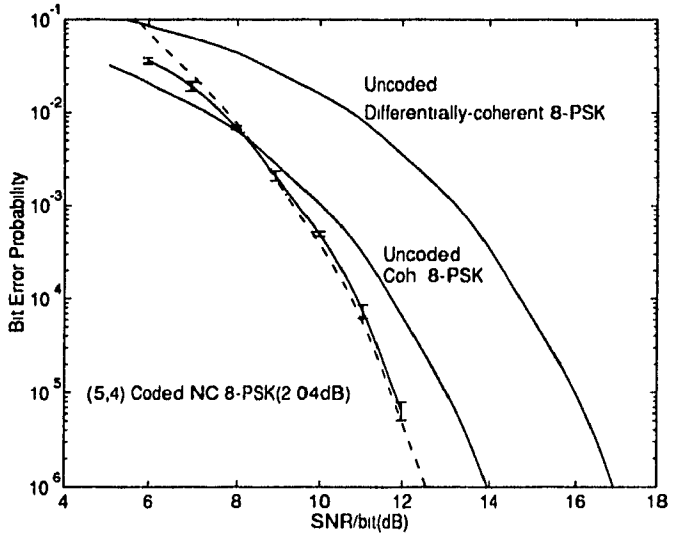

Figure 4.3: A (5,4) Code in $\mathcal{Z}_8$

A (7,5) Code with $d^2_{NC} = .844$

We will now consider the decoding of a more powerful code which has an asymptotic coding gain of 3.13dB over uncoded coherent 8–PSK. Since this code has a larger d^2_{NC} than in the previous case, we must try to cover more errors in the first stage of the decoder. In order to do so, we will try to use a minimum number of information sets to cover any single error. With the generator matrix given in Appendix A we have found that this cannot be done. A search for an equivalent code with enough useful information sets to cover these errors was performed. An equivalent code with the following generator

matrix was found

$$\mathbf{G_c} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 6 & 0 \\ 0 & 1 & 0 & 0 & 0 & 2 & 5 \\ 0 & 0 & 1 & 0 & 0 & 6 & 6 \\ 0 & 0 & 0 & 1 & 0 & 7 & 7 \\ 0 & 0 & 0 & 0 & 1 & 7 & 6 \end{pmatrix}. \tag{4.20}$$

Using this matrix, we require four information sets to cover any single error. The four generator matrices and their corresponding information sets used by the decoder are given in Table 4.2

Information Set	Generator Matrix	Information Set	Generator Matrix
0 1 2 3 4	$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 3 & 4 \\ 0 & 1 & 0 & 0 & 0 & 5 & 2 \\ 0 & 0 & 1 & 0 & 0 & 3 & 7 \\ 0 & 0 & 0 & 1 & 0 & 5 & 7 \\ 0 & 0 & 0 & 0 & 1 & 6 & 7 \end{pmatrix}$	0 3 4 5 6	$\begin{pmatrix} 1 & 3 & 2 & 0 & 0 & 0 & 0 \\ 0 & 2 & 3 & 1 & 0 & 0 & 0 \\ 0 & 7 & 5 & 0 & 1 & 0 & 0 \\ 0 & 3 & 6 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}$
1 2 3 5 6	$\begin{pmatrix} 5 & 1 & 0 & 0 & 6 & 0 & 0 \\ 1 & 0 & 1 & 0 & 3 & 0 & 0 \\ 3 & 0 & 0 & 1 & 3 & 0 & 0 \\ 3 & 0 & 0 & 0 & 4 & 1 & 0 \\ 2 & 0 & 0 & 0 & 7 & 0 & 1 \end{pmatrix}$	1 2 4 5 6	$\begin{pmatrix} 7 & 1 & 0 & 6 & 0 & 0 & 0 \\ 6 & 0 & 1 & 7 & 0 & 0 & 0 \\ 1 & 0 & 0 & 3 & 1 & 0 & 0 \\ 7 & 0 & 0 & 4 & 0 & 1 & 0 \\ 3 & 0 & 0 & 3 & 0 & 0 & 1 \end{pmatrix}$

Table 4.2: Information sets and generator matrices for decoding a (7,5) code

The simulation results are shown in Fig. 4.4. In this case, we have used $L = K = 5$. It is seen that we achieve a performance very close to the union bound for maximum likelihood decoding, with far fewer comparisons (128 instead of 32768). Even at fairly low error rates, significant coding gain can be achieved over coherent detection. Compared with differentially-coherent detection at $P_b = 10^{-4}$ we attain close to 5.5dB gain in SNR.

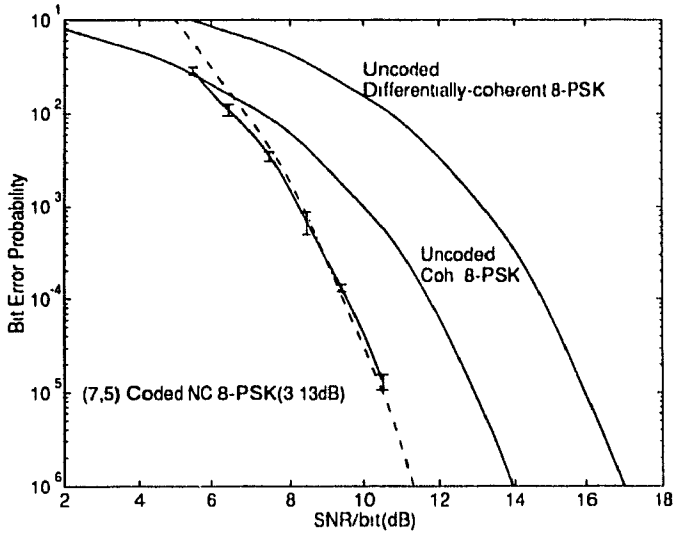

Figure 4 4: A (7,5) Code in $\mathcal{Z}_8$

4.4.4 Bandwidth Efficient Codes

An (8,4) Code in $\mathcal{Z}_4$ with $d^2_{NC} = 4.00$

This is a bandwidth efficient code which has an asymptotic coding gain of 3.01dB over uncoded coherent BPSK. As was the case for the (7,4) code in $\mathcal{Z}_4$, we have found that up to two errors may be covered using seven information sets, although not with the code whose generator matrix found in Appendix A. The resulting matrix of an equivalent code after performing an extended search is

$$\mathbf{G_c} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 3 & 3 & 2 \\ 0 & 1 & 0 & 0 & 1 & 3 & 1 & 3 \\ 0 & 0 & 1 & 0 & 3 & 2 & 3 & 3 \\ 0 & 0 & 0 & 1 & 3 & 0 & 2 & 3 \end{pmatrix}. \tag{4.21}$$

The seven information sets and their corresponding generator matrices used by the decoder are given in Table 4.3.

The simulation results for this decoder are shown in Fig.4.5. As for the (7,4) code in $\mathcal{Z}_4$, two decoders were simulated, one having $L = K$ and the other having $L = K - 1$.

Information Set	Generator Matrix	Information Set	Generator Matrix
0 1 2 3	$\begin{pmatrix} 1 & 0 & 0 & 0 & 3 & 0 & 1 & 2 \\ 0 & 1 & 0 & 0 & 2 & 2 & 1 & 3 \\ 0 & 0 & 1 & 0 & 1 & 0 & 3 & 1 \\ 0 & 0 & 0 & 1 & 2 & 1 & 2 & 1 \end{pmatrix}$	0 4 5 6	$\begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 3 \\ 0 & 3 & 3 & 2 & 1 & 0 & 0 & 2 \\ 0 & 0 & 2 & 1 & 0 & 1 & 0 & 3 \\ 0 & 3 & 2 & 2 & 0 & 0 & 1 & 1 \end{pmatrix}$
1 2 4 7	$\begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 3 & 3 & 0 \\ 3 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 3 & 0 & 0 & 2 & 1 & 2 & 3 & 0 \\ 2 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$	0 3 4 7	$\begin{pmatrix} 1 & 1 & 3 & 0 & 0 & 2 & 3 & 0 \\ 0 & 1 & 0 & 1 & 0 & 3 & 3 & 0 \\ 0 & 3 & 3 & 0 & 1 & 2 & 0 & 0 \\ 0 & 1 & 2 & 0 & 0 & 2 & 3 & 1 \end{pmatrix}$
1 5 6 7	$\begin{pmatrix} 2 & 1 & 3 & 2 & 3 & 0 & 0 & 0 \\ 3 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 3 & 0 & 2 & 0 & 3 & 0 & 1 & 0 \\ 3 & 0 & 3 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$	2 3 4 6	$\begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 3 \\ 2 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 3 & 0 & 0 & 1 & 2 & 0 & 3 \\ 2 & 3 & 0 & 0 & 0 & 2 & 1 & 1 \end{pmatrix}$
0 2 3 6	$\begin{pmatrix} 1 & 3 & 0 & 0 & 1 & 2 & 0 & 3 \\ 0 & 1 & 1 & 0 & 3 & 2 & 0 & 0 \\ 0 & 2 & 0 & 1 & 2 & 1 & 0 & 3 \\ 0 & 1 & 0 & 0 & 2 & 2 & 1 & 3 \end{pmatrix}$		

Table 4.3: Information sets and generator matrices for decoding a (8,4) code in $\mathcal{Z}_4$

The curve for $L = K$ is below the union bound for a ML decoder down to $P_b = 10^{-5}$. The $L = K$ decoder requires 112 comparisons compared to 256 for a ML decoder, whereas the $L = K - 1$ decoder requires 56. There is, however, a degradation ($\approx .25$dB) as a result of this reduction in complexity. For the $L = K$ decoder at $P_b = 10^{-5}$, a coding gain of slightly less than 2dB is attained over uncoded coherent BPSK, and approximately 2.6dB over uncoded differentially-coherent BPSK.

A (14,7) Code in $\mathcal{Z}_4$ with $d^2_{NC} = 5.566$

This is the most powerful BPSK equivalent code that was found and has an asymptotic coding gain of 4.55dB over uncoded coherent BPSK. we have found a generator matrix for a code capable of covering up to three errors using twenty information sets, which is again different that the one found in Appendix A. It is given by

$$\mathbf{G_c} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 2 & 3 & 3 & 1 & 1 & 2 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 3 & 3 & 1 & 0 & 1 & 3 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 3 & 3 & 2 & 2 & 2 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 3 & 2 & 0 & 3 & 3 & 3 & 2 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 3 & 3 & 3 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 3 & 1 & 2 & 2 & 3 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 3 & 0 & 0 & 2 & 1 & 0 & 2 \end{pmatrix}. \tag{4.22}$$

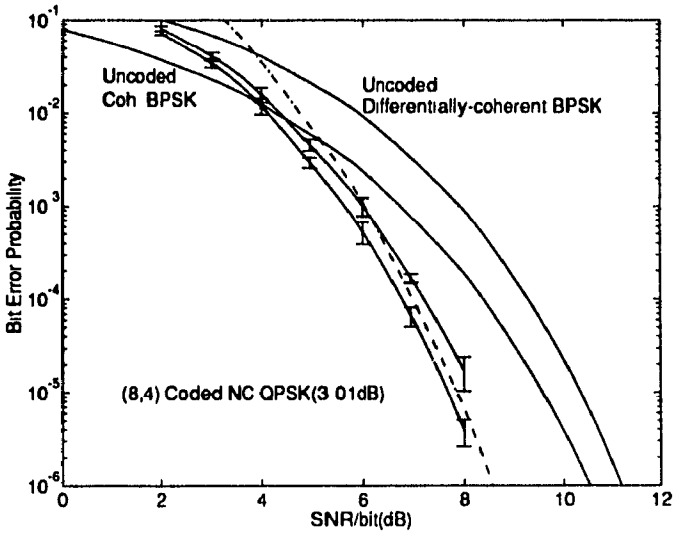

Figure 4.5: An (8,4) Code in $\mathcal{Z}_4$

The information sets are shown in Table 4.4. The corresponding generator matrices have not been included here because of space limitations, but can be found simply by (4.12). The simulation results for this decoder are shown in Fig.4.6. Since this code is much more complex than the previous ones, we were not able to simulate for low error rates ($< 10^{-5}$) because of the computing times required for accurate results. We see, however, that the simulation curve is below the union bound for a ML decoder, indicating that the reduced complexity decoder performs quite close to a ML decoder. This decoder requires 2560 comparisons, whereas the ML decoder requires 16384. At $P_b = 10^{-3}$ this code attains a coding gain of approximately 2dB over uncoded coherent BPSK, and approximately 3.5dB over uncoded differentially-coherent BPSK.

(0 1 2 3 4 5 6)	(0 1 7 8 9 10 11)
(2 3 4 7 10 12 13)	(5 6 8 9 11 12 13)
(0 1 5 6 7 10 13)	(0 1 2 3 8 9 12)
(2 3 4 5 7 8 11)	(0 4 6 9 10 11 12)
(0 4 5 8 9 10 13)	(1 4 6 7 8 9 13)
(1 2 5 10 11 12 13)	(0 2 3 6 7 11 12)
(1 3 4 5 7 11 12)	(1 2 3 6 8 10 11)
(0 4 5 6 7 8 12)	(2 3 5 6 9 10 13)
(0 1 2 4 7 11 13)	(0 3 4 6 8 10 11)
(2 5 7 8 9 10 12)	(0 1 2 3 5 6 7)

Table 4.4: Information sets for decoding a (14,7) code in $\mathcal{Z}_4$

4.5 Performance over Correlated Rayleigh Fading Channels

Up until this point we have only considered the non-coherent AWGN channel. It would be very interesting to investigate the performance of the codes combined with the reduced-complexity decoding strategies over correlated Rayleigh fading channels, which is a mathematical model appropriate for a mobile radio environment. The corresponding problem for non-coherent block detection of uncoded MPSK is considered in [22].

Most of the coding systems for fading channels employ some form of interleaving in order to decorrelate the received sequence. Although in theory these systems achieve significant performance improvements, the assumption of ideal interleaving is not valid on many practical mobile radio channels because it would require an unacceptably large time delay. Consequently, many of these systems fail to perform as expected when employed over these channels. In this section we will show that it is possible to obtain significant performance improvements using the codes of the previous chapter combined with the reduced-complexity decoding strategy. In some cases, these performance improvements can be obtained *without* symbol interleaving.

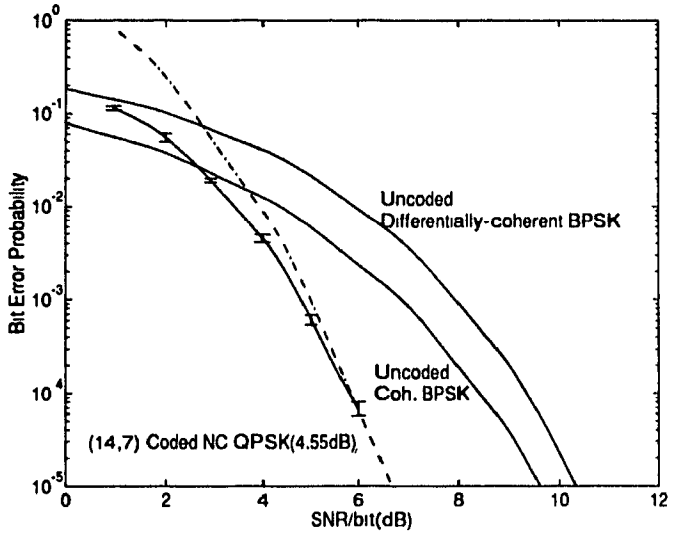

Figure 4.6: A (14,7) Code in $\mathcal{Z}_4$

4.5.1 The Correlated Rayleigh Fading Model

Correlated Rayleigh fading can modeled as a correlated complex gaussian process, u_l, which multiplies the transmitted symbols. The received symbols, y_l, as in (2.13), are therefore given by

$$y_l = Tu_l f_{il} + n_l \quad l = 0, \cdots, N-1. \tag{4.23}$$

It should be noted that the u_l's are statistically independent of the n_l's. In order to express (4.23) in vector form, we create the diagonal matrix $\mathbf{F}_m$, whose main diagonal is the vector $\mathbf{f}_m$, so that the received vector may be expressed as

$$\mathbf{r} = T\mathbf{u}\mathbf{F}_m + \mathbf{n} \tag{4.24}$$

where $\mathbf{u}$ is the vector made up of the u_l's, $\mathbf{n}$ is as in (2.14), and T is the symbol duration. The power spectrum of the u_l's may take on various forms to model different situations. Here we will use the *land–mobile* model for the power spectrum which has the following form

$$S_u(f) = \begin{cases} \dfrac{1}{\sqrt{\pi^2(f^2 - f_D^2)}} & |f| \le f_D \\ 0 & |f| > f_D \end{cases}. \tag{4.25}$$

The constant f_D is known as the *Doppler frequency*, and the product $f_D T$ as the *fade rate.* The shape of $S_u(f)$ is shown in Fig. 4.7. The autocorrelation function corresponding to $S_u(f)$, $\phi_{uu}(m)$, is given by

$$\phi_{uu}(i) = \frac{1}{2}E(u_l^* u_{l+i}) = J_0(2\pi i f_D T), \tag{4.26}$$

where $J_0(\cdot)$ is the zero-order bessel function. Since this is a general gaussian detection problem, it is shown in [23, p. 98] that the ML decoding rule is the following minimization

$$\min_{m=1,2,\cdots,|\mathcal{C}|} \mathbf{r}\mathbf{\Phi}_m^{-1}\mathbf{r}^* \tag{4.27}$$

where $\mathbf{\Phi}_m$ is the autocorrelation matrix for $\mathbf{r}$ assuming the m^{th} codeword was transmitted. This matrix is expressed as

$$\mathbf{\Phi}_m = \frac{1}{2}E(\mathbf{r}^*\mathbf{r}) = \mathbf{F}_m^*(\mathbf{\Phi}_{uu} + \frac{1}{\gamma}\mathbf{I})\mathbf{F_m}, \tag{4.28}$$

where γ is the SNR and $\mathbf{\Phi}_{uu}$ is the autocorrelation matrix of $\mathbf{u}$, whose elements are given by

$$\mathbf{\Phi}_{uu}(i,j) = \phi_{uu}(|i-j|). \tag{4.29}$$

The decoding rule is therefore as follows,

$$\min_{m=1,2,\cdots,|\mathcal{C}|} \mathbf{r}\mathbf{F_m}(\mathbf{\Phi}_{uu} + \frac{1}{\gamma}\mathbf{I})^{-1}\mathbf{F_m}^*\mathbf{r}^*. \tag{4.30}$$

It turns out for a flat fading channel (ie. when $f_D T = 0$) that (4.30) is equivalent to the decision rule for the AWGN channel [24].

4.5.2 Error performance

The error performance of non-coherent block detection over correlated Rayleigh fading channels is treated in [22] for uncoded systems. In this work an exact expression for the pairwise error event probability, $P(\mathbf{c}_0 \rightarrow \mathbf{c}_m)$, is given. We will use the same expression to generate a union bound for two of the codes of the previous chapter. It also assumed

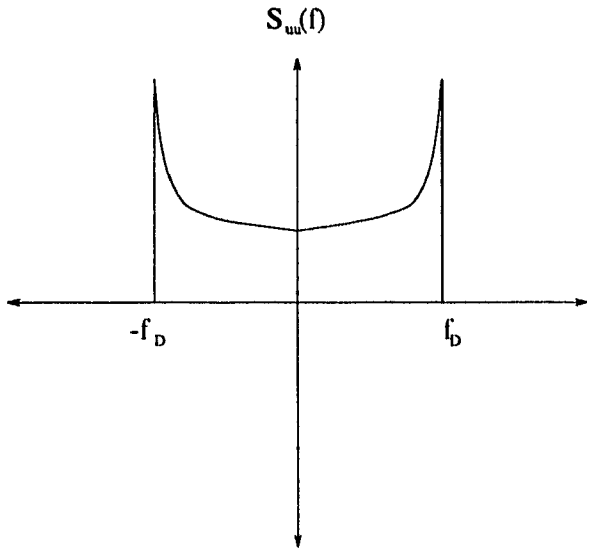

Figure 4.7: Power spectrum of the land-mobile fading model

in [22] that the all-zero codeword may be used to assess the error performance without any loss of generality. This follows directly from the form of the decoding rule in (4.30).

As was the case for the gaussian channel, more than just the minimum-distance codewords were used for the evaluation of the union bound. Moreover, the codewords with the same d^2_{NC} cannot be considered as being equivalent over these channels, in terms of $P(\mathbf{c}_0 \rightarrow \mathbf{c}_m)$, since the performance criterion is no longer d^2_{NC}. Not surprisingly, $P(\mathbf{c}_0 \rightarrow \mathbf{c}_m)$ is somewhat dependent on the location of symbols within a codeword. We had to therefore evaluate $P(\mathbf{c}_0 \rightarrow \mathbf{c}_m)$ for each of the codewords within a group sharing the same d^2_{NC} seperately in order to be as accurate as possible in calculating the union bound.

In Fig.4.8 we show the performance of uncoded binary DPSK and the union bound on the performance of the (8,4) and (14,7) BPSK equivalent codes in $\mathcal{Z}_4$, whose generator matrices are given in (4.21) and (4.22), for fade rates of $f_D T = .001, .01$ and $.1$. We see that as the fade rate increases there is a diversity effect, since the symbols within a codeword become less correlated. Consequently, the performance of the coded system improves with increasing fade rates while that of the uncoded system degrades. For the

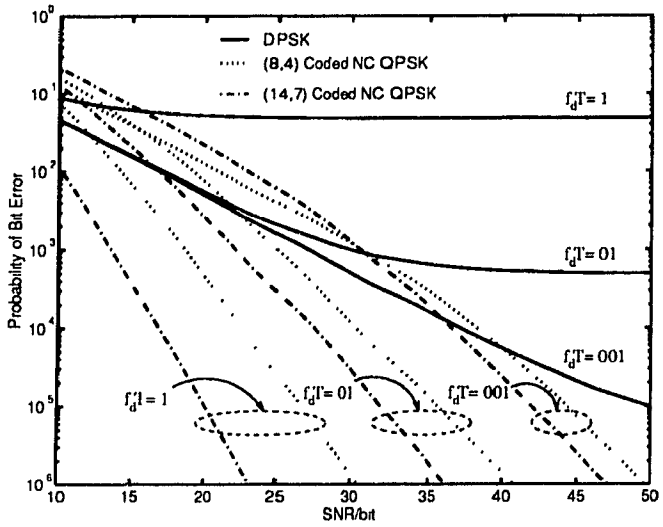

Figure 4.8: Union bound on the performance of two BPSK equivalent codes in $\mathcal{Z}_4$ for fade rates of $f_DT = .1, .01$ and $.001$

slow fading case ($f_DT = .001$) we see that the union bounds are worse than the uncoded system for error rates less than 10^{-4}.

In order to alleviate the situation at very low fade rates, we may wish to use some symbol interleaving. Let us assume that we use an interleaver with depth D_i. This means that the symbols of the codeword will be spaced D_i transmitted symbols apart, and we may therefore look at this as artificially increasing the fade rate by a factor of D_i, as far as the codeword is concerned. More precisely, the matrix Φ_{uu} is calculated using a fade rate of f_DTD_i, rather than f_DT. For both codes in Fig.4.8 at $f_DT = .001$, the performance is quite poor for low error rates. If, however, we use only 10 symbol interleaving, which is rather small, we have an effective fade rate of $f_DTD_i = .01$, which yields a significant improvement. The amount of interleaving required depends, of course, on the fade rates experienced in the environment. We see, however, that even the rather simple (8,4) code performs quite well at moderate fade rates without interleaving and at slow fade rates with only a small amount of interleaving.

We will now present some simulation results of the reduced–complexity decoding strategy presented in the first part of this chapter over correlated fading channels. First of all, to simulate correlated fading channels, we use independent complex gaussian random variables obtained from a pseudo–random sequence generator as input to an FIR filter whose magnitude response closely approximates $\sqrt{S_u(f)}$. The output of this filter is then the process u_l. The filter was designed for a fade rate of $f_D T = .1$ using 512 coefficients. The tap length of the filter has to be long because of the sharp cutoff in the power spectrum of the process. In order to obtain lower fade rates, we linearly interpolate the output of the filter. This was necessary since we found that for lower fade rates the length of the filter had to be much longer to closely approximate the desired magnitude response, and would therefore significantly increase simulation times. In Fig.4.9 we show the ideal autocorrelation function and that of our simulation (averaged using a window of 1000 symbols) for a fade rate of $f_D T = .01$. We see that if we use codewords of moderate length, the autocorrelation function of the simulation closely matches that of the ideal model. In Fig.4 10 we show the magnitude and phase of a

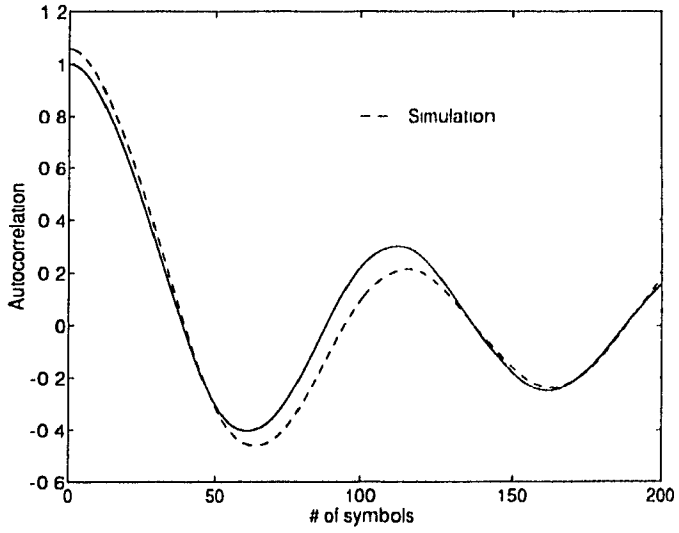

Figure 4.9: Comparison of ideal and simulation autocorrelation functions

typical fading process spanning many symbols, whereas Fig.4.11 shows the variation

over a small number of symbols. In a span of ten or so symbols, we see that both the amplitude and phase can vary significantly.

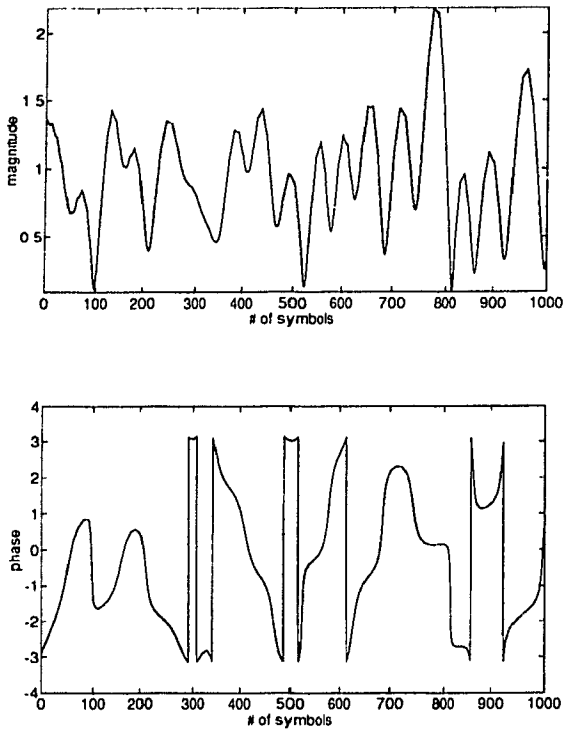

Figure 4.10: Magnitude and phase of the fading process over many symbols ($f_DT = .10$)

The (8,4) and (14,7) bandwidth efficient codes in $\mathcal{Z}_4$ have both been simulated over a correlated Rayleigh fading channel with $f_DT = .01$, and the (8,4) code over a channel with $f_DT = .1$ as well. Both have been decoded using the reduced-complexity decoding method of the previous section with $L = K$. Figs.4.12–4.14 show the results of the simulations. In each figure, we compare the simulation result with the union bound for a ML decoder for that code, the result of a simulation for a differentially–coherent system ($M = 2$) over the same channel and the performance of ideal coherent BPSK over a flat Rayleigh fading channel ($f_DT = 0$). We see that the simulation of the differentially–coherent systems match the analytical curves in Fig.4.8 exactly,

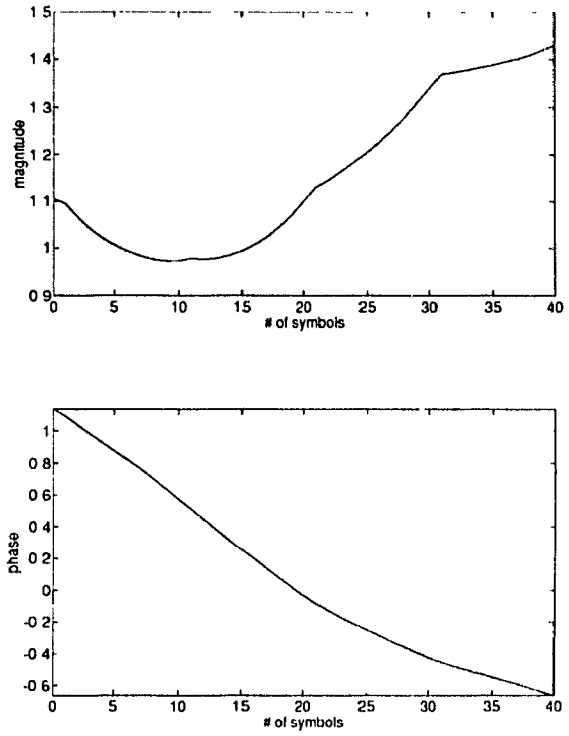

Figure 4.11: Magnitude and phase of the fading process over few symbols ($f_D T = .10$)

verifying the correctness of our channel simulation. In each case, we see that the union bound is quite pessimistic for low SNR, and that the slopes of the coded curves are much steeper than the corresponding curve for coherent detection. This accounts for the significant improvement in performance. It is interesting to note that the two codes with $f_D T = .01$ attain the same slope for two orders of magnitude in P_b (between 10^{-3} and 10^{-5}). For both codes at $f_D T = .01$, we also notice that the irreducible error-floor of the differentially–coherent system is completely eliminated (at least down to $P_b = 10^{-5}$). For the (8,4) code with $f_D T = .1$, which represents a fairly high fade rate, significant performance improvement is obtained even at fairly high error–rates. We do see, however, that the reduced–complexity decoding strategy breaks down at low error rates. This can be attributed to the first stage of decoding which uses differential

detection and results in an irreducible error floor around $P_b = 10^{-6}$. It is very important

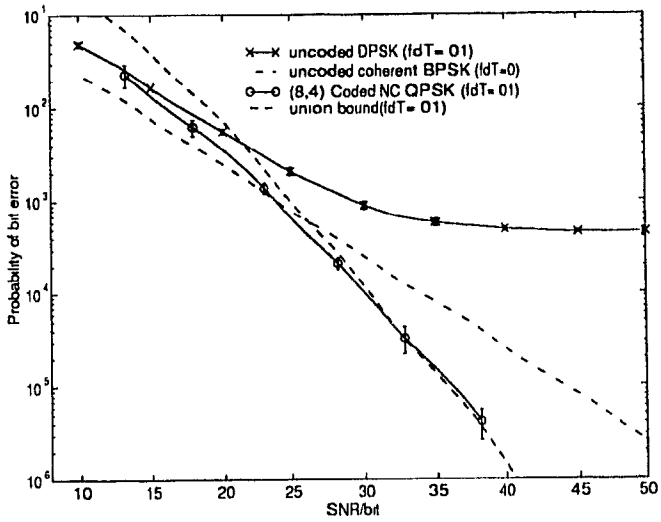

Figure 4.12: Simulation results for an (8,4) code in $\mathcal{Z}_4$ ($f_D T = .01$)

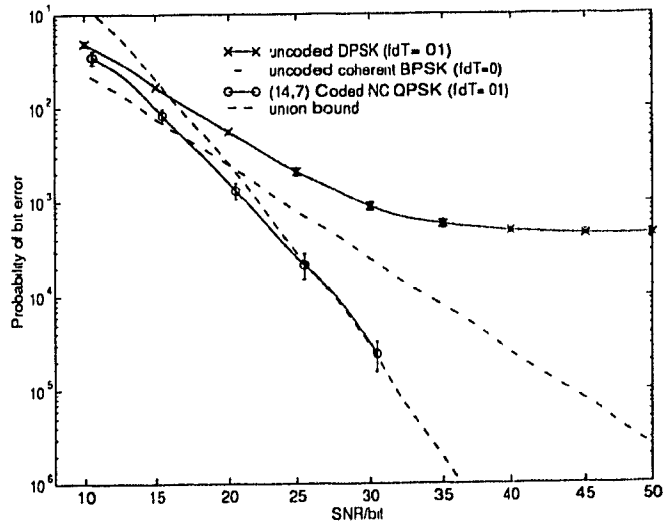

Figure 4.13: Simulation results for a (14,7) code in $\mathcal{Z}_4$ ($f_D T = .01$)

to point out that, in each case, these performance enhancements are obtained without the use of symbol interleaving. Most other coding schemes for fading channels with fade

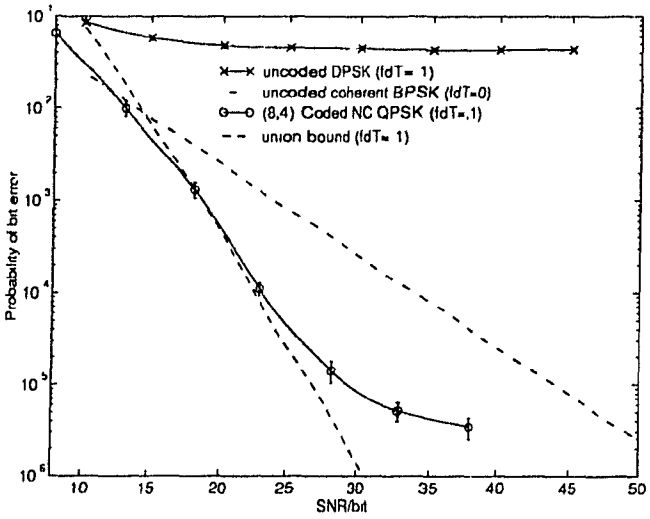

Figure 4.14: Simulation results for an (8,4) code in $\mathcal{Z}_4$ ($f_D T = .1$)

rates as high as $f_D T = .01$, and sometimes higher, use some interleaving, see for instance [7, Chap.9].

Chapter 5

Conclusion

In this thesis, we addressed the problem of channel coding for the non–coherent AWGN channel. We have generalized non–coherent block detection of MPSK as a coding problem, and presented several examples of codes which achieve significant coding gain over uncoded coherent MPSK.

Using a coding framework, a review of non–coherent block detection of MPSK was presented. We defined a distance measure for non–coherent block detection, which was used primarily as a benchmark for comparison and in the design of codes . We then investigated a class of block codes called *module–phase codes* which are well matched to MPSK. These codes have a rich algebraic structure, as they are based on elements of module theory and in many ways resemble traditional linear block codes.

A method for constructing module–phase codes for the non–coherent distance measure was introduced. It was shown that *differential–encoding* , when considered on a block basis, is a particular example of a class of module–phase codes that approaches the performance of uncoded coherent MPSK as the block length is increased. Examples of more powerful codes which achieve significant coding gain over uncoded coherent MPSK were presented. The coding gain is achieved in one case at the expense of bandwidth expansion, and in the other case at the expense of signal constellation expansion.

The performance of these codes becomes even more impressive when compared with traditional differentially–coherent detection of MPSK.

The issue of reduced–complexity/sub–optimal decoding was then addressed. A decoding/demodulation strategy was presented which uses a combination of *information set decoding* and a sub–optimal method proposed for non–coherent block detection of differentially–encoded MPSK. It was found that codes which share the same distance properties, and are therefore equivalent for ML decoding, are not all necessarilly well–suited for this decoding technique. Consequently, it was necessary, in some cases, to search for equivalent codes which were more amenable to this type of decoding. While this strategy significantly reduces arithmetic complexity compared with an exhaustive ML decoder, computer simulations for various codes indicate that very little performance, if any, is sacrificed.

Finally, we present some results on the performance of these codes combined with the reduced–complexity decoding method over correlated Rayleigh fading channels, again through the use of computer simulations. We have shown that it is possible to achieve significant performance enhancement compared with ideal coherent detection and even more so compared with differentially–coherent detection. It should also be noted these performance improvements, in some cases, are attained *without* symbol interleaving, which is characteristic of most other coding systems over fading channels.

Appendix A

Computer Searches and Code Descriptions

A.1 Computer Searches

A.1.1 Searching for h–vectors

Here we will briefly outline the methods used for selecting the h–vectors for the construction method proposed in Chapter 3. We start by identifying the codewords which must be excluded from the code, assuming that codeword overlapping is performed, and knowing the desired parameters for the code (ie. N,K,M, and d^2_{NC}). In order to reduce the running time of the search, we start by searching for the vector, $\mathbf{h}_{N-K}$, since it has the smallest number of components, and then $\mathbf{h}_{N-K-1}$ until we reach $\mathbf{h}_1$. Once we reach $\mathbf{h}_1$, if no vector is found which removes the remaining codewords, we start at the beginning with a smaller set of target codewords (ie. a code with a smaller d^2_{NC}.) If N,K, and the set of vectors to be removed are not too large, we may search exhaustively for the h–vectors. Otherwise, we search randomly, with a pseudo–random number generator, for an h–vector which maximizes the number of codewords excluded from the code until, after a specified number of repetitions, no improvement is found.

Aside from shorter running times, the random approach has another advantage. The exhaustive search will find an h–vector which maximizes the number of codewords from the target set which are excluded but, in general, there are many such vectors. Some of them may be better choices when combined with the other h–vectors. The random approach, if attempted several times using a different seed for the pseudo–random generator, will have the possibility of choosing these better vectors. We have found that better codes sometimes result by using the random approach, and at the same time require much less running time.

A.1.2 Searching for G_c

If N,K are quite small, we may search for optimal generator matrices, G_c, by searching exhaustively through all the possible combinations of its elements and calculating its d^2_{NC}. This was done for some of the simple bandwidth-efficient BPSK equivalent codes. For the BPSK equivalent codes in higher-order rings, the search was done by searching randomly for possible generator matrices until, after a specified number of iterations, no improvement in the minimum distance of the code was found. This is, of course, an incomplete search and the resulting codes are not necessarily optimum.

A.2 Code Descriptions for Bandwidth–Expanding Codes

In this section we present a more complete description of the bandwidth-expanding codes which were presented in the Tables 3.2 and 3.3 of Chapter 3. For each code we present the **h**-vectors as well as the generator matrix which describe it, a portion of its distance profile with the corresponding average number of bit errors for each of the distances, and its asymptotic coding gain.

A (3,2) Code in $\mathcal{Z}_4$

$\mathbf{h}_1 = (1 \quad 1 \quad 1)$

$$\mathbf{G_c} = \begin{pmatrix} 1 & 0 & 3 \\ 0 & 1 & 3 \end{pmatrix}$$

d^2_{NC}	Weight	$\overline{e_b}$
2.00	12	2.00
4.00	3	2.67

Coding Gain: 1.25dB over uncoded coherent QPSK

A (6,5) Code in $\mathcal{Z}_4$

$\mathbf{h}_1 = (1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1)$

$$\mathbf{G_c} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 3 \\ 0 & 1 & 0 & 0 & 0 & 3 \\ 0 & 0 & 1 & 0 & 0 & 3 \\ 0 & 0 & 0 & 1 & 0 & 3 \\ 0 & 0 & 0 & 0 & 1 & 3 \end{pmatrix}$$

d^2_{NC}	Weight	$\overline{e_b}$
2.00	61	2.62
2.88	42	6.19
3.39	120	3.33
4.00	135	3.33
4.76	360	5.56

Coding Gain: 2.22dB over uncoded coherent QPSK

A (7,6) Code in $\mathcal{Z}_4$

$\mathbf{h_1} = (1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1)$

$$\mathbf{G_c} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 3 \\ 0 & 1 & 0 & 0 & 0 & 0 & 3 \\ 0 & 0 & 1 & 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 1 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 & 1 & 0 & 3 \\ 0 & 0 & 0 & 0 & 0 & 1 & 3 \end{pmatrix}$$

d^2_{NC}	Weight	$\overline{e_b}$
2.000	56	3.000
2.343	70	3.429
3.528	336	4.714
4.000	448	5.143

Coding Gain: 2.34dB over uncoded coherent QPSK

A (7,4) Code in $\mathcal{Z}_4$

$\mathbf{h_1} = (3 \quad 2 \quad 1 \quad 2 \quad 0 \quad 0 \quad 1)$

$\mathbf{h_2} = (3 \quad 2 \quad 1 \quad 0 \quad 0 \quad 1)$

$\mathbf{h_3} = (3 \quad 1 \quad 1 \quad 1 \quad 1)$

$$\mathbf{G_c} = \begin{pmatrix} 1 & 0 & 0 & 0 & 3 & 1 & 1 \\ 0 & 1 & 0 & 0 & 3 & 2 & 2 \\ 0 & 0 & 1 & 0 & 3 & 3 & 3 \\ 0 & 0 & 0 & 1 & 3 & 0 & 2 \end{pmatrix}$$

d^2_{NC}	Weight	$\overline{e_b}$
3.528	44	3.364
4.000	20	2.800
5.172	40	4.200
6.000	116	4.241

Coding Gain: 3.04dB over uncoded coherent QPSK

A (9,5) Code in $\mathcal{Z}_4$

$\mathbf{h_1} = (3 \quad 0 \quad 1 \quad 2 \quad 2 \quad 0 \quad 0 \quad 0 \quad 1)$

$\mathbf{h_2} = (1 \quad 3 \quad 2 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1)$

$\mathbf{h_3} = (3 \quad 1 \quad 1 \quad 0 \quad 1 \quad 0 \quad 1)$

$\mathbf{h_4} = (1 \quad 1 \quad 1 \quad 1 \quad 0 \quad 1)$

$$\mathbf{G_c} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 3 & 1 & 3 & 1 \\ 0 & 1 & 0 & 0 & 0 & 3 & 3 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 3 & 3 & 2 & 3 \\ 0 & 0 & 0 & 1 & 0 & 3 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 1 & 0 & 3 & 0 & 2 \end{pmatrix}$$

d^2_{NC}	Weight	$\overline{e_b}$
4.000	17	2.118
4.169	24	3.500
4.343	20	5.200
4.901	40	4.300
5.528	72	4.111
5.757	40	4.900
6.000	56	5.143

Coding Gain: 3.47dB over uncoded coherent QPSK

A (10,9) Code in $\mathcal{Z}_4$

$\mathbf{h_1} = (1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1)$

$$\mathbf{G_c} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 3 \end{pmatrix}$$

d^2_{NC}	Weight	$\overline{e_b}$
2.000	91	1.978
2.456	90	7.200
2.938	530	5.026
3.720	720	3.600
4.000	1395	4.413
4.292	1680	7.200

Coding Gain: 2.55dB over uncoded coherent QPSK

A (10,7) Code in $\mathcal{Z}_4$

$\mathbf{h_1} = (1 \quad 0 \quad 1 \quad 0 \quad 2 \quad 1 \quad 1 \quad 0 \quad 0 \quad 1)$

$\mathbf{h}_2 = (2 \quad 2 \quad 1 \quad 1 \quad 0 \quad 0 \quad 2 \quad 0 \quad 1)$

$\mathbf{h}_3 = (1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 0 \quad 1)$

$$\mathbf{G_c} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 2 & 3 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 3 & 2 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 3 & 3 & 3 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 3 & 3 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 3 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 3 & 0 & 3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 2 & 3 \end{pmatrix}$$

d^2_{NC}	Weight	$\overline{e_b}$
2.938	36	3.556
3.190	34	3.882
3.720	38	5.000
4.000	82	4.780
4.292	84	5.548
4.597	174	4.805

Coding Gain: 3.13dB over uncoded coherent QPSK

A (10,6) Code in $\mathcal{Z}_4$

$\mathbf{h}_1 = (3 \quad 1 \quad 3 \quad 2 \quad 3 \quad 2 \quad 3 \quad 1 \quad 3 \quad 1)$

$\mathbf{h}_2 = (1 \quad 1 \quad 1 \quad 3 \quad 1 \quad 1 \quad 3 \quad 1 \quad 1)$

$\mathbf{h}_3 = (3 \quad 3 \quad 1 \quad 2 \quad 2 \quad 3 \quad 2 \quad 1)$

$\mathbf{h}_4 = (3 \quad 3 \quad 3 \quad 3 \quad 3 \quad 1 \quad 1)$

$$\mathbf{G_c} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 3 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 3 & 1 & 2 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 3 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 2 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 & 1 & 3 & 3 & 3 & 1 \end{pmatrix}$$

d^2_{NC}	Weight	$\overline{e_b}$
3.720	44	4.227
4.000	29	4.828
4.292	8	1.750
4.597	62	4.258
4.917	22	4.091
5.615	182	5.484

Coding Gain: 3.49dB over uncoded coherent QPSK

An (11,10) Code in $\mathcal{Z}_4$

$\mathbf{h}_1 = (1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1)$

$$G_c = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 3 \end{pmatrix}$$

d^2_{NC}	Weight	$\overline{e_b}$
2.000	132	3.333
3.056	990	4.848
3.754	1320	5.758
4.000	3036	6.061
4.789	11088	6.970
5.675	31680	7.879
6.000	24420	8.182

Coding Gain: 2.60dB over uncoded coherent QPSK

An (11,8) Code in $\mathcal{Z}_4$

$h_1 = (0 \quad 1 \quad 0 \quad 1 \quad 0 \quad 1 \quad 1 \quad 1 \quad 0 \quad 0 \quad 1)$

$h_1 = (3 \quad 2 \quad 2 \quad 1 \quad 1 \quad 0 \quad 2 \quad 1 \quad 0 \quad 1)$

$h_1 = (1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 0 \quad 0 \quad 1)$

$$G_c = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 2 & 3 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 3 & 2 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 3 & 3 & 3 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 3 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 3 & 0 & 3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 2 & 3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 3 & 3 \end{pmatrix}$$

d^2_{NC}	Weight	$\overline{e_b}$
2.945	38	2.842
3.056	6	5.333
3.398	40	4.800
3.515	32	4.750
3.754	56	5.643
4.000	112	4.000
4.384	120	5.467

Coding Gain: 3.31dB over uncoded coherent QPSK

An (11,7) Code in $\mathcal{Z}_4$

$h_1 = (0 \quad 0 \quad 0 \quad 0 \quad 2 \quad 0 \quad 3 \quad 3 \quad 3 \quad 2 \quad 1)$

$\mathbf{h_2} = (2 \quad 2 \quad 0 \quad 2 \quad 0 \quad 0 \quad 3 \quad 1 \quad 0 \quad 1)$

$\mathbf{h_3} = (1 \quad 0 \quad 1 \quad 3 \quad 1 \quad 1 \quad 0 \quad 3 \quad 1)$

$\mathbf{h_4} = (1 \quad 1 \quad 3 \quad 1 \quad 1 \quad 1 \quad 3 \quad 1)$

$$\mathbf{G_c} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 2 & 3 & 3 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 3 & 3 & 3 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 3 & 0 & 3 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 3 & 2 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 3 & 2 & 1 & 3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 3 \end{pmatrix}$$

d^2_{NC}	Weight	$\overline{e_b}$
3.515	6	3.667
3.754	48	4.250
4.000	38	4.947
4.789	176	4.932
5.675	500	5.480
6.000	336	5.464
6.343	464	5.759
7.528	2704	6.655

Coding Gain: 3.50dB over uncoded coherent QPSK

An (11,6) Code in $\mathcal{Z}_4$

$\mathbf{h_1} = (2 \quad 2 \quad 1 \quad 1 \quad 0 \quad 1 \quad 0 \quad 3 \quad 2 \quad 1 \quad 1)$

$\mathbf{h_2} = (2 \quad 3 \quad 0 \quad 0 \quad 0 \quad 1 \quad 2 \quad 2 \quad 2 \quad 1)$

$\mathbf{h_3} = (0 \quad 0 \quad 3 \quad 2 \quad 1 \quad 3 \quad 1 \quad 2 \quad 1)$

$\mathbf{h_4} = (3 \quad 0 \quad 2 \quad 3 \quad 1 \quad 0 \quad 0 \quad 1)$

$\mathbf{h_5} = (1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 3 \quad 1)$

$$\mathbf{G_c} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 3 & 1 & 3 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 3 & 0 & 1 & 1 & 3 \\ 0 & 0 & 1 & 0 & 0 & 0 & 3 & 2 & 2 & 2 & 3 \\ 0 & 0 & 0 & 1 & 0 & 0 & 3 & 1 & 1 & 2 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 3 & 3 & 2 & 0 & 3 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 2 \end{pmatrix}$$

d^2_{NC}	Weight	$\overline{e_b}$
4.384	14	3.333
4.789	28	4.848
4.929	42	5.758
5.675	72	6.061
6.000	48	6.970
6.169	108	7.879
6.343	52	8.182

Coding Gain: 3.79dB over uncoded coherent QPSK

A (5,4) Code in $\mathcal{Z}_8$

$\mathbf{h}_1 = (1 \quad 1 \quad 1 \quad 1 \quad 1)$

$$\mathbf{G_c} = \begin{pmatrix} 1 & 0 & 0 & 0 & 7 \\ 0 & 1 & 0 & 0 & 7 \\ 0 & 0 & 1 & 0 & 7 \\ 0 & 0 & 0 & 1 & 7 \end{pmatrix}$$

d^2_{NC}	Weight	$\overline{e_b}$
0.586	20	1.600
0.969	20	6.400
1.172	30	3.200
1.474	40	5.600
1.528	10	6.400
1.566	60	3.200

Coding Gain: 2.04dB over uncoded coherent 8–PSK

A (6,5) Code in $\mathcal{Z}_8$

$\mathbf{h}_1 = (1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1)$

$$\mathbf{G_c} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 7 \\ 0 & 1 & 0 & 0 & 0 & 7 \\ 0 & 0 & 1 & 0 & 0 & 7 \\ 0 & 0 & 0 & 1 & 0 & 7 \\ 0 & 0 & 0 & 0 & 1 & 7 \end{pmatrix}$$

d^2_{NC}	Weight	$\overline{e_b}$
0.586	30	1.667
0.838	30	6.667
1.172	90	3.333
1.285	12	5.833
1.419	120	6.667
1.570	120	3.333
1.757	20	5.000

Coding Gain: 2.22dB over uncoded coherent 8–PSK

A (7,6) Code in $\mathcal{Z}_8$

$\mathbf{h}_1 = (1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1)$

$$\mathbf{G_c} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 7 \\ 0 & 1 & 0 & 0 & 0 & 0 & 7 \\ 0 & 0 & 1 & 0 & 0 & 0 & 7 \\ 0 & 0 & 0 & 1 & 0 & 0 & 7 \\ 0 & 0 & 0 & 0 & 1 & 0 & 7 \\ 0 & 0 & 0 & 0 & 0 & 1 & 7 \end{pmatrix}$$

d^2_{NC}	Weight	$\overline{e_b}$
0.586	56	3.000
1.172	420	5.143
1.572	336	5.143
1.757	560	6.429
2.000	56	5.571

Coding Gain: 2.34dB over uncoded coherent 8-PSK

A (7,5) Code in $\mathcal{Z}_8$

$\mathbf{h}_1 = (0 \quad 3 \quad 2 \quad 1 \quad 2 \quad 0 \quad 1)$

$\mathbf{h}_2 = (2 \quad 1 \quad 1 \quad 1 \quad 0 \quad 1)$

$$\mathbf{G_c} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 6 & 0 \\ 0 & 1 & 0 & 0 & 0 & 7 & 5 \\ 0 & 0 & 1 & 0 & 0 & 7 & 6 \\ 0 & 0 & 0 & 1 & 0 & 7 & 7 \\ 0 & 0 & 0 & 0 & 1 & 0 & 6 \end{pmatrix}$$

d^2_{NC}	Weight	$\overline{e_b}$
0.844	12	2.667
1.027	4	5.000
1.129	4	6.000
1.172	22	3.909
1.287	4	1.000
1.426	24	4.667
1.508	8	3.250

Coding Gain: 3.13dB over uncoded coherent 8-PSK

An (8,7) Code in $\mathcal{Z}_8$

$\mathbf{h}_1 = (3 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1)$

$$\mathbf{G_c} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 5 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 7 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 7 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 7 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 7 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 7 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 7 \end{pmatrix}$$

d^2_{NC}	Weight	$\overline{e_b}$
0.586	42	1.714
0.606	42	5.286
0.848	14	6.857
1.045	70	3.571
1.106	14	7.000
1.172	560	5.250
1.203	56	5.821
1.287	2	8.000
1.431	210	6.857

Coding Gain: 3.34dB over uncoded coherent 8–PSK

An (8,6) Code in $\mathcal{Z}_8$

$\mathbf{h_1} = (0 \quad 3 \quad 2 \quad 1 \quad 3 \quad 2 \quad 0 \quad 1)$

$\mathbf{h_2} = (2 \quad 1 \quad 1 \quad 1 \quad 0 \quad 0 \quad 1)$

$$\mathbf{G_c} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 6 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 7 & 5 \\ 0 & 0 & 1 & 0 & 0 & 0 & 7 & 6 \\ 0 & 0 & 0 & 1 & 0 & 0 & 7 & 7 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 5 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 6 \end{pmatrix}$$

d^2_{NC}	Weight	$\overline{e_b}$
0.848	16	2.750
1.045	18	6.000
1.106	2	3.000
1.172	42	3.762
1.203	2	3.000
1.225	2	4.000
1.287	6	3.000

Coding Gain: 3.34dB over uncoded coherent 8–PSK

A (9,8) Code in $\mathcal{Z}_8$

$\mathbf{h_1} = (3 \quad 2 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1)$

$$\mathbf{G_c} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 5 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 6 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 7 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 7 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 7 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 7 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 7 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 7 \end{pmatrix}$$

d^2_{NC}	Weight	$\overline{e_b}$
0.586	42	1.714
0.635	14	6.143
0.730	42	5.286
0.761	70	4.571
0.851	70	4.171
0.945	2	16.000
1.059	126	6.317

Coding Gain: 2.50dB over uncoded coherent 8–PSK

A (9,7) Code in $\mathcal{Z}_8$

$\mathbf{h_1} = (5 \quad 4 \quad 3 \quad 2 \quad 1 \quad 3 \quad 2 \quad 0 \quad 1)$

$\mathbf{h_2} = (1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 0 \quad 0 \quad 1)$

$$\mathbf{G_c} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 7 & 3 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 7 & 4 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 7 & 5 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 7 & 6 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 7 & 7 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 5 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 6 \end{pmatrix}$$

d^2_{NC}	Weight	$\overline{e_b}$
0.851	32	3.062
1.059	32	4.438
1.127	2	3.000
1.172	58	4.379
1.205	18	7.444
1.246	2	3.000
1.286	6	10.667
1.288	2	1.000
1.435	124	5.565
1.527	32	4.312

Coding Gain: 3.54dB over uncoded coherent 8–PSK

A (10,9) Code in $\mathcal{Z}_8$

$\mathbf{h_1} = (1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1)$

$$\mathbf{G_c} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 7 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 7 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 7 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 7 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 7 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 7 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 7 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 7 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 7 \end{pmatrix}$$

d^2_{NC}	Weight	$\overline{e_b}$
0.586	90	1.800
0.659	90	7.200
1.144	20	9.000
1.172	1260	3.600
1.231	1680	7.200
1.377	420	14.40
1.577	720	3.600

Coding Gain: 2.55dB over uncoded coherent 8–PSK

A (10,8) Code in $\mathcal{Z}_8$

$\mathbf{h}_1 = (2 \quad 1 \quad 0 \quad 1 \quad 0 \quad 3 \quad 2 \quad 1 \quad 0 \quad 1)$

$\mathbf{h}_2 = (5 \quad 5 \quad 5 \quad 4 \quad 4 \quad 1 \quad 1 \quad 1 \quad 1)$

$$\mathbf{G_c} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 6 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 7 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 4 & 7 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 7 & 5 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 7 & 6 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 7 & 7 \end{pmatrix}$$

d^2_{NC}	Weight	$\overline{e_b}$
0.773	10	5.200
0.830	28	4.714
0.854	36	2.389
0.950	2	2.000
1.070	28	3.643
1.172	64	3.781
1.231	34	5.647
1.281	16	5.000
1.334	80	5.775

Coding Gain: 3.25dB over uncoded coherent 8–PSK

A (10,7) Code in $\mathcal{Z}_8$

$\mathbf{h}_1 = (0 \quad 0 \quad 0 \quad 0 \quad 2 \quad 0 \quad 2 \quad 0 \quad 0 \quad 1)$

$\mathbf{h}_2 = (3 \quad 6 \quad 4 \quad 2 \quad 5 \quad 3 \quad 1 \quad 0 \quad 1)$

$\mathbf{h}_3 = (2 \quad 1 \quad 1 \quad 1 \quad 0 \quad 0 \quad 0 \quad 1)$

$$\mathbf{G_c} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 6 & 5 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 7 & 2 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 7 & 4 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 7 & 6 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 3 & 6 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 7 & 6 \end{pmatrix}$$

d^2_{NC}	Weight	$\overline{e_b}$
1.070	18	2.778
1.172	26	3.000
1.231	10	9.600
1.377	8	10.500
1.438	2	9.000
1.533	22	2.455
1.577	14	2.857

Coding Gain: 4.08dB over uncoded coherent 8–PSK

A.3 Code Descriptions for Bandwidth–Efficient Codes

In this section we present more complete descriptions of the codes presented in Table3.4 of Chapter 3. Some of these codes were found either by exhaustive computer search for optimal generator matrices or by an incomplete random search, and no **h**–vectors are given. The remaining codes were found using the construction method of Chapter 3.

A (4,1) Code in $\mathcal{Z}_{16}$

$\mathbf{G_c} = (1 \quad 2 \quad 5 \quad 8)$

d^2_{NC}	Weight	$\overline{e_b}$
2.631	2	1.000
2.764	6	2.333
2.980	2	3.000
3.615	2	2.000
4.000	1	2.000
4.378	2	2.000

Coding Gain: 1.19dB over uncoded coherent BPSK

A (6,3) Code in $\mathcal{Z}_4$

$$\mathbf{G_c} = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 2 \\ 0 & 1 & 0 & 0 & 2 & 1 \\ 0 & 0 & 1 & 2 & 0 & 1 \end{pmatrix}$$

d^2_{NC}	Weight	$\overline{e_b}$
3.394	14	2.429
4.000	12	3.833
4.764	18	2.556
6.000	19	3.423

Coding Gain: 2.30dB over uncoded coherent BPSK

A (6,2) Code in $\mathcal{Z}_8$

$$\mathbf{G_c} = \begin{pmatrix} 1 & 0 & 0 & 3 & 5 & 7 \\ 0 & 1 & 3 & 0 & 5 & 7 \end{pmatrix}$$

d^2_{NC}	Weight	$\overline{e_b}$
4.000	42	2.857
6.000	21	3.429

Coding Gain: 3.00dB over coherent BPSK

A (5,1) Code in $\mathcal{Z}_{32}$

$\mathbf{G_c} = (1 \quad 3 \quad 5 \quad 16 \quad 25)$

d^2_{NC}	Weight	$\overline{e_b}$
3.172	2	2.000
3.277	4	2.500
3.284	4	2.500
3.367	4	1.500
3.551	4	2.500
3.859	4	3.500
4.000	1	2.000
4.966	4	2.500
5.212	4	3.500

Coding Gain: 2.00dB over uncoded coherent BPSK

A (6,1) Code in $\mathcal{Z}_{64}$

$\mathbf{G_c} = (1 \quad 48 \quad 8 \quad 27 \quad 42 \quad 52)$

d^2_{NC}	Weight	$\overline{e_b}$	d^2_{NC}	Weight	$\overline{e_b}$
3.539	2	1.000	3.971	2	2.000
3.598	2	2.333	4.960	3	2.000
3.633	2	3.000	4.022	2	2.000
3.786	2	2.000	4.217	2	2.000
3.794	2	2.000	4.282	2	2.000
3.818	2	2.000	4.297	2	2.000
3.865	2	2.000	4.377	2	2.000
3.918	2	2.000	4.623	2	2.000
3.960	4	2.000	4.723	2	2.000

Coding Gain: 2.48dB over uncoded coherent BPSK

A (7,1) Code in $\mathcal{Z}_{128}$

$\mathbf{G_c} = (1 \quad 105 \quad 41 \quad 78 \quad 93 \quad 96 \quad 98)$

d^2_{NC}	Weight	$\overline{e_b}$	d^2_{NC}	Weight	$\overline{e_b}$
3.787	2	3.000	4.245	2	3.000
3.832	2	2.000	4.247	2	6.000
3.834	2	3.000	4.303	2	5.000
3.839	2	2.000	4.304	2	2.000
3.911	2	5.000	4.358	2	3.000
3.985	2	4.000	4.404	2	4.000
3.993	2	5.000	4.489	2	4.000
4.000	2	2.000	4.531	2	4.000
4.056	2	4.000	4.565	2	3.000
4.185	2	5.000	4.643	2	2.000
4.214	2	4.000	4.710	2	5.000
4.241	2	1.000	4.844	2	2.000

Coding Gain: 2.77dB over uncoded coherent BPSK

An (8,1) Code in $\mathcal{Z}_{256}$

$\mathbf{G_c} = (1 \quad 190 \quad 188 \quad 26 \quad 19 \quad 153 \quad 143 \quad 100)$

d^2_{NC}	Weight	$\overline{e_b}$	d^2_{NC}	Weight	$\overline{e_b}$
4.019	2	4.000	4.521	2	2.000
4.164	2	3.000	4.522	2	5.000
4.169	2	3.000	4.563	2	5.000
4.211	2	4.000	4.571	2	5.000
4.253	2	4.000	4.640	2	3.000
4.254	2	4.000	4.653	2	3.000
4.390	2	4.000	4.746	2	5.000
4.395	2	3.000	4.775	2	4.000
4.440	2	4.000	4.797	2	5.000
4.455	2	3.000	4.800	2	3.000
4.469	2	6.000	4.807	2	4.000
4.514	2	6.000	4.822	2	5.000

Coding Gain: 3.03dB over uncoded coherent BPSK

An (8,4) Code in $\mathcal{Z}_4$

$\mathbf{h}_1 = (1 \quad 1 \quad 1 \quad 2 \quad 0 \quad 0 \quad 0 \quad 1)$

$\mathbf{h}_2 = (3 \quad 1 \quad 1 \quad 0 \quad 0 \quad 0 \quad 1)$

$\mathbf{h}_3 = (1 \quad 1 \quad 2 \quad 0 \quad 0 \quad 1)$

$\mathbf{h}_4 = (3 \quad 3 \quad 1 \quad 1 \quad 1)$

$$\mathbf{G_c} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 3 & 1 & 3 \\ 0 & 1 & 0 & 0 & 1 & 3 & 3 & 3 \\ 0 & 0 & 1 & 0 & 3 & 2 & 3 & 3 \\ 0 & 0 & 0 & 1 & 3 & 0 & 0 & 2 \end{pmatrix}$$

d^2_{NC}	Weight	$\overline{e_b}$
4.000	26	3.000
4.877	18	3.111
5.394	44	4.000
6.000	28	4.500

Coding Gain: 3.01dB over uncoded coherent BPSK

A (10,5) Code in $\mathcal{Z}_4$

$\mathbf{h}_1 = (2 \quad 1 \quad 1 \quad 0 \quad 3 \quad 3 \quad 2 \quad 0 \quad 1 \quad 1)$

$\mathbf{h}_2 = (0 \quad 3 \quad 3 \quad 0 \quad 1 \quad 2 \quad 0 \quad 1 \quad 1)$

$\mathbf{h}_3 = (2 \quad 0 \quad 3 \quad 2 \quad 0 \quad 0 \quad 1 \quad 1)$

$\mathbf{h}_4 = (1 \quad 0 \quad 0 \quad 2 \quad 1 \quad 2 \quad 1)$

$\mathbf{h}_5 = (2 \quad 1 \quad 1 \quad 1 \quad 2 \quad 1)$

$$\mathbf{G_c} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 2 & 3 & 3 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 3 & 2 & 2 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 3 & 2 & 3 & 0 & 2 \\ 0 & 0 & 0 & 1 & 0 & 3 & 0 & 2 & 0 & 3 \\ 0 & 0 & 0 & 0 & 1 & 2 & 3 & 1 & 2 & 3 \end{pmatrix}$$

d^2_{NC}	Weight	$\overline{e_b}$
4.597	18	3.778
4.917	26	3.923
5.615	38	4.368
6.000	87	4.253

Coding Gain: 3.61dB over uncoded coherent BPSK

A (12,6) Code in $\mathcal{Z}_4$

$\mathbf{h_1} = (0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 2 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1)$

$\mathbf{h_2} = (2 \quad 3 \quad 1 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1)$

$\mathbf{h_3} = (1 \quad 3 \quad 1 \quad 3 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1)$

$\mathbf{h_4} = (3 \quad 1 \quad 0 \quad 2 \quad 2 \quad 0 \quad 0 \quad 0 \quad 1)$

$\mathbf{h_5} = (3 \quad 2 \quad 2 \quad 1 \quad 0 \quad 0 \quad 0 \quad 1)$

$\mathbf{h_6} = (1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1)$

$$\mathbf{G_c} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 3 & 1 & 1 & 3 & 2 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 3 & 2 & 3 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 3 & 2 & 0 & 3 & 3 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 3 & 3 & 2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 3 & 2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 3 & 0 & 0 & 0 & 3 & 2 \end{pmatrix}$$

d^2_{NC}	Weight	$\overline{e_b}$
4.938	26	3.154
5.190	12	3.667
5.720	30	4.800
6.000	25	3.920
6.292	54	4.630
6.597	68	4.765

Coding Gain: 3.93dB over uncoded coherent BPSK

A (14,7) Code in $\mathcal{Z}_4$

$\mathbf{h_1} = (1 \quad 3 \quad 0 \quad 2 \quad 0 \quad 1 \quad 1 \quad 1 \quad 0 \quad 3 \quad 2 \quad 1 \quad 0 \quad 1)$

$\mathbf{h_2} = (2 \quad 0 \quad 2 \quad 2 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 3 \quad 0 \quad 2 \quad 1)$

$\mathbf{h_3} = (1 \quad 2 \quad 3 \quad 1 \quad 0 \quad 3 \quad 2 \quad 1 \quad 1 \quad 0 \quad 1 \quad 1)$

$\mathbf{h_4} = (0 \quad 0 \quad 0 \quad 0 \quad 2 \quad 0 \quad 1 \quad 1 \quad 0 \quad 1 \quad 1)$

$\mathbf{h_5} = (2 \quad 3 \quad 2 \quad 1 \quad 2 \quad 0 \quad 3 \quad 3 \quad 1 \quad 1)$

$\mathbf{h_6} = (1 \quad 0 \quad 0 \quad 1 \quad 3 \quad 2 \quad 3 \quad 3 \quad 1)$

$\mathbf{h_7} = (1 \quad 1 \quad 1 \quad 1 \quad 0 \quad 1 \quad 1 \quad 1)$

$$\mathbf{G_c} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 2 & 3 & 2 & 0 & 1 & 3 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 3 & 3 & 1 & 0 & 0 & 1 & 3 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 3 & 3 & 2 & 3 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 3 & 2 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 2 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 3 & 1 & 2 & 3 & 2 & 1 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 3 & 0 & 0 & 0 & 3 & 2 & 1 \end{pmatrix}$$

d^2_{NC}	Weight	$\overline{e_b}$
5.566	8	5.750
5.780	42	4.381
6.000	14	4.571
6.456	30	5.867
6.938	168	5.595
7.190	92	5.543
7.720	158	5.734

Coding Gain: 4.45dB over uncoded coherent BPSK

A (9,6) Code in $\mathcal{Z}_8$

$\mathbf{h_1} = (7 \quad 5 \quad 2 \quad 4 \quad 6 \quad 1 \quad 2 \quad 5 \quad 1)$

$\mathbf{h_2} = (2 \quad 4 \quad 7 \quad 7 \quad 3 \quad 4 \quad 5 \quad 1)$

$\mathbf{h_3} = (6 \quad 6 \quad 6 \quad 6 \quad 3 \quad 6 \quad 1)$

$$\mathbf{G_c} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 2 & 4 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 2 & 2 & 5 \\ 0 & 0 & 1 & 0 & 0 & 0 & 2 & 7 & 7 \\ 0 & 0 & 0 & 1 & 0 & 0 & 2 & 7 & 5 \\ 0 & 0 & 0 & 0 & 1 & 0 & 5 & 4 & 4 \\ 0 & 0 & 0 & 0 & 0 & 1 & 2 & 2 & 1 \end{pmatrix}$$

d^2_{NC}	Weight	$\overline{e_b}$	d^2_{NC}	Weight	$\overline{e_b}$
1.286	10	4.200	1.757	8	4.500
1.300	4	4.000	1.772	16	5.000
1.435	20	6.500	1.802	6	4.667
1.527	4	2.000	1.808	14	6.000
1.576	6	6.667	1.837	10	5.800
1.637	12	5.000	1.844	14	5.857
1.701	10	5.000	1.873	4	3.500
1.754	2	2.000	1.876	4	3.000

Coding Gain: 1.09dB over uncoded coherent QPSK

A (12,8) Code in $\mathcal{Z}_8$

$\mathbf{h}_1 = (6 \quad 1 \quad 3 \quad 6 \quad 2 \quad 6 \quad 7 \quad 5 \quad 0 \quad 7 \quad 7 \quad 1)$

$\mathbf{h}_2 = (7 \quad 4 \quad 6 \quad 6 \quad 1 \quad 6 \quad 4 \quad 5 \quad 6 \quad 5 \quad 1)$

$\mathbf{h}_3 = (5 \quad 1 \quad 2 \quad 2 \quad 7 \quad 5 \quad 2 \quad 3 \quad 2 \quad 1)$

$\mathbf{h}_4 = (5 \quad 1 \quad 1 \quad 1 \quad 4 \quad 0 \quad 1 \quad 4 \quad 1)$

$$\mathbf{G_c} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 5 & 6 & 5 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 7 & 1 & 5 & 5 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 7 & 0 & 0 & 3 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 4 & 1 & 2 & 2 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 4 & 1 & 2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 3 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 7 & 0 & 2 & 3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 4 & 5 & 2 & 2 \end{pmatrix}$$

d^2_{NC}	Weight	$\overline{e_b}$	d^2_{NC}	Weight	$\overline{e_b}$
1.494	12	5.333	1.757	10	7.800
1.530	12	5.167	1.822	4	4.000
1.542	2	1.000	1.847	30	6.000
1.578	2	3.000	1.875	4	2.500
1.669	10	7.400	1.906	8	7.000
1.748	2	2.000			

Coding Gain: 1.74dB over uncoded coherent QPSK

A (15,10) Code in $\mathcal{Z}_8$

$\mathbf{h}_1 = (0 \quad 6 \quad 4 \quad 4 \quad 2 \quad 2 \quad 1 \quad 6 \quad 7 \quad 0 \quad 7 \quad 2 \quad 1 \quad 3 \quad 1)$

$\mathbf{h}_2 = (5 \quad 5 \quad 0 \quad 4 \quad 5 \quad 2 \quad 7 \quad 4 \quad 4 \quad 6 \quad 5 \quad 7 \quad 2 \quad 1)$

$\mathbf{h}_3 = (5 \quad 4 \quad 3 \quad 4 \quad 1 \quad 7 \quad 5 \quad 4 \quad 5 \quad 6 \quad 6 \quad 3 \quad 1)$

$\mathbf{h}_4 = (4 \quad 5 \quad 5 \quad 2 \quad 2 \quad 0 \quad 0 \quad 0 \quad 6 \quad 4 \quad 4 \quad 1)$

$\mathbf{h}_5 = (1 \quad 4 \quad 5 \quad 0 \quad 5 \quad 0 \quad 1 \quad 4 \quad 1 \quad 1 \quad 1)$

$$\mathbf{G_c} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 7 & 0 & 1 & 6 & 4 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 3 & 3 & 4 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 7 & 6 & 4 & 7 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 6 & 2 & 6 & 4 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 3 & 2 & 7 & 0 & 6 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 4 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 7 & 4 & 5 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 4 & 0 & 4 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 7 & 6 & 7 & 1 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 7 & 0 & 0 & 7 & 2 \end{pmatrix}$$

d^2_{NC}	Weight	$\overline{e_b}$	d^2_{NC}	Weight	$\overline{e_b}$
1.699	6	5.333	1.890	10	6.400
1.749	12	5.667	1.962	10	7.800
1.757	10	8.600	2.000	2	2.000
1.776	2	4.000	2.032	22	7.364
1.834	2	3.000	2.051	4	5.000
1.876	4	2.500	2.053	24	6.500

Coding Gain: 2.30dB over uncoded coherent QPSK

Bibliography

[1] D. Divsalar and M. K. Simon, "Multiple-symbol differential detection of MPSK", *IEEE Trans. Commun.*, vol. COM-38, pp. 300-308, 1990

[2] S. G. Wilson , J. Freebersyser, C. Marshall , "Multi-symbol detection of M-DPSK, Global Telecomm. Conf., GLOBECOM'89 Dallas, Texas, Nov.27-30, 1989, Conf.Rec.,1692-1697

[3] H. Leib and S. Pasupathy, "Optimal Noncoherent Block Demodulation of Differential Phase Shift Keying (DPSK)", *Archiv für Elektronik und Übertragungstechnik*, Vol. 45,pp. 299-305, 1991

[4] J. Astola, "Convolutional Codes for Phase-Modulated Channels",*Cybernetics and Systems: An International Journal*, 17:89-101, 1986

[5] I. Ingemarsson, "Commutative Group Codes for the Gaussian Channel",*IEEE Trans. Inform. Theory*,Vol. IT-19, No. 2, pp. 215-219, Jan. 1973.

[6] G. Ungerboeck, "Channel Coding with multilevel/phase signals," *IEEE Trans. Inform. Theory*, Vol. IT-28, No. 1, pp. 55-67, Jan. 1982.

[7] E. Biglieri, D.Divsalar, P. J. McLane, M. K. Simon, *Introduction to Trellis-Coded Modulation with Applications*, McMillan, 1991

[8] E. Prange, "The Use of Information Sets in Decoding Cyclic Codes", *IRE Trans. Inform. Theory*, Vol. IT-8, pp.S5-S9, 1962.

[9] J.L. Massey, "Coding and modulation in digital communications," *Proc. 1974 International Zürich Seminar on Digital Communications,* Zürich, Switzerland, pp. E2(1)-E2(4), Mar. 1974

[10] K. Nakamura, "A Class of Error Correcting Codes for DPSK Channels", ICC '79, pp 45.4.1-45.4.5

[11] S. A. Rhodes, "Differentially coherent FEC block coding," *Comsat Technical Review,* Vol. 17, No. 2, pp. 283-309, Fall 1987

[12] S. Samejima, K. Enomoto, Y. Watanabe, "Differential PSK System with Nonredundant Error Correction," *IEEE Journal on Select Areas in Comm.*, vol. SAC-1, No.1, pp. 74-81, Jan. 1983

[13] D. Divsalar, M. K. Simon, M. Shahshahani, "The Performance of Trellis-Coded MDPSK with Multiple Symbol Detection,", *IEEE Trans. Commun.*, vol. COM-**38**,, pp. 1391-1403, Sept. 1990

[14] H. Leib and S. Pasupathy, "Noncoherent Block Demodulation of MSK with Inherent and Enhanced Encoding", *IEEE Trans. Comm., to appear*

[15] G.C. Clark, J.B. Cain, *Error-correction Coding for Digital Communications*, New York:Plenum Press, 1981.

[16] H. Loeliger, "Signal Sets Matched To Groups", *IEEE Trans. Inform. Theory*, Vol. IT-37, No. 6, pp. 1675-1682, Nov. 1991.

[17] Proakis, John G., *Digital Communications*, McGraw-Hill, 1983.

[18] S. Benedetto, M. Ajmone Marsan, G. Albertengo, and E. Giachin, "Combined coding and modulation: Theory and applications",*IEEE Trans. on Inform. Theory,* vol. 34, pp. 223-236, Mar. 1988

[19] Blake, I.F., "Codes Over Integer Residue Rings", *Information and Control*, No. 29, pp. 295–300, 1975.

[20] C. Satyanarayana, "Lee Metric Codes over Integer Residue Rings",*IEEE Trans. Inform. Theory*,Vol. IT–25, No. 2, pp. 250–254, March 1979.

[21] W. C. Lindsey and M. K. Simon, *Telecommunications System Engineering*, Englewood Cliffs, NJ, Prentice–Hall,1973

[22] P. HO and D. Fung, "Error Performance of Multiple–Symbol Differential Detection of PSK Signals Transmitted Over Correlated Rayleigh Fading Channels",*IEEE Trans. Comm.*, Vol.40, No.10, pp. 1566–1569, Oct. 1992.

[23] H. van Trees, *Detection, Estimation, and Modulation Theory: Part I.*, New York: Wiley, 1968

[24] P. Ho and D. Fung, "Error performance of multiple symbol differential detection of PSK signals transmitted over correlated Rayleigh fading channels," in *Proc. IEEE ICC'91*, Denver, CO. June 1991, pp. 19.6.1–19.6.7.