Iterative decoding for rate adaptive forward error correction on the mobile satellite channel

by

Michel Pénicaud

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

Master of Applied Science

Ottawa-Carleton Institute for Electrical Engineering
Faculty of Engineering
Department of Electrical Engineering
University of Ottawa
Ottawa, Ontario, Canada, K1N 6N5

May 15, 1996

ISBN 0-612-16454-3

Canada

UNIVERSITÉ D'OTTAWA
UNIVERSITY OF OTTAWA

Abstract

In this thesis, we examine rate adaptive forward error correction codes for the mobile satellite channel. We are especially interested in codes that can be iteratively decoded.

Multilevel coded modulations are considered: partitioning of constellations, coding principle and iterative decoding are studied. We present design rules for multilevel coded modulations through the asymptotic coding gain, the minimum Hamming distance of the code and with information theory arguments. Computer simulations have been run to confirm the validity of the design rules and to determine the impact of the interleaver size on the performance of a coded modulation scheme. A new construction for multilevel coded modulation with parallel concatenation is suggested. Computer simulations have confirmed the efficiency of this new technique.

Product codes are also considered: code construction and iterative decoding algorithms are studied. We focus on the two-dimensional (24,12) Golay product code and find its performances on AWGN and Rayleigh fading channels through computer simulations.

Three rate coding systems for the mobile satellite channel are presented. A performance criterion for speech transmission with such systems have been determined. Adaptive rate techniques can save up to 5.0 dB and 8.8 dB (depending on the type of environment) in terms of transmitted power.

Résumé

Dans cette thèse, nous étudions les codes correcteurs d'erreurs à taux adaptatif pour le canal mobile-satellite. Nous sommes plus particulièrement intéressés par les codes pouvant être décodés de manière itérative.

Nous considérons pour commencer les modulations codées multiniveau: la partition de l'ensemble des signaux, le principe du codage multiniveau et le décodage itératif sont étudiés. Nous donnons des règles de construction des modulations multiniveau tenant compte du gain de codage asymptotique, de la distance minimale de Hamming et d' arguments de théorie de l'information. Nous avons simulé par ordinateur les codes pour confirmer la validité des règles de construction et déterminer l'impact de la taille de l'entrelaceur sur leurs performances. Une nouvelle construction de modulation codée avec concaténation parallèle est proposée. Des simulations informatiques confirment l'efficacité de cette nouvelle technique.

Nous considérons ensuite les codes produit: nous en étudions la construction et le décodage itératif. Nous nous intéressons plus particulièrement au produit à deux dimensions du code de Golay (24,12). Ses performances sont évaluées sur les canaux gaussien et à évanouissement de Rayleigh.

Nous présentons des systèmes de codage adaptatif à trois taux destinés au canal mobile-satellite. Nous avons déterminé un critère de performance pour un tel système adapté à la transmission de parole. Les techniques de taux adaptatifs présentent un gain de 5.0 dB à 8.8 dB (suivant l'environement) sur la puissance transmise.

Acknowledgments

I would like to thank my two supervisors, Dr. Abbas Yongaçoğlu and Dr. Jean-Yves Chouinard for their constant help and strong support on this work.

I wish to thank Mr. Marc Trichard for his helpful suggestions concerning computer simulations.

Contents

List of Figures

List of Tables

List of Acronyms

ACG	Asymptotic coding gain
ACTS	Advanced communication technologies satellite
APK	Amplitude phase shift keying
ARQ	Automatic repeat request
AWGN	Additive white Gaussian noise
BER	Bit error rate
BCM	Block coded modulation
CRC	Communication Research Centre
CSI	Channel state information
CDMA	Code division multiple access
FEC	Forward error correction
FER	Frame error rate
LEOS	Low earth orbit satellite
MAP	Maximum a posteriori probability
MSAT	Mobile Satellite
MLSE	Maximum likelihood sequence estimation
PSK	Phase shift keying
PCE	Power control error
QAM	Quadrature amplitude modulation
QPSK	Quadrature phase shift keying

RSC	Recursive systematic convolutional code
SOVA	Soft output Viterbi algorithm
TCM	Trellis coded modulation
TDMA	Time division multiple access
dB	Decibel

List of Symbols

c	Direct to multipath energy ratio in Rice fading channel
μ	Mean power level
σ	Standard deviation of the power level
A	Time share of shadowing
P_{Rice}	Probability density function of a Rice distributed random variable
P_{Rayl}	Probability density function of a Rayleigh distributed random variable
P_{LN}	Probability density function of a Log-normaly distributed random variable
S_0	Short time average power
S	Momentary received power
n	Number of levels in the signal set
$\mathcal{S}$	Signal set containing 2^n symbols
δ_i^2	Stage i minimum squared Euclidean distance
M	Memory of convolutional code
r	Code rate
C_i	Level i (N, K_i) component code
K_i	Number of information bits in code C_i
CM	Coded Modulation
$S(.)$	Signal mapping function
$d^2(.,.)$	Squared Euclidean distance
$d^2_{min}[CM]$	Minimum squared Euclidean distance of coded modulation CM
$\eta[CM]$	Spectral efficiency of coded modulation CM

d_i	Minimum Hamming distance of level i component code C_i
$\delta_H[CM]$	Minimum Hamming distance of the coded modulation CM
N	Number of symbols in a block
E_s	Energy per symbol
E_b	Energy per information bit
N_0	Single sided power spectral density of white noise
$F(z)$	Probability density function
$I(.;.)$	Mutual information functional
$I(.;.\|.)$	Conditional mutual information functional
C	Capacity of the channel
Z	Zero mean complex gaussian noise
σ^2	Variance of Z
S_{symb}	Transmitted symbol
$B_1B_2\ldots$	Level 1 bit, level 2 bit, ...
f_d	Doppler frequency of the channel
T	Symbol duration
f_dT	Normalized Doppler frequency

Chapter 1

Introduction

1.1 Background

Most of the industrialized countries now benefit from efficient terrestrial mobile telephone networks. A cellular network represents an important investment in terms of equipment. This investment is profitable only in regions where there are enough users (for instance, in cities, near highways). What about rural zones? In general, there is not enough traffic to justify a cellular network. Nevertheless, in these regions the demand for services exists. Other systems for mobile communication have to be designed. Thanks to satellite mobile links, the problem of network and equipment cost can be solved: a satellite can cover a very large area (about the size of Europe or North America with a single geostationary satellite). The cost per unit surface area is small.

A single satellite may not have enough capacity for all the users of an entire continent, even though the area covered is about this size. The constraint is on the total number of users per satellite. This link turns out to be more efficient for low density users on a very large area that can go up to the size of one continent. This may be appropriate for desert type regions where it is not practical to install a wired network.

Mobile satellite communication systems can be divided into three generations [1].

The first generation is characterized by global beam features and rather large user terminals, although these terminals are transportable in principle. The second generation is characterized by multiple beams mobile satellite systems serving typically lap top terminals (the second generation systems have been in service since 1994). A good example of second generation system is the system developed under the MSAT (Mobile SATellite) program led by CRC (Communication Research Centre) in cooperation with NASA in the United States (see [23]). MSAT services have become operational in the last quarter of 1995. MSAT now provides wide-area voice, data and fax services to land, marine and aeronautical mobile users anywhere in North America. The services offered consist of the following five types of narrow band communications services:

- Telephone calls
- Group voice calls
- Fax calls
- Circuit-switched data calls
- Packet switched data calls.

The third generation systems, which are expected to be operational at the beginning of the twenty first century, will be cellular-like systems using non-geostationary satellites and very large number of small beams. For the third generation, mobile satellite communications will be embedded in a universal concept for mobile communications capable of satisfying the need for mobile communication for medium to high traffic densities (using the cellular network) and low traffic density (using mobile satellite link). Large constellation of non-geostationary low earth orbit satellites (LEOS) is very likely to be used in the future as opposed to the first and second generation geostationary satellites.

As we will see, the problem is to maintain an acceptable output error probability in urban environments where the communication link reliability is severely degraded

by signal fading and shadowing. In this work, we attempt to solve this problem by using rate adaptive forward error correction and iterative decoding techniques.

1.2 Motivation

One of the challenges of mobile satellite communications is the time varying channel impairments due to rain attenuation (in the case of K/Ka band) or caused by a change of channel conditions (rural/urban environments in the case of L-band). Reliable communications are much more difficult to achieve when the direct path signal is shadowed by buildings or trees: fading appears on the channel and there is a power loss. Rate adaptive forward error correction is a powerful technique to cope with the time varying nature of the channel. The idea of this adaptive coding scheme consists in adapting the data rate to the channel state and at the same time adapt the code rate to obtain the same number of channel symbols per unit of time. We choose to use three different data rates $(R, \frac{R}{2}, \frac{R}{4})$: these data rates, for instance, may represent compressed speech data rates at 9600 bit/s, 4800 bit/s and 2400 bit/s.

However, for practical reasons, it is not desirable to use three different decoders. A coding problem is to find codes for the three rates that are compatible, i.e., they can be decoded by the same decoder.

Iterative decoding has a convenient property when using different data rates: as the data rate decreases, the number of decoding iterations can increase, keeping the decoding time constant. In this work, we will focus on codes that can be decoded iteratively and the corresponding iterative decoding algorithms. We consider 2-dimensional (24,12) extended Golay product code and the large family of multilevel codes. The first code has the advantage to have a short block size (144 information bits) and the multilevel codes form a very large family and can be designed easily.

In a rate adaptive transmission system, even before performing any coding, by dividing the data rates by two while keeping the message energy the same, the link transmission margin is increased by 3 dB (6 dB for rate $\frac{R}{4}$). With coding we are

looking for an extra 4-5 dB improvement by taking advantage of the extra decoding time.

1.3 Contributions

This thesis attempts to add new results to the existing literature about channel coding for mobile satellite communications. These are:

1. Simulation of the 2-dimensional (24,12) extended Golay product code on AWGN as well as on a Rayleigh fading channel.

2. Modification of multilevel codes by using parallel concatenation.

3. Performance evaluation of rate adaptive forward error correction systems built with multilevel codes and 2-dimensional (24,12) Golay product code on the mobile satellite channel.

1.4 Outline of the thesis

Chapter 2 of this thesis presents the channel model used for mobile satellite communication link. Important channel constraints and communication systems' limitations are also described.

Chapter 3 provides a review of multilevel coded modulation construction and an iterative decoding algorithm. A procedure to select optimum code rates for the component codes is presented. Three coded modulation schemes with 1, 2 and 4 bits/symbol spectral efficiency are evaluated through Monte-Carlo computer simulations on the AWGN channel.

Chapter 4 presents a new approach to construct multilevel codes using parallel concatenation. The coded modulation schemes are evaluated through computer simulations on the AWGN channel.

In Chapter 5 the construction of product codes and an iterative decoding algorithm are briefly described. The two-dimensional (24,12) extended Golay product code is presented and evaluated through computer simulations for AWGN and Rayleigh fading channels.

Chapter 6 presents an evaluation criterion for rate adaptive forward error correction systems. This criterion is applied to the three rate adaptive systems which are built with codes described in the previous chapters.

Conclusions and suggestions for further researches are provided in Chapter 7.

Chapter 2

Land mobile satellite transmission problem description

Considerable research and developement activities aimed at the introduction of land mobile satellite communication have been undertaken recently by different organizations all over the world. MSAT of NASA (U.S.), MSAT program of DOC (Canada), experimental program of Japan, Mobilesat of AUSSAT (Australia) and PRODAT of ESA (Europe) represent some of the research effort in that direction. For these land mobile satellite systems, two frequency bands are essentially considered:

- L-band (1-2 GHz):

  In this case, the channel behavior is well known and documented, several experiments have been conducted and accurate channel models are available [18], [19]. There are several experimental satellites working at those frequencies such as the geostationary satellite MARECS system developed by INMARSAT.

- K/Ka band (20-30 GHz):

  Experimental activities have been undertaken in the last few years in this frequency range. The Advanced Communication Technologies Satellite (ACTS), for instance operates in this frequency band. However, there are many unresolved problems and issues, e.g., no reliable channel model is available at the

present time. It is known that the transmission is affected by rain attenuation: and that the multipath propagation is almost non existent because of very high gain directive antennas typically used in the mobile terminal. The channel can be described as one AWGN with channel outages due to shadowing.

However, for link performance evaluation, we need a channel model that we can rely on: this is the reason why we cannot use the K/Ka band and will use the L-band channel.

2.1 Land mobile satellite channel model

2.1.1 Two-state model

The communication channel modeled is the land mobile satellite L-band one. In reference [19], the mobile satellite channel is modeled as a two-state channel. This channel can be represented as a two-state Markov chain. The probability of the channel being in a bad state is A (time share of shadowing), corresponding to the fraction of time that the channel experiences shadowing. It is changing with the type of environment, e.g., rural or urban environment.

- The good state corresponds to the case when there is a direct path component between the satellite and the mobile. Here a multipath component may exist due to multiple reflections on buildings, hills, or other objects. The multipath component is Rayleigh distributed. Then the resulting channel is Rician. The parameter $c = \frac{\text{Direct path energy}}{\text{Multipath energy}}$ is given for different situations, depending on the elevation angle and the environment (highway, city). Note that, the architecture of the city can affect the channel, for example depending on the streets' size.

- The bad state corresponds to the case when the direct path to the satellite is obstructed. Then there is only indirect path components. The received signal envelope is Rayleigh distributed, with a varying short time average power.

The distribution of the short time average power is log-normal with mean μ and variance σ^2 depending on the channel conditions (elevation angle, type of environment).

The channel parameters are:

- $c = \frac{\text{Direct path energy}}{\text{Multipath energy}}$ for the Rice channel.
- μ, the mean power level (with the direct path energy as a reference), σ its standard deviation.
- A, the probability to be in the bad state.

The bit error process on the physical link can be described by a digital channel model. The two-state Gilbert model [8] with suitable parameters closely approximate the behavior of the land mobile satellite channel.

We want to find the parameters of an "average" mobile satellite channel. We first divide the problem into two parts: What is the type of environment, highway or city? These two environments give completely different sets of parameters. Taking an average on channel parameters has no meaning, this is why we will use the parameters for Hamburg, where the satellite elevation angle is 21°, an average elevation angle given that for the MARECS satellite (geostationary 26°W), the elevation in Europe can vary from 43° in Cadiz to 13° in Stockholm. For simplicity of the analysis, we only use the transmission parameters of one location.

Recall that it is very likely that non-geostationary low earth orbit satellites will be used in third generation systems. The minimum elevation angle achieved by a constellation of satellites depends on the number of satellites and the orbital altitude. Curves are given in [20]: for a given minimum elevation angle, the minimum number of satellites required is plotted as a function of the altitude. For an orbital altitude of 900 km and a minimum elevation angle of 20°, about 150 satellites are required to cover the entire surface of the earth. The channel parameters set we use with elevation angle 21° in Hamburg is also typical for the mobile low earth orbit satellite

channels. More specifically the "average" channel model is the one with parameters measured in Hamburg with the MARECS satellite :

- $c = 16.6$ dB
- $\mu = -7.1$ dB, $\sigma = 5.5$ dB
- $A = 0.03$

on highway environments. For urban propagation conditions those MARECS satellite parameters become:

- $c = 10.6$ dB
- $\mu = -12.3$ dB, $\sigma = 5.0$ dB
- $A = 0.57$

The Doppler frequency of the channel varies from one environment to an other and depend on the mobile speed. In [19], it is shown that $f_d = 30$ Hz is a minimum value and $f_d = 200$ Hz a maximum value.

The resulting channel model is realistic: reliable communication is possible on this channel but difficult to achieve in practice. It would not have been the same with other parameters: an "easy" channel for which transmission can be done with traditional techniques or an "impossible" channel that makes reliable communication cost prohibitively high.

The main difference between the two environments is the time share of shadowing A that is much higher in cities (higher density of buildings shadowing the direct path). We also note that the urban channels lead to worst propagation conditions. This results in less power in the "bad state" and more multipath power in the "good state".

In practice, this model is too complicated to compute the real performance of a system; the log-normal distribution of the short time average power makes it difficult to compute BER. It is not practical to use a moving short time average power in

computer simulations because it makes simulations prohibitively long (considering the high complexity of the codes considered in this thesis) to reach reliable probabilities of error. The probability density functions (pdf) of the momentary received power are:

$$P_{Rice}(S) = ce^{-c(S+1)} I_0(2c\sqrt{S})$$

$$P_{Exp}(S|S_0) = \frac{1}{S_0} \exp(-S/S_0)$$

given that the pdf of the short time average power S_0 follows a log-normal distribution:

$$P_{LN}(S_0) = \frac{10}{\sqrt{2\pi}\sigma \ln 10} \frac{1}{S_0} \exp\left[-\frac{(10 \log S_0 - \mu)^2}{2\sigma^2}\right]$$

Note that, the power of the direct path component has been normalized to unity.

2.1.2 Three-state model

The problem is the varying short time average power of the multipath component in the bad state channel. We divide the bad state into two channel substates: average substate and bad substate. If the mean of the short time power level is μ (in dB) and its standard deviation σ, we use two Rayleigh fading channel substates with power level $\mu + \sigma$ and $\mu - \sigma$, respectively. The two channel substates are equiprobable with probability $\frac{A}{2}$.

The three-state channel model simplification is equivalent to approximate a Gaussian distribution (μ, σ) by a discrete equiprobable distribution with two values: $\mu + \sigma$ and $\mu - \sigma$. It is easy to understand that the simulation is much simpler with this model: the estimated bit error rate converge much faster toward the actual value. There is a trade-off between simulation complexity and the model accuracy.

This simplified model leads to the three states (or substates) model with the following channel complex envelopes:

- Good state: Rice fading channel: $1 + \sqrt{U}e^{j\theta} \times 10^{-\frac{\mu}{20}}$ with probability $1 - A$.
- Average state: Rayleigh fading channel: $\sqrt{U}e^{j\theta} \times 10^{\frac{\mu+\sigma}{20}}$ with probability $\frac{A}{2}$.

- Bad state: Rayleigh fading channel: $\sqrt{U}e^{j\theta} \times 10^{\frac{\mu-\sigma}{20}}$ with probability $\frac{A}{2}$.

The multipath component monentary normalized power U is exponentially distributed with a unity mean. $P_{Exp}(U) = \exp(-U)$. The parameter θ represents the phase of the multipath component of the signal, uniformly distributed in $[0, 2\pi[$.

The channel parameters for the "average" three-states channel model are:

- Good state: Rice fading channel: $c = 16.6$ dB with a probability of 0.97.
- Average state: Rayleigh fading channel with average power -1.6 dB with probability 0.015.
- Bad state: Rayleigh fading channel with average power -12.6 dB with probability 0.015.

for highway propagation conditions, whereas for a city environment these parameters change to:

- Good state: Rice fading channel: $c = 10.6$ dB with a probability of 0.43.
- Average state: Rayleigh fading channel with an average power of -7.3 dB with a probability of 0.285.
- Bad state: Rayleigh fading channel with an average power of -17.3 dB with a probability of 0.285.

2.2 Additional communication channel constraints

2.2.1 Speech transmission

For speech transmission, additional constraints appear. The source coding data rates that we consider in this research are 9600 bit/s, 4800 bit/s and 2400 bit/s. For transmissions with 4, 2 and 1 bits per symbol, these rates corresponds to a symbol duration $T = 0.417$ ms. Mobile communications propagation delay is at least 240

ms when the satellite is geostationary, but it can be as low as 60 ms for a non-geostationary low earth orbit satellite. The maximum delay tolerable for speech applications is estimated to be approximately 300 ms [1]. Therefore, a digitized speech packet contains 60 ms of speech as a maximum with geostationary satellite, but can contain up to 240 ms of speech for LEOS. The geostationary case will be considered, since it is the most difficult one: the code efficiency increases with the block size. That corresponds to a packet size of 144 information bits for a data rate of 2400 bit/s. Note that, these figures are not exact since additional bits are used, they contain rate adaptation information or routing information. This gives a rough estimate of the minimum packet size. If the block is too large, one may choose to send less than 50 ms of speech within a packet. Due to the high propagation delay, an Automatic Repeat Request (ARQ) protocol cannot be used in the system.

For reliable speech transmission, the bit error rate (BER) should be less than 10^{-3}. We will consider the codes performance for this BER.

2.2.2 Data transmission

In the case of data transmission, decoding delay is not as important as in speech transmission. The packets can then contain more information bits and Hybrid ARQ/FEC protocols can be used if needed: ARQ/FEC error control schemes have been especially designed for time varying channels in [7]. The interleaving depth can become larger, long codes being in general more efficient than short codes. This is the reason why we prefer to focus on speech transmission, but most of the result can be applied to the "easier" case of data transmission without any modification. However, in contrast to speech transmission, data transmission does require a much lower BER than 10^{-3}. In general, a BER $\leq 10^{-9}$ is considered as an acceptable for data transmission applications.

[1] From [28], we know that one-way delays up tu 100 ms are not perceptible by the users, however delay up to 300 ms are noticable but tolerable in satellite communication.

2.3 Simulation set-up

The information bits are generated at a rate that varies from 2400 bit/s to 9600 bit/s. The symbols are transmitted on the channel at a rate that depends on the spectral efficiency of the coding system, here the rates are: 2400 symbols/s, 4800 symbol/s and 17200 symbol/s. These rates correspond to the following symbol durations: $T = 0.417$ ms, $T = 0.208$ ms and $T = 0.052$ ms.

With the Doppler frequencies observed between 30 Hz and 200 Hz, the channel is slow fading: the fading amplitude can be considered constant for a symbol duration. The minimum normalized Doppler frequencies become $f_dT = 0.012$, $f_dT = 0.0062$ and $f_dT = 0.0015$. In our simulations, $f_dT = 0.1$ and $f_dT = 0.01$ are used.

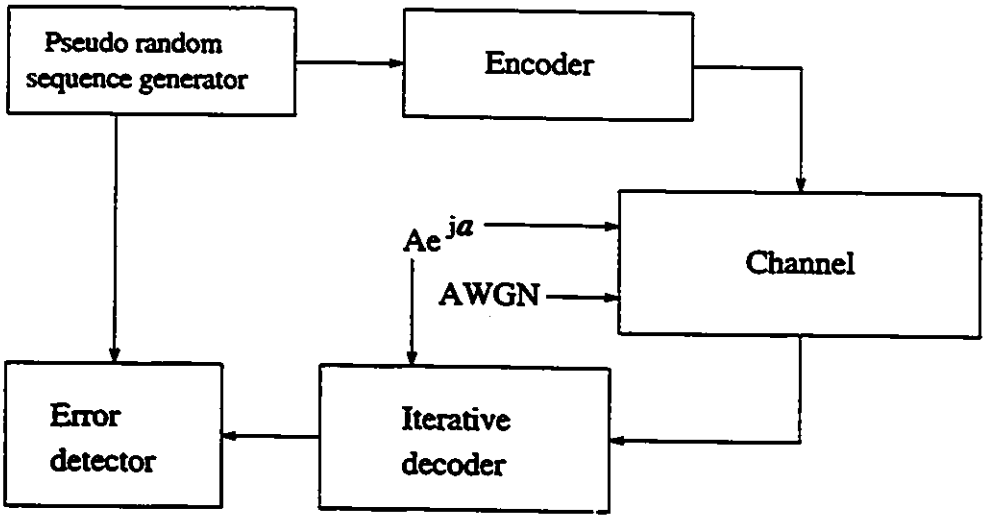

Figure 2.1: *Block diagram of the simulation.*

The simulation programs were written in C language. Figure 2.1 gives the blocks diagram of the simulation set-up. The information bits are generated by a pseudo random sequence, the encoder transforms the information bits into channel symbols that are sent on the channel: the fading amplitude (Ae^{ja}) is multiplied and white noise is added. At the receiving end, we assume perfect channel state information estimation, i.e., we assume that the the fading amplitude of the channel is known by the receiver.

Chapter 3

Multilevel codes

In this chapter, we are looking for coding techniques which can be spectrally efficient. The advantage of the multilevel codes we have chosen is that an iterative decoding algorithm is available (see [31]). To the author's knowledge, such an iterative decoding algorithm does not exist for Ungerboeck's Trellis Coded Modulation[1] (TCM) [29], they are therefore not studied in this thesis. The basic idea of multilevel codes is expressed in papers by Imai [11], Sayegh [27] and Pottie [25]. Note that, standard TCM becomes a special case of multilevel codes. Figure 3.1 present the TCM encoding scheme. We will see in the following that we can construct coded modulation systematically with arbitrarily large minimum squared distance from the Hamming distances of component codes.

3.1 Introduction to multilevel codes

3.1.1 Set partitioning

Let $\mathcal{S}$ be a 2^n symbols signal set. We want to assign n-bits to each symbol in such a way that the minimum distance of each stage is increased (or at least be non decreasing). The minimum distance for stage 1 is $\delta_1^2 = \min_{x,y \in S} d^2(x,y)$.

[1] In the literature, TCM can also denote any coded modulation built with convolutional codes as opposed to BCM (Block Coded Modulation) built with block codes

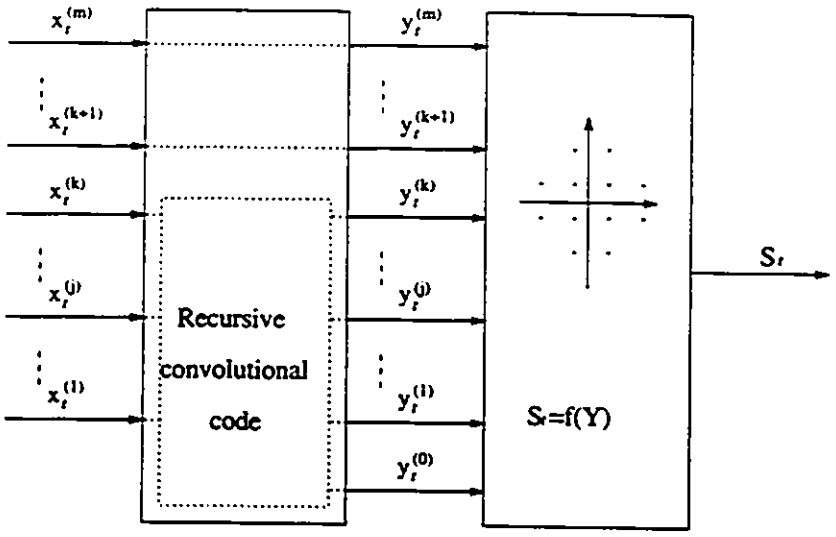

Figure 3.1: *Model of TCM coding scheme.*

Let us define a partition: let $\mathcal{S}$ be a set. $(\mathcal{S}_1 \ldots, \mathcal{S}_m)$ is said to be a partition of $\mathcal{S}$ if:

$$\mathcal{S} = S_1 \cup S_2 \cup \ldots \cup S_m$$

$$\mathcal{S}_i \cap \mathcal{S}_j = \emptyset \text{ for } i \neq j$$

We are particularly interested in partitions that divide the set $\mathcal{S}$ into two sets having the same number of symbols (i.e., 2^{n-1} symbols). We choose the partition $(\mathcal{S}_0, \mathcal{S}_1)$ such that

$$\delta_2^2 = \min(\min_{x,y \in \mathcal{S}_0} d^2(x,y), \min_{x,y \in \mathcal{S}_1} d^2(x,y))$$

is maximum (this way, we maximize the minimum squared distance for stage 2). Then we refine the partition $(\mathcal{S}_0, \mathcal{S}_1)$: choose a partition $(\mathcal{S}_{00}, \mathcal{S}_{01})$ of $\mathcal{S}_0$ and a partition $(\mathcal{S}_{10}, \mathcal{S}_{11})$ of $\mathcal{S}_1$ following the same rules, maximizing the minimum squared distance for stage 3.

We continue the partitioning process until we get 2^n sets, each containing only one symbol. The single element in $\mathcal{S}_{i_1,i_2,\ldots,i_n}$ is identified as $S(i_1 i_2 \ldots i_n)$. $S(.)$ is the function that maps n-bit sequences to symbols in $\mathcal{S}$. We can also identify the symbols with a number m: $S(m) = S(i_1 i_2 \ldots i_n)$ with $m = \sum_{j=1}^{j=n} i_j 2^{j-1}$, considering that the

left bit is the least significant bit (which is also the least protected bit). i_q is the bit from level q, the minimum squared symbol distance for stage q is:

$$\delta_q^2 = \min_{p_i=r_i \text{ for } i<q} d^2(S(p_1p_2\ldots p_n), S(r_1r_2\ldots r_n))$$

The meaning of δ_q^2 is the minimum squared symbol distance given that the bits from the previous levels are known. One of the properties of this partitioning scheme is that: $\delta_1^2 \leq \delta_2^2 \leq \ldots \leq \delta_n^2$. The $\{\delta_i^2\}$ are also called intraset distances.

Set partitioning of the 8PSK constellation

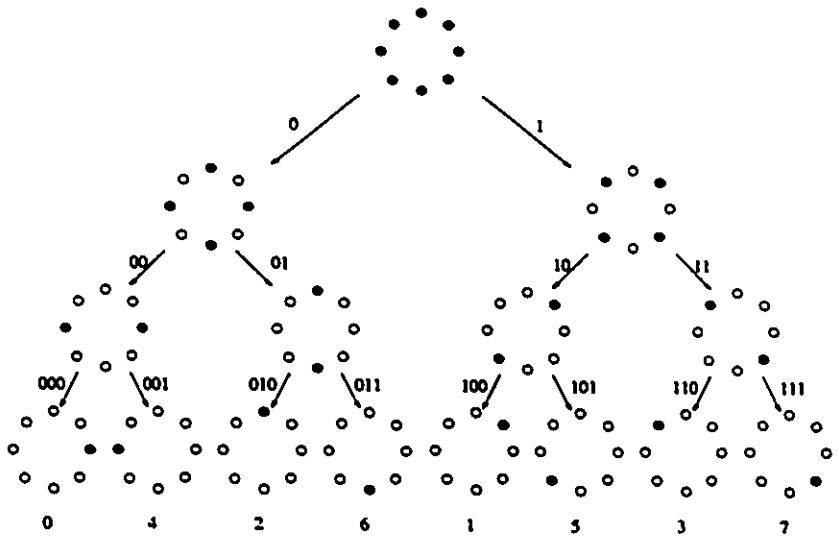

Figure 3.2: *Set partitioning of the 8PSK constellation.*

Figure 3.2 shows how the 8PSK constellation is partitioned in 3 steps and the corresponding signal mapping. The increasing minimum squared distances are: $\delta_1^2 = 2 - \sqrt{2} = 0.586$, $\delta_2^2 = 2$ and $\delta_3^2 = 4$. The average symbol energy is $E_s = 1$. This is only a simple example. The 8PSK constellation is not optimum in the sense that a better ratio $\frac{\delta_1^2}{E_s}$ can be achieved with other 8-symbol sets (with 8APK for example).

Set partitioning of the 8APK constellation

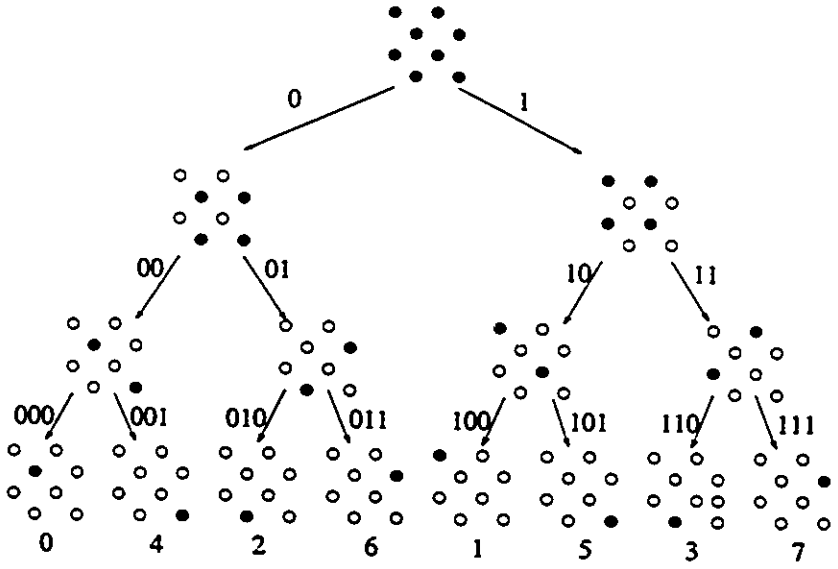

Figure 3.3: *Set partitioning of the 8APK constellation.*

Figure 3.3 shows how the 8APK constellation is partitioned in three steps and the corresponding signal mapping. The minimum squared distances are: $\delta_1^2 = 2$, $\delta_2^2 = 4$ and $\delta_3^2 = 8$ with average symbol energy $E_s = 2.5$. Note that, to compare this partitioning with the one for 8PSK, the average symbol energy should be normalized to $E_s = 1$.

Set partitioning of the 32APK constellation

Figure 3.4 shows the first two steps of the set partitioning for the 32APK constellation. After two successive partitions of the set, we get four 8APK subconstellations. The partitioning for these 4 subconstellations is done like for a standard 8APK constellation.

The minimum squared distances are: $\delta_1^2 = 2$, $\delta_2^2 = 4$, $\delta_3^2 = 8$, $\delta_4^2 = 16$ and $\delta_5^2 = 32$, with an average symbol energy $E_s = 10.5$.

The 16QAM signal constellation is not considered here because we are interested in coded modulations with 1, 2 and 4 bits/symbols spectral efficiencies, which correspond

respectively to 4, 8 and 32 symbols' signal sets after adding one parity bit.

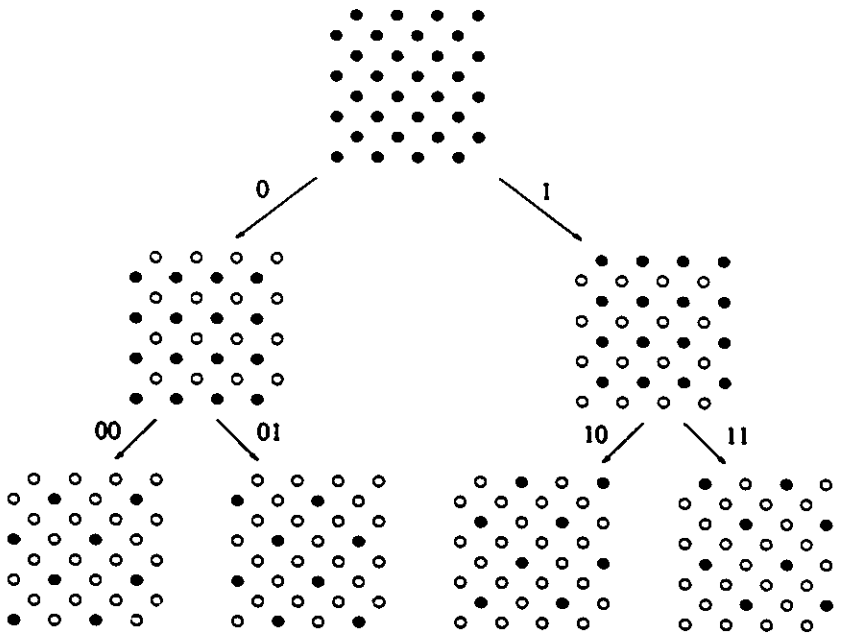

Figure 3.4: *Set partitioning of the 32APK constellation.*

3.1.2 Coding principle

In this section, we explain how the coding is done for multilevel codes.

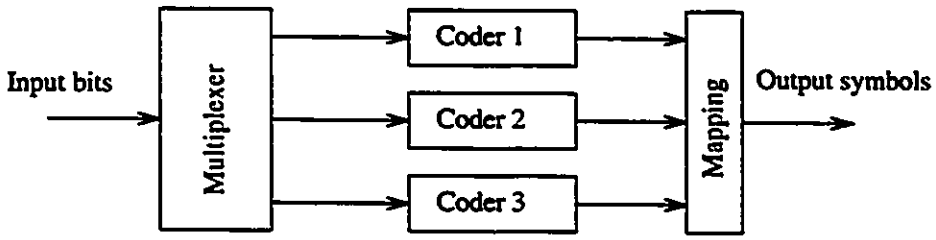

Figure 3.5: *Multilevel encoder with three levels.*

We can encode the bits from each level in an independent way, as shown in Figure 3.5. Suppose we are using block codes: this assumption is general since when using a convolutional code (of rate r and memory M) to encode K bits, the reset bits make the convolutional code a block code of rate $r' = \frac{K}{K+M}r$. If N is the number

of symbols sent, then for each level, we need N encoded bits. For stage q, K_q information bits are encoded in N bits with the code C_q of rate $r_q = \frac{K_q}{N}$ and minimum distance d_q. At that point, we have N n-bit symbols that are mapped in the signal set. $CM = S(C_1 C_2 \dots C_n)$ denotes the set of all possible N-symbol sequences. This is coded modulation that depends on the signal mapping and the n codes. CM is defined by the signal set, the signal mapping $S(.)$ (that is determined by the partitioning) and by the n codes. Note that, before forming the N n-bit symbols, we can permute the bits inside the same partitioning level. This type of interleaving is interesting when using convolutional codes to limit the error propagation from one stage to the next, as it will be seen in details in Section 3.3.1 with an example.

Here, the modulation is divided into n levels, but it can only be divided into any number q of levels with $q \leq n$. Only q codes are used and levels are grouped such that code C_i for $1 \leq i \leq q$ is a $(L_i N, K_i)$ code and bits from level $m_{i-1} + 1$ to level m_i form a codeword in C_i. Here $m_0 = 0$, $m_q = n$ and $m_i = \sum_{j=1}^{i} L_j$. Now, we can clearly see that the TCM scheme described in Figure 3.1 is a special case of a two level multilevel coded modulation scheme, using a recursive convolutional code for the first stage and uncoded bits for the second stage.

3.1.3 Minimum distance and spectral efficiency.

The minimum Euclidean squared distance $d^2_{min}[CM]$ of the coded modulation CM is

$$d^2_{min}[CM] = \min_{x,y \in S(C_1 \dots C_n)} d^2(x,y) = \min_{1 \leq q \leq n} d_q \delta_q^2$$

The demonstration is as follows:
let $S^{(1)} = (S(i_1^{(k)} i_2^{(k)} \dots i_n^{(k)}))_{k=1}^{k=N}$ and $S^{(2)} = (S(j_1^{(k)} j_2^{(k)} \dots j_n^{(k)}))_{k=1}^{k=N}$, be two different symbol sequences in $S(C_1 C_2 \dots C_n)$,

$$d^2(S^{(1)}, S^{(2)}) = \sum_{k=1}^{N} d^2(S(i_1^{(k)} i_2^{(k)} \dots i_n^{(k)}), S(j_1^{(k)} j_2^{(k)} \dots j_n^{(k)}))$$

Since $S^{(1)} \neq S^{(2)}$, we can find the first stage p for which the bits differ,

$$d^2(S(i_1^{(k)} \dots i_{p-1}^{(k)} i_p^{(k)} \dots i_n^{(k)}), S(j_1^{(k)} \dots j_{p-1}^{(k)} j_p^{(k)} \dots j_n^{(k)})) \geq \delta_p^2$$

This is the definition of δ_p^2 since $i_p^{(k)} \neq j_p^{(k)}$ and $i_q^{(k)} = j_q^{(k)}$ for $q < p$. The two codewords on stage p differ at least on d_p bits, and then $d^2(S^{(1)}, S^{(2)}) \geq d_q \delta_q$. Finally,

$$d_{min}^2[CM] \geq \min_{1 \leq q \leq n} d_p \delta_p^2$$

It is now easy to find in each particular case an example for which the equality holds, but it cannot be done for general modulation.

The spectral efficiency $\eta[CM]$ expressed in terms of bits per symbol:

$$\eta[CM] = \frac{1}{N} \sum_{q=1}^{n} K_q = \sum_{q=1}^{n} r_q$$

where r_q is the rate of the code C_q for stage q.

3.2 Multilevel code design

Here, we show how to select the codes in order to achieve a desired performance. The two parameters to compute are the asymptotic coding gain and the minimum Hamming distance. This section will also give an idea of the best performances we can expect from a code.

3.2.1 Asymptotic coding gain on AWGN channel

In Section 3.1.3, we showed how to compute the minimum distance of a coded modulation scheme, given the symbol set, the partitioning and the n codes. The higher the minimum squared distance is the better it is. However, we also have to keep in mind that the spectral efficiency, number of information bits per symbol is fixed (or is limited to a given range).

A first remark is that we should use codes $C_1, C_2 \ldots C_n$ of decreasing power since the minimum distance is increasing. The asymptotic coding gain is defined as follows: for a given coded modulation and uncoded modulation (i.e., reference modulation) of the same spectral efficiency, the asymptotic coding gain is the ratio of the normalized minimum squared distances (normalized means that $E_b = 1$ or the squared distance

divided by E_b). The asymptotic coding gain is the coding gain on AWGN channel as $\frac{E_b}{N_0} \rightarrow +\infty$. We will see in the section on decoding that the objective is to achieve an effective coding gain as close to the asymptotic coding gain as possible for medium bit error rates. This is the best coding gain we can expect for $\frac{E_b}{N_0} < +\infty$.

Let us take as an example 8PSK modulation with a block size $N = 100$ and the following codes parameters:

- $r_1 = \frac{1}{4}$, $d_1 = 10$.
- $r_2 = \frac{13}{16}$, $d_2 = 3$.
- $r_3 = \frac{15}{16}$, $d_3 = 2$.

where d_i denotes the minimum distance of code C_i. The minimum Euclidean distance is $d^2_{min}[CM] = \min(10 \times 0.586, 3 \times 2, 2 \times 4) = 5.86$. The spectral efficiency is $\eta[CM] = \frac{1}{4} + \frac{13}{16} + \frac{15}{16} = 2$ (in bits per symbol). The symbol energy is $E_s = 1$. Then the information bit energy is $E_b = \frac{E_s}{\eta} = \frac{1}{2} = 0.5$. We choose 4PSK as the reference for a 2 bits/symbol spectral efficiency ($E_b = \frac{1}{2}, d^2_{min} = 2$). The asymptotic coding gain is then:

$$ACG = \frac{11.72}{4} = 2.93 = 4.67 \text{ dB}$$

3.2.2 Minimum Hamming distance on fading channels

The Hamming distance between two coded symbol sequences x and y is the number of positions in which the symbols differ, it is denoted $\delta_H(x, y)$. The minimum Hamming distance of the coded modulation CM is denoted $\delta_H[CM] = \min_{x,y \in CM} \delta_H(x, y)$.

We need a way to compute this parameter as easily as the minimum squared distance. For the minimum Hamming distance of the coded modulation CM:

$$\delta_H[CM] = \min_{1 \leq q \leq n} d_q$$

Here is the proof: Let $S^{(1)}$ and $S^{(2)}$ be two different symbol sequences in $S(C_1 C_2 \ldots C_n)$. Since $S^{(1)} \neq S^{(2)}$, the information bits differ at least on one stage q. Since the encoded

bits for stage q differ at least for d_q bits, one obtains

$$\delta_H[CM] \geq \min_{1 \leq q \leq n} d_q$$

Equality holds, if we choose p such that $d_p = \min_{1 \leq q \leq n} d_q$ and choose $S^{(1)}$ and $S^{(2)}$ with d_p different bits for stage p and the exact same bits for the other stages.

$$\delta_H(S^{(1)}, S^{(2)}) = d_p = \min_{1 \leq q \leq n} d_q$$

For a fading channel, the minimum Hamming distance has to be maximized to get a powerful code. This is intuitive if we think about the fading channel as an erasure channel. During deep fades, the intersymbol squared distance is not relevant since the signal energy received is almost zero (and thus we can consider the symbol as erased). Then the higher is the minimum Hamming distance, the more bits needs to be erased before getting a decoding error.

If we go back to the example given in Section 3.2.1, the minimum Hamming distance is $\delta_H[CM] = \min(10, 3, 2) = 2$. The first erased bit causes a high probability of error, and consequently this coded modulation is very weak against fading.

It is clear that two design parameters are not enough to predict the performances of the coded modulation on AWGN channel and Rayleigh fading channel. We also need to consider the path multiplicity of the codes, especially the multiplicity of the limiting stages (i.e. stages that achieve the minimum squared distance).

We will see in Section 3.4, how information theory can help to select the codes $C_1, C_2 \ldots C_n$. In fact, code design strongly depends on the decoding procedure used. The most accurate way to find the performance of a code is to simulate it, again, we need a decoding algorithm.

3.3 Decoding

On an AWGN channel, the problem for choosing a decoding algorithm is to get an effective coding gain as close to the asymptotic coding gain as possible for a reasonable BER (e.g. $10^{-3} - 10^{-6}$). A trade-off has to be made between the decoding

complexity and the BER for which the asymptotic coding gain is reached. When considering decoding, we always look at the optimum procedures: optimum in two different ways, maximum likelihood sequence estimation (MLSE) (very popular for convolutional codes thanks to Viterbi algorithm) or maximum a posteriori probability (MAP) detection. In the case of coded modulations, the complexity of these decoding strategies are prohibitively high (unless the block size is less than 12 information bits, in which case coding becomes practically useless). This is why we are looking at sub-optimal decoding algorithms with reasonable complexity and significant coding gain. Three decoding algorithms are presented: multistage decoding with hard information, multistage decoding with reliability information, and iterative multistage decoding with soft information.

3.3.1 Multistage decoding with hard information and interleaver

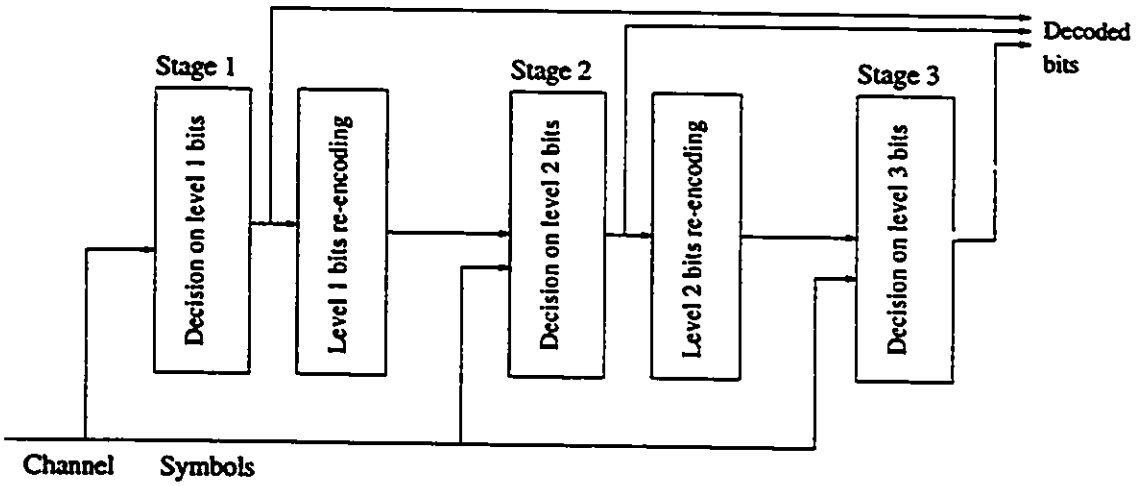

Figure 3.6: *Block diagram of a multistage decoder using hard information with a three level code.*

The multistage decoding with hard information procedure is quite simple. First, the received channel symbols are fed into decoder 1 (for code C_1) which estimates the

transmitted sequence of level 1 information bits. The bits are then reencoded and delivered to decoder 2. Based on information from decoder 1 and channel outputs, decoder 2 estimates the transmitted sequence of level 2 information bits. Stages $3,\ldots,n$ are decoded exactly the same way, using the reencoded channel bits from the previous levels and the channel outputs. Figure 3.6 shows a block diagram of a three stage hard information multistage decoder.

Often, the codes are convolutional codes and the decoder is a Viterbi decoder. The Viterbi algorithm creates error bursts in the estimated sequence of channel bits, and the Viterbi decoder for the next level sees error bursts at the input. This produces an error propagation effect since the Viterbi decoder is very sensitive to error bursts (it is especially designed for correcting independent errors). To solve this error propagation problem, we can use an interleaver that will break the error bursts and create randomlike errors for the next step decoder. Woerz and Hagenauer showed in [31] that the interleaver can provide a 0.5 dB improvement over the non-interleaved code for BER = 10^{-3}. They used the following three stages coded modulation scheme with a SPSK constellation. The three codes are:

- $r_1 = \frac{1}{4}$, $M = 3$ convolutional code.
- $r_2 = \frac{3}{4}$, $M = 3$ convolutional code (punctured from the first code).
- $r_3 = \frac{8}{9}$, $M = 3$ convolutional code (punctured from the first code).

M represents the memory length of the convolutional encoder. The spectral efficiency is $\eta = \frac{1}{4} + \frac{3}{4} + \frac{8}{9} = 1.89$ bits/symbols. The interleaver size is 64×64.

3.3.2 Multistage decoding with reliability information

As opposed to the previous decoding strategy, the information that goes from one stage to the next ones is not hard in nature (i.e., decision based on a bit), but it is a reliability information: typically the log-likelihood ratios or the probabilities of the bits (information or channel bits). We need an algorithm for each stage that provides

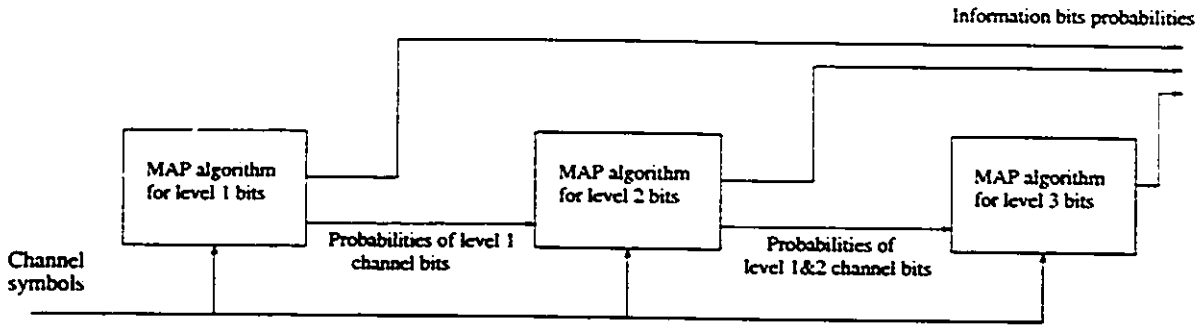

Figure 3.7: *Block diagram of a multistage decoder using reliability information with a three level code.*

soft outputs. We can use the maximum a posteriori probability algorithm (MAP) which is not of high complexity since the decoding is done only stage by stage. Since we need information on channel bits (as opposed to information bits), we use the modified MAP algorithm given by Bahl et al. in [2]. Another solution is to use the SOVA algorithm presented in [10] which is not as complex as the MAP algorithm. Lodge et al. in [2] distinguish between MAP filtering (that outputs the probabilities of channel bits) and MAP decoding (that outputs the probabilities of information bits). The decoding algorithm is the following:

- Stage 1: MAP filtering and decoding is done on level 1 bits with the channel outputs.
- Stage 2: MAP filtering and decoding is done on level 2 bits with the channel outputs as well as the channel bits probabilities found in stage 1.
- ...
- Stage n: And finally MAP decoding is done on level n bits with the channel outputs and the channel bits probabilities found in all previous stages.

Figure 3.7 show a block diagram of a multistage decoder using reliability information for a three level code. Appendix A gives a detailed description of the steps followed

by the MAP algorithm. This decoding algorithm does not propagate as much errors from one stage to the next as the previous one. Thanks to the soft outputs of the modified MAP algorithm and the soft inputs used by the MAP algorithm on the next stage, reliability information is used all the way and error propagations are limited. But in practice, the use of reliability information doesn't provide large coding gains over the interleaved code. Considering the example given in Section 3.3.1, the computer simulations presented in [31] show that the improvement given by the reliability information over the hard information decoder (both using an interleaver) is only 0.1 dB for a BER = 10^{-4}. This is therefore not enough to justify the use of reliability information.

3.3.3 Multistage iterative decoding using reliability information

We have seen that the reliability information doesn't provide significant additional coding gain, but by using soft information in all the stages, we can adapt the previous algorithm to do iterative decoding and use information from level $2,\ldots,n$ channel bits to decode level 1 bits. After level n bits are decoded with the previous algorithm, we can go back to stage 1. If the bit error rates for level q ($q \geq 2$) are low enough (approximately BER $\leq 10^{-2}$) the decoding on level 1 bits can be improved. Symbols are not equally likely anymore: we have an indication of their reliability based on the probabilities of channel bits from level q ($q \geq 2$). The decoding of level 1 bits is then much more powerful than the first one, since soft information propagates through the stages, decoding of all stages is improved. We can go back again to stage 1 and perform a third iteration and so on.

One decoding cycle is:

- Stage 1: MAP filtering is done for level 1 channel bits.
- Stage 2: MAP filtering is done for level 2 channel bits.
- ...

- Stage n: And finally MAP filtering is done for level n channel bits.

For the last cycle MAP filtering and decoding is performed: we want to decode the information bits. The algorithm given in Section 3.3.3 is in fact, a special case of this algorithm with the number of decoding cycles equal to one. The improvement provided by the second decoding cycle is significant: 1.3 dB for the example presented in Section 3.3.1. The following cycles do not improve the performances as much as the second cycle.

Example of multistage iterative decoding

Here we give an example of a code using 8PSK constellation and 1.81 bits per symbol spectral efficiency.

- $r_1 = \frac{1}{4}$, $M = 4$ convolutional code.
- $r_2 = \frac{2}{3}$, $M = 4$ convolutional code (punctured from the first code).
- $r_3 = \frac{8}{9}$, $M = 4$ convolutional code (punctured from the first code).

The spectral efficiency of the coded modulation is $\eta[CM] = \frac{1}{4} + \frac{2}{3} + \frac{8}{9} = 1.81$ bits/symbol.

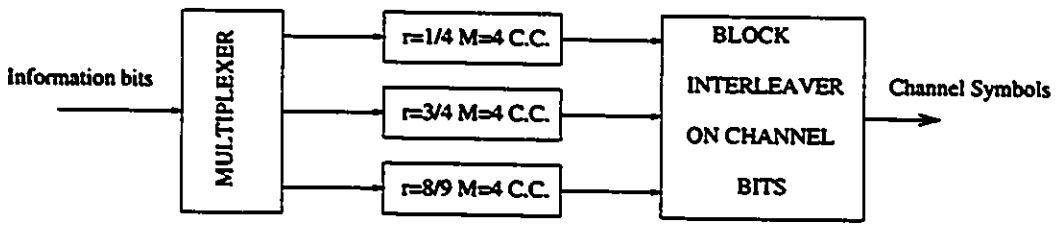

Figure 3.8: *Block diagram of the encoder of the $r_1 = \frac{1}{4}$, $r_2 = \frac{2}{3}$, $r_3 = \frac{8}{9}$ multilevel code.*

The block is 360 channel symbols long. Interleaving is done as follows: channel symbols are arranged in an array (18×20) where the channel bits are written column

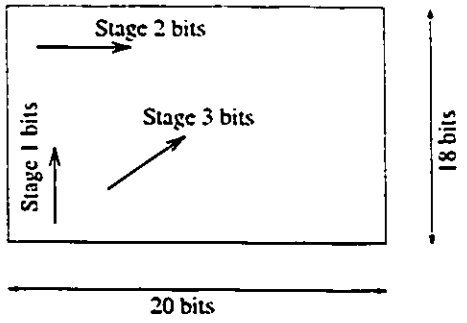

Figure 3.9: *Interleaver for three level coded modulation.*

by column for level 1 bits, row by row for level 2 bits and in diagonal for level 3 bits. The encoder is depicted in Figure 3.8: note that the three outputs of the multiplexer are generated at different rates whereas the output of convolutional encoders are produced at the same rate. The interleaver is depicted in Figure 3.9 for a three level code. To understand precisely how the interleaver is implemented, Figure 3.10 shows the symbols in the interleaver: the 360 symbols are arranged in an array (18 rows and 20 columns).

The coded bits are assigned to symbols as indicated in Figure 3.9: bits are coming out of the encoder in this order

- Level 1: S_{341}, S_{321}, S_{301}, S_{281}, S_{261}, S_{241}, S_{221}, S_{201}, S_{181}, S_{161}, S_{141}, S_{121}, S_{101}, S_{81}, S_{61}, S_{41}, S_{21}, S_1, S_{342}, S_{322}, S_{302}, S_{282}, S_{262},...

- Level 2: S_1, S_2, S_3, S_4, S_5, S_6, S_7,S_8, S_9, S_{10}, S_{11}, S_{12}, S_{13}, S_{14}, S_{15}, S_{16}, S_{17}, S_{18}, S_{19}, S_{20}, S_{21}, S_{22}, S_{23}, ...

- Level 3: S_{341}, S_{322}, S_{303}, S_{284}, S_{265}, S_{246}, S_{227}, S_{208}, S_{189}, S_{170}, S_{151}, S_{132}, S_{113}, S_{94}, S_{75}, S_{56}, S_{37}, S_{18}, S_{359}, S_{340}, S_{342}, S_{323}, S_{304}, S_{285}, S_{266} ...

The way that symbols are read in the array is not important when the channel is AWGN, but it is very critical for fading channels. When computer simulations are done for the AWGN channel, then channel interleaving is not specified (this is the case in this chapter). The bits from the three different levels are interleaved: an error

S_1	S_2	S_3	S_4	S_5	S_6	S_7	...	S_{14}	S_{15}	S_{16}	S_{17}	S_{18}	S_{19}	S_{20}
S_{21}	S_{22}	S_{23}	S_{24}	S_{25}	S_{26}	S_{27}	...	S_{34}	S_{35}	S_{36}	S_{37}	S_{38}	S_{39}	S_{40}
S_{41}	S_{42}	S_{43}	S_{44}	S_{45}	S_{46}	S_{47}	...	S_{54}	S_{55}	S_{56}	S_{57}	S_{58}	S_{59}	S_{60}
S_{61}	S_{62}	S_{63}	S_{64}	S_{65}	S_{66}	S_{67}	...	S_{74}	S_{75}	S_{76}	S_{77}	S_{78}	S_{79}	S_{80}
S_{81}	S_{82}	S_{83}	S_{84}	S_{85}	S_{86}	S_{87}	...	S_{94}	S_{95}	S_{96}	S_{97}	S_{98}	S_{99}	S_{100}
S_{101}	S_{102}	S_{103}	S_{104}	S_{105}	S_{106}	S_{107}	...	S_{114}	S_{115}	S_{116}	S_{117}	S_{118}	S_{119}	S_{120}
S_{121}	S_{122}	S_{123}	S_{124}	S_{125}	S_{126}	S_{127}	...	S_{134}	S_{135}	S_{136}	S_{137}	S_{138}	S_{139}	S_{140}
S_{141}	S_{142}	S_{143}	S_{144}	S_{145}	S_{146}	S_{147}	...	S_{154}	S_{155}	S_{156}	S_{157}	S_{158}	S_{159}	S_{160}
S_{161}	S_{162}	S_{163}	S_{164}	S_{165}	S_{166}	S_{167}	...	S_{174}	S_{175}	S_{176}	S_{177}	S_{178}	S_{179}	S_{180}
S_{181}	S_{182}	S_{183}	S_{184}	S_{185}	S_{186}	S_{187}	...	S_{194}	S_{195}	S_{196}	S_{197}	S_{198}	S_{199}	S_{200}
S_{201}	S_{202}	S_{203}	S_{204}	S_{205}	S_{206}	S_{207}	...	S_{214}	S_{215}	S_{216}	S_{217}	S_{218}	S_{219}	S_{220}
S_{221}	S_{222}	S_{223}	S_{224}	S_{225}	S_{226}	S_{227}	...	S_{234}	S_{235}	S_{236}	S_{237}	S_{238}	S_{239}	S_{240}
S_{241}	S_{242}	S_{243}	S_{244}	S_{245}	S_{246}	S_{247}	...	S_{254}	S_{255}	S_{256}	S_{257}	S_{258}	S_{259}	S_{260}
S_{261}	S_{262}	S_{263}	S_{264}	S_{265}	S_{266}	S_{267}	...	S_{274}	S_{275}	S_{276}	S_{277}	S_{278}	S_{279}	S_{280}
S_{281}	S_{282}	S_{283}	S_{284}	S_{285}	S_{286}	S_{287}	...	S_{294}	S_{295}	S_{296}	S_{297}	S_{298}	S_{299}	S_{300}
S_{301}	S_{302}	S_{303}	S_{304}	S_{305}	S_{306}	S_{307}	...	S_{314}	S_{315}	S_{316}	S_{317}	S_{318}	S_{319}	S_{320}
S_{321}	S_{322}	S_{323}	S_{324}	S_{325}	S_{326}	S_{327}	...	S_{334}	S_{335}	S_{336}	S_{337}	S_{338}	S_{339}	S_{340}
S_{341}	S_{342}	S_{343}	S_{344}	S_{345}	S_{346}	S_{347}	...	S_{354}	S_{355}	S_{356}	S_{357}	S_{358}	S_{359}	S_{360}

Figure 3.10: *Interleaver: symbol transmitted.*

burst from one stage (this is the type of errors encountered with convolutional codes) is distributed into pseudorandom errors.

Results of a computer simulation for the AWGN channel are presented in Figure 3.11 with 1 and 2 decoding cycles. As can be observed from that figure, the second cycle improvement is 1 dB for BER = 10^{-4}. The coding gain over 4PSK at BER = 10^{-4} is equal to $(8.4 - 4.7) = 3.7$ dB. The best Ungerboeck code for 8PSK with a 64 states decoder, which is about of the same decoding complexity, achieves a coding gain of $8.4 - 5.4 = 3$ dB (see [30]). Then with a small bandwidth expansion (spectral efficiency 1.81 instead of 2), the effective coding gain can be increased by 0.7 dB.

3.4 Rate selection via information theory

As said previously, selecting the best code rate for each level is not easy. Information theory helps us to find the maximum code rate for a particular channel: this is the channel capacity. Here we will find the maximum code rate we can use for each level. This is a complicated subject as we will see through an example. Rate selection for multilevel codes via information theory was first presented by Kofman et al. in [12] where the example of 8PSK modulation is treated. In this Section, similar results on 8APK and 32APK are presented. We skipped the 16QAM again, because we are interested in coded modulation with 1, 2 and 4 bits/symbol spectral efficiency that are built with 4,8 and 32 symbols signal sets.

3.4.1 Rate selection for 8 APK

Consider the example of the partition of 8APK of Section 3.1.1. The channel has a discrete input (8 level) and a continuous output (due to additive gaussian noise). B_1, B_2, B_3 are the three bits associated with a symbol: this is the input of the channel. The output is $Y = S(B_1B_2B_3) + Z$ (Z is complex gaussian noise of variance $E[|Z|^2] = 2\sigma^2$). $S_{symb} = S(B_1B_2B_3)$ is the transmitted symbol. The average mutual information

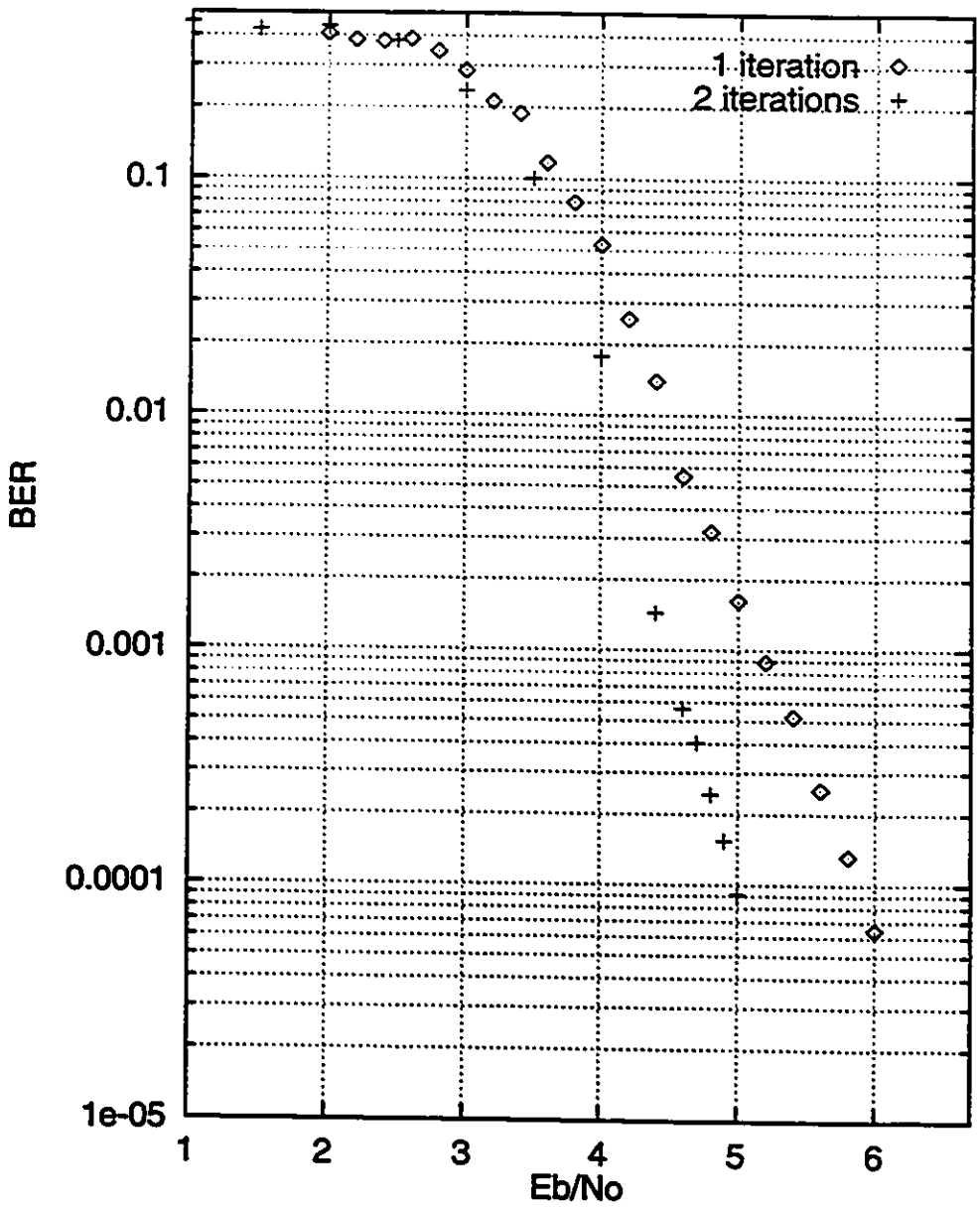

Figure 3.11: *Multilevel code* $r_1 = \frac{1}{4}$, $r_2 = \frac{2}{3}$ *and* $r_3 = \frac{8}{9}$ *for 1 and 2 decoding cycles*

[5] that goes through the channel can be written as:

$$I(Y;S_{symb}) = I(B_1,B_2,B_3;Y) = I(B_1;Y) + I(B_2;Y|B_1) + I(B_3;Y|B_1,B_2)$$

since we are doing multilevel decoding, this decomposition has a meaning: $I(B_1;Y)$ is the amount of information on B_1 provided by Y. $I(B_2;Y|B_1)$ is the amount of information on B_2 provided by Y assuming B_1 is known (this is exactly what we are doing in multistage decoding, if we neglect error propagation). We can write the conditional probability density functions as follows:

$$F(z|b_1=1) = \frac{1}{2\pi\sigma^2} \sum_{i=0,\ i\ \text{odd}}^{\tau} P(S_{symb}=S(i)|b_1=1) \exp\left[-\frac{|z-S(i)|^2}{2\sigma^2}\right]$$

$$F(z|b_1=0) = \frac{1}{2\pi\sigma^2} \sum_{i=0,\ i\ \text{even}}^{\tau} P(S_{symb}=S(i)|b_1=0) \exp\left[-\frac{|z-S(i)|^2}{2\sigma^2}\right]$$

$F(z)$ is the marginal probability density function:

$$F(z) = \frac{1}{2\pi\sigma^2} \sum_{i=0}^{\tau} p_i \exp\left[-\frac{|z-S(i)|^2}{2\sigma^2}\right]$$

Where $p_i = P(S_{symb} = S(i))$. C_1 is the contribution of the first symbol bit to the channel capacity in bit per symbol (if the base of the logarithm is 2).

$$C_1 = \max_{p_i} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \sum_{l=0,1} P(b_1=l) F(z|b_1=l) \log\left[\frac{F(z|b_1=l)}{F(z)}\right] dz$$

For other reason (such as signal spectral properties), we assume equiprobable input symbols, in this case $p_i = \frac{1}{8}$ and $P(b_1=l) = \frac{1}{2}$, $P(S_{symb}=S(i)|b_1=l) = \frac{1}{4}$. Then we don't compute the capacity C_1 as such but the average mutual information for a given set of probabilities $\{p_i\}$. There is no more maximization to be done since the parameters $p_i = \frac{1}{8}$ are fixed. We can now compute the average mutual information $I(B_1;Y)$. Since the analytical results for this type of integral are unknown, numerical integral results must be computed.

$$I(B_1;Y) = \sum_{l=0,1} \frac{1}{2} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} F(z|b_1=l) \log\left[\frac{F(z|b_1=l)}{F(z)}\right] dz$$

Finally for symmetry reasons that are verified in SAPK, $I(B_1;Y)$ can be simplified to

$$I(B_1;Y) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} F(z|b_1=0)\log\left[\frac{F(z|b_1=0)}{F(z)}\right]dz$$

For $I(B_2;Y|B_1)$ and $I(B_3;Y|B_1,B_2)$, the same type of integral must be numerically evaluated. The same simplifying procedures lead to:

$$I(B_2;Y|B_1) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} F(z|b_1=0,b_2=0)\log\left[\frac{F(z|b_1=0,b_2=0)}{F(z|b_1=0)}\right]dz$$

$$I(B_3;Y|B_1,B_2) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} F(z|b_1=0,b_2=0,b_3=0)\log\left[\frac{F(z|b_1=0,b_2=0,b_3=0)}{F(z|b_1=0,b_2=0)}\right]dz$$

The results of the numerical integration for $\frac{E_s}{N_0}$ from 0 to 15 dB are plotted in Figure 3.12. We know that

$$I(Y;S_{symb}) = I(B_1,B_2,B_3;Y) = I(B_1;Y) + I(B_2;Y|B_1) + I(B_3;Y|B_1,B_2)$$

The curves can be used as follows: for a given spectral efficiency η, we find the minimum $\frac{E_s}{N_0}$ required for an error free transmission with $I(Y;S_{symb})$ and find the corresponding rate for level 1,2 and 3 with the rules suggested in [12]. The code rates are $r_1 = I(B_1;Y)$, $r_2 = I(B_2;Y|B_1)$, $r_3 = I(B_3;Y|B_1,B_2)$ for this particular $\frac{E_s}{N_0}$. Note that, as expected: $I(B_1;Y) < I(B_2;Y|B_1) < I(B_3;Y|B_1,B_2)$, as we go further in the partition chain, the contribution to channel capacity increases and the component codes need to be less powerful. As an example, we want to compute the code rates for a $\eta = 2$ coded modulation scheme: $\frac{E_s}{N_0} = 5.33$ dB is the minimum signal-to-noise ratio required for an error free transmission and then $r_1 = 0.364$, $r_2 = 0.676$ and $r_3 = 0.961$. These values do not depend on any target BER. Since the code rates are always a fraction of integer, we try to approximate these numbers by $r_1 = \frac{4}{11}$, $r_2 = \frac{2}{3}$ and $r_3 = \frac{19}{20}$. This type of computation can be done for any signal set and any partition. For instance, this is done in [12] for 8PSK. In fact, the derivations are the same for any 2^n partitioned signal set, numerical integrations can be done for any of them, it is only a problem of computing power.

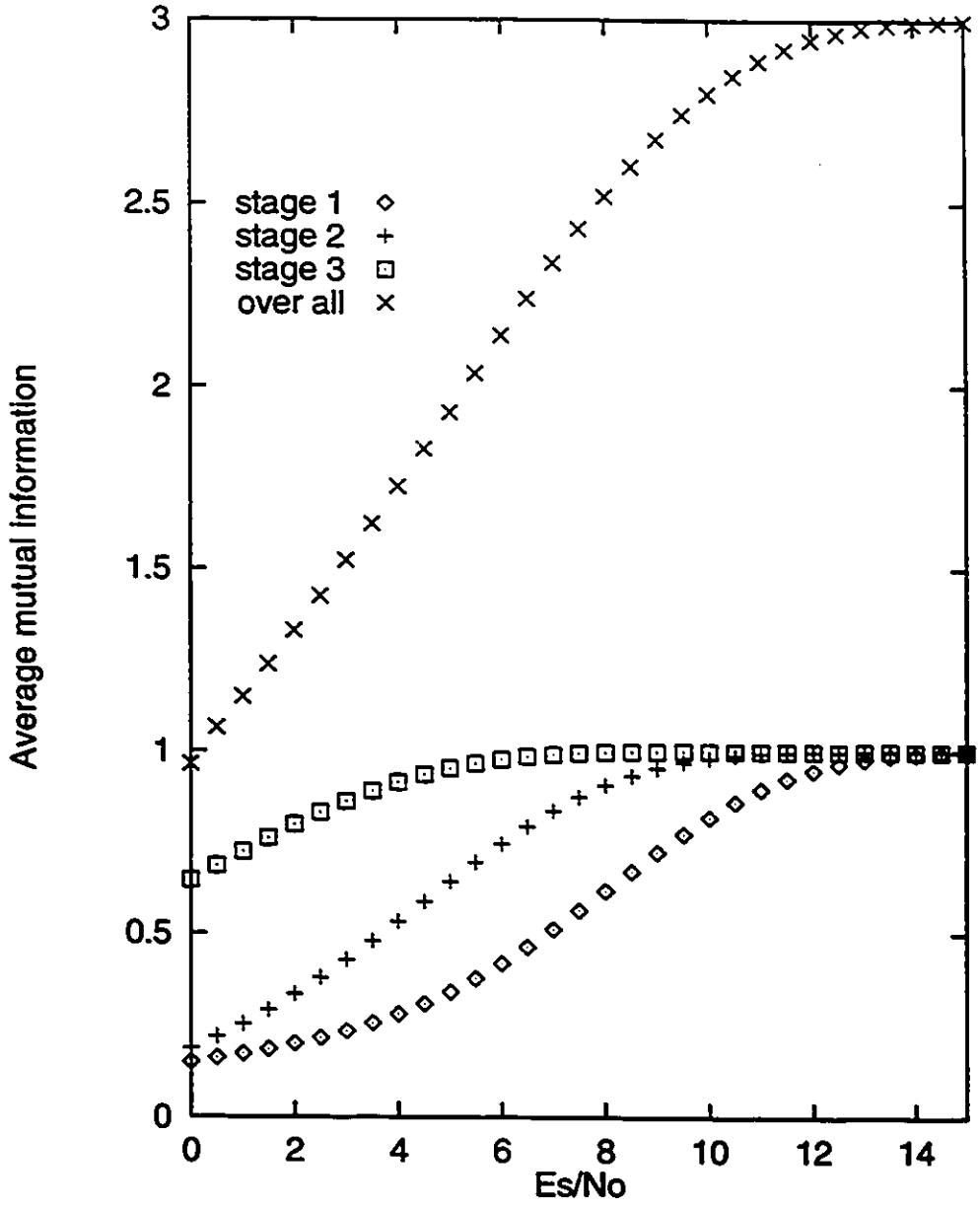

Figure 3.12: *Average mutual information* $I(B_1;Y)$, $I(B_2;Y|B_1)$, $I(B_3;Y|B_1,B_2)$ *and* $I(Y;S_{symb})$ *in bits/symbol as a function of* $\frac{E_s}{N_0}$ *(in dB) for the 8 APK constellation.*

3.4.2 Rate selection for 32 APK

The same computations have been done for the 32 APK signal set (this constellation has been partitioned in section 3.1.1). The computations have simplified as for SAPK.

$$I(B_1;Y) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} F(z|b_1=0)\log\left[\frac{F(z|b_1=0)}{F(z)}\right]dz$$

with

$$F(z|b_1=0) = \frac{1}{2\pi\sigma^2}\sum_{i=0,\ i\ \text{even}}^{31} P(S_{symb}=S(i)|b_1=0)\exp\left[-\frac{|z-S(i)|^2}{2\sigma^2}\right]$$

$$F(z) = \frac{1}{2\pi\sigma^2}\sum_{i=0}^{31} p_i \exp\left[-\frac{|z-S(i)|^2}{2\sigma^2}\right]$$

$I(B_2;Y|B_1)$ $I(B_3;Y|B_1,B_2)$ $I(B_4;Y|B_1,B_2,B_3)$ and $I(B_5;Y|B_1,B_2,B_3,B_4)$ can be simplified and computed exactly the same way, this leads to the graph given in Figure 3.13.

The graph can be interpreted as follows: we want to compute the code rates to design a coded modulation with $\eta = 4$ spectral efficiency. $I(B_1,B_2,B_3,B_4,B_5;Y) = 4$ is achieved for $\frac{E_s}{N_0} = 12.7561$ dB. This is the minimum $\frac{E_s}{N_0}$ required to transmit 4 bits/symbol (corresponding to a $\frac{E_b}{N_0} = 6.736$ dB). Note that, the loss, in terms of $\frac{E_s}{N_0}$ due to the use of 32-ary modulation, is only 1 dB: the capacity of the AWGN channel in bit/symbol is: $C = \log_2(1+\frac{E_s}{N_0})$. The capacity of 4 bits/symbols is achieved for $\frac{E_s}{N_0} = 2^4 - 1 = 15 = 11.761$ dB.

The rates that can be derived on the graph with linear approximation are $r_1 = 0.331$, $r_2 = 0.703$, $r_3 = 0.966$, $r_4 = 0.999$ and $r_5 = 1$. These rates can be approximated by ratio of integer by $r_1 = \frac{1}{3}$, $r_2 = \frac{2}{3}$, $r_3 = \frac{20}{21}$, $r_4 = 1$ $r_5 = 1$. Then no coding is needed for the last two stages: the optimum mapping is not the one derived previously anymore, Gray coding provides the best mapping but it doesn't change the performance significantly. This signal set with the partitioning presented should provide good results.

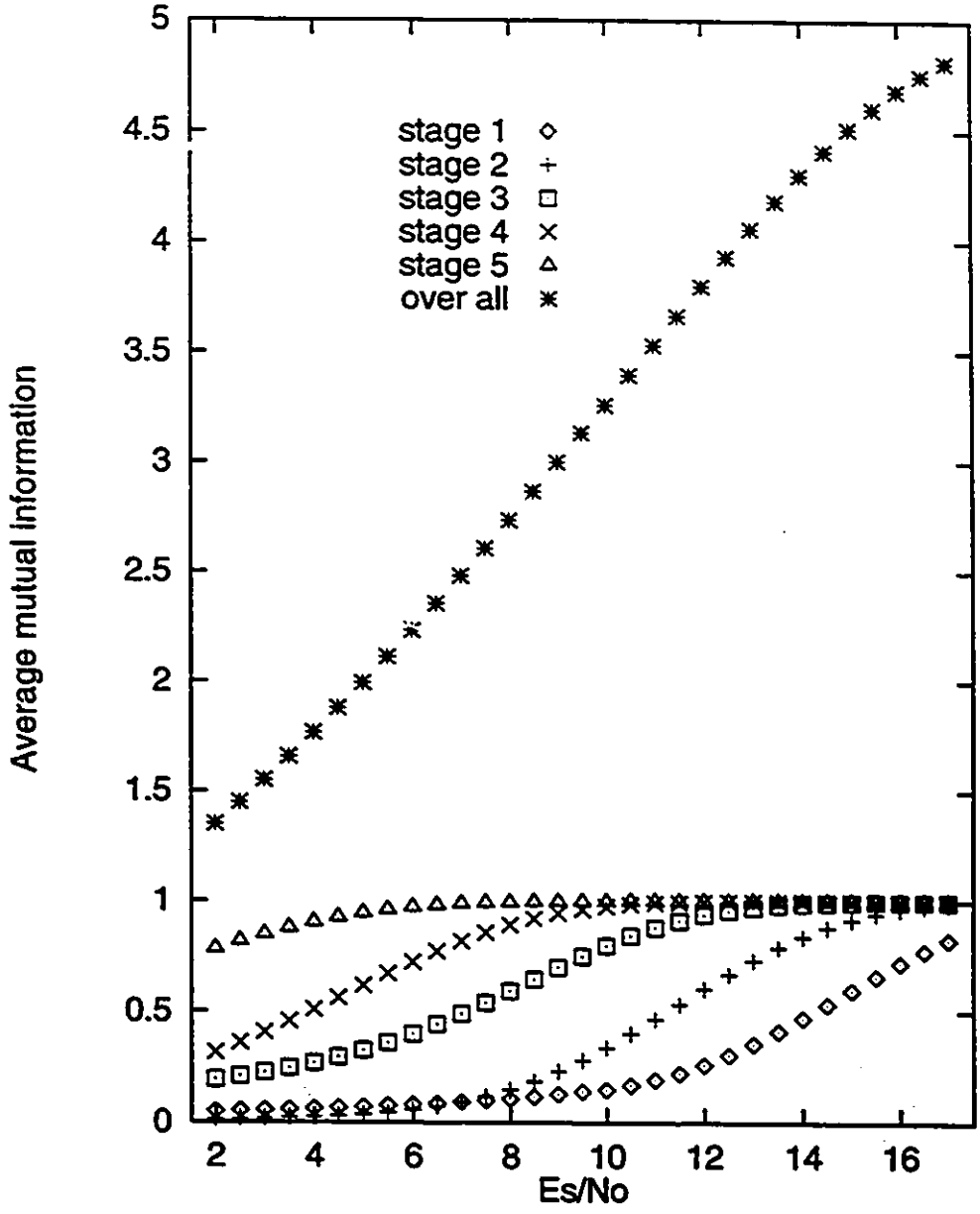

Figure 3.13: *Average mutual information in bits/symbol as a function of* $\frac{E_s}{N_0}$ *(in dB) for 32APK constellation.*

3.5 Simulations on additive white Gaussian noise channel

The additive white Gaussian noise channel (AWGN) is used as a reference to compare BER performances from one coded modulation scheme to another.

In this section, we give simulation results for three coded modulations with spectral efficiencies 1, 2 and 4 bits/symbol. These numbers correspond to a first given data rate and two other rates multiplied by 2 and 4 (they are closely related to the problem).

3.5.1 Coded modulation with one information bit per symbol

The signal set under consideration is the 4PSK constellation. The codes parameters are the following:

- $r_1 = \frac{1}{3}$ $M = 5$ convolutional code. $d_1 = 13$.
- $r_2 = \frac{2}{3}$ $M = 5$ convolutional code, punctured from the first code. $d_2 = 6$.

The minimum squared distance is $d^2_{min}[CM_1] = \min(13 \times 2, 6 \times 4) = 24$, and $E_b = 1$. The uncoded reference modulation is BPSK (Binary Phase Shift Keying). The asymptotic coding gain compared to BPSK is $\mathrm{ACG}[CM_1] = \frac{24}{4} = 6 = 7.78$ dB. The minimum Hamming distance is $\delta_H[CM_1] = \min(13, 6) = 6$, and the spectral efficiency is $\eta[CM_1] = \frac{1}{3} + \frac{2}{3} = 1$ bit/symbol.

Figure 3.14 shows results of computer simulations for coded modulation with 1 and 2 decoding iterations. The effective coding gain over standard BPSK for BER between 10^{-3} and 10^{-4} are 3.7 dB and 4 dB respectively, which is significantly less than the asymptotic coding gain. How does the system work compared to the more classical system: 4PSK with Gray coding and $r = \frac{1}{2}$ convolutional code? In the second case, BER $= 10^{-3}$ is achieved for $\frac{E_b}{N_0} = 3.3$ dB for $M = 5$ and $\frac{E_b}{N_0} = 3$ dB for

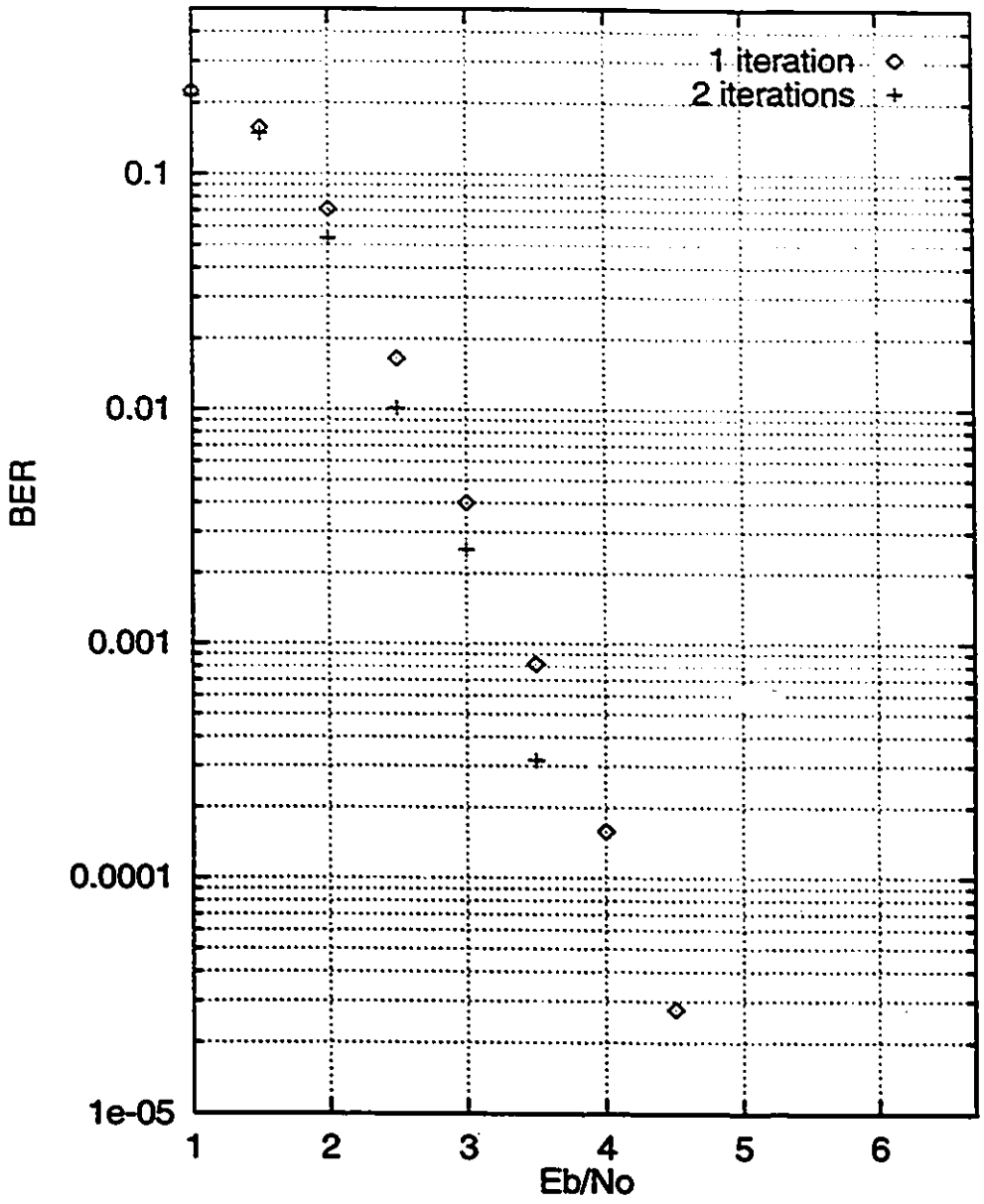

Figure 3.14: *Performance of QPSK constellation with* $r_1 = \frac{1}{3}$, $r_2 = \frac{2}{3}$ *multilevel code and interleaver size 360 in AWGN channel.*

$M = 6$ with soft decision Viterbi decoding. In this case, the advantage of multistage codes is not clear considering the decoding complexity. The gain provided by second iteration is only about 0.3 dB which is not much compared to the 1 dB gain that the second decoding iteration can sometimes provide. The "poor" performance of the multilevel coded modulation is due to high path multiplicity that have strong effect for such high BER. The advantage of a larger minimum distance will be clear for small BERs we cannot easily reach with computer simulations.

3.5.2 Coded modulation with two information bits per symbol

The signal set is the SAPK constellation presented and partitioned in Section 3.1.1. The code parameters are the following:

- $r_1 = \frac{1}{3}$ $M = 5$ convolutional code with $d_1 = 13$.
- $r_2 = \frac{2}{3}$ $M = 5$ convolutional code, punctured from the first code with $d_2 = 6$.
- $r_3 = \frac{9}{10}$ $M = 5$ convolutional code, punctured from the first code with $d_3 = 3$.

The minimum squared distance is $d^2_{min}[CM_2] = \min(13 \times 2, 6 \times 4, 3 \times 8) = 24$. The spectral efficiency is $\eta[CM_2] = \frac{1}{3} + \frac{2}{3} + \frac{9}{10} = 1.9$ bit/symbol. The bit energy is $E_b = \frac{2.5}{1.9} = 1.316$. The asymptotic coding gain compared to 4PSK is $\mathrm{ACG}[CM_2] = \frac{24}{4 \times 1.316} = 4.56 = 6.59$ dB. The minimum Hamming distance is $\delta_H[CM_2] = \min(13, 6, 3) = 3$. The results of computer simulations are plotted in Figure 3.15 for the coded modulations with 1 and 2 decoding iterations. The effective coding gains at BER $= 10^{-4}$ are 3.5 dB and 4.1 dB, which is good for medium BER.

We can test the effect of the interleaver size and use a bigger interleaver even though we know that this will require many information bits per block. Simulation results are plotted in Figure 3.16 for 2 decoding iterations with an interleaver of size 1800. Multiplying the interleaver size by a factor five gave a 0.5 dB gain at 10^{-4}. This bit error rate is achieved for $\frac{E_b}{N_0} = 4.5$ dB, that provided a 4 dB coding gain, only

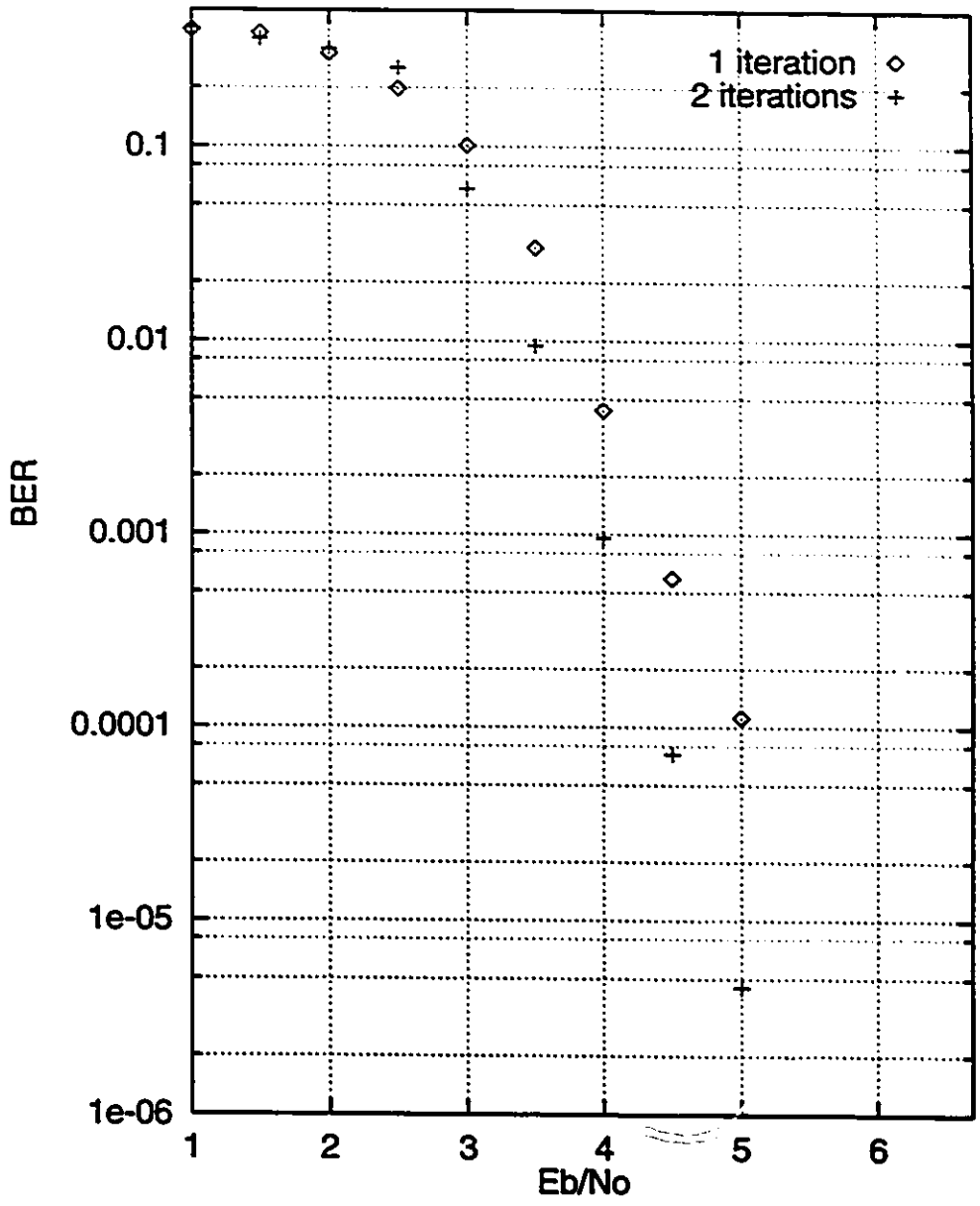

Figure 3.15: *Performance of 8APK constellation with* $r_1 = \frac{1}{3}$, $r_2 = \frac{2}{3}$, $r_3 = \frac{9}{10}$ *multilevel code and interleaver size 360 in AWGN channel.*

2.6 dB under the asymptotic coding gain. This simulation shows the importance of the interleaver as a parameter of the code. The small size of the 360 channel symbol interleaver limits the performance of the multilevel coded modulation for medium to high bit error rates.

3.5.3 Coded modulation with four information bits per symbol

The signal set is the 32APK constellation presented and partitioned in Section 3.1.1.

- $r_1 = \frac{1}{3}$ $M = 5$ convolutional code with $d_1 = 13$.
- $r_2 = \frac{3}{4}$ $M = 5$ convolutional code, punctured from the first code with $d_2 = 4$.
- $r_3 = \frac{9}{10}$ $M = 5$ convolutional code, punctured from the first code with $d_3 = 3$.
- $r_4 = \frac{11}{12}$ $M = 5$ convolutional code, punctured from the first code with $d_4 = 3$.
- $r_5 = 1$ $d_5 = 1$.

The minimum squared distance is $d^2_{min}[CM_3] = \min(13\times2, 4\times4, 3\times8, 3\times16, 1\times32) = 16$. The spectral efficiency is $\eta[CM_3] = \frac{1}{3}+\frac{3}{4}+\frac{9}{10}+\frac{11}{12}+1 = 3.9$ bits/symbol. The bit energy $E_b = \frac{10.5}{3.9} = 2.692$. The uncoded reference modulation is 16QAM that gives a spectral efficiency 4 bits/symbol (with $E_b = 1.25$ and $d^2_{min} = 1$). The asymptotic coding gain compared to 16QAM is $\mathrm{ACG}[CM_3] = \frac{16\times1.25}{2.692} = 7.429 = 8.71$ dB.

A BER $= 10^{-4}$ is achieved for $\frac{E_b}{N_0} = 9.3$ dB and 8.6 dB for one and two decoding cycles, respectively. With 16QAM, a BER=10^{-4} is achieved for $\frac{E_b}{N_0} = 13.6$ dB. The coding gains are then 4.3 dB and 5 dB. Even tough only 50% and 57% of the asymptotic coding gain is achieved at 10^{-4}, these coding gains are significant. The second iteration gives a 0.7 dB gain, which is more than for modulations CM_{11} and CM_{12} with smaller spectral efficiency and same interleaver size. We note that as the spectral efficiency increases, the multilevel coded modulation becomes more powerful.

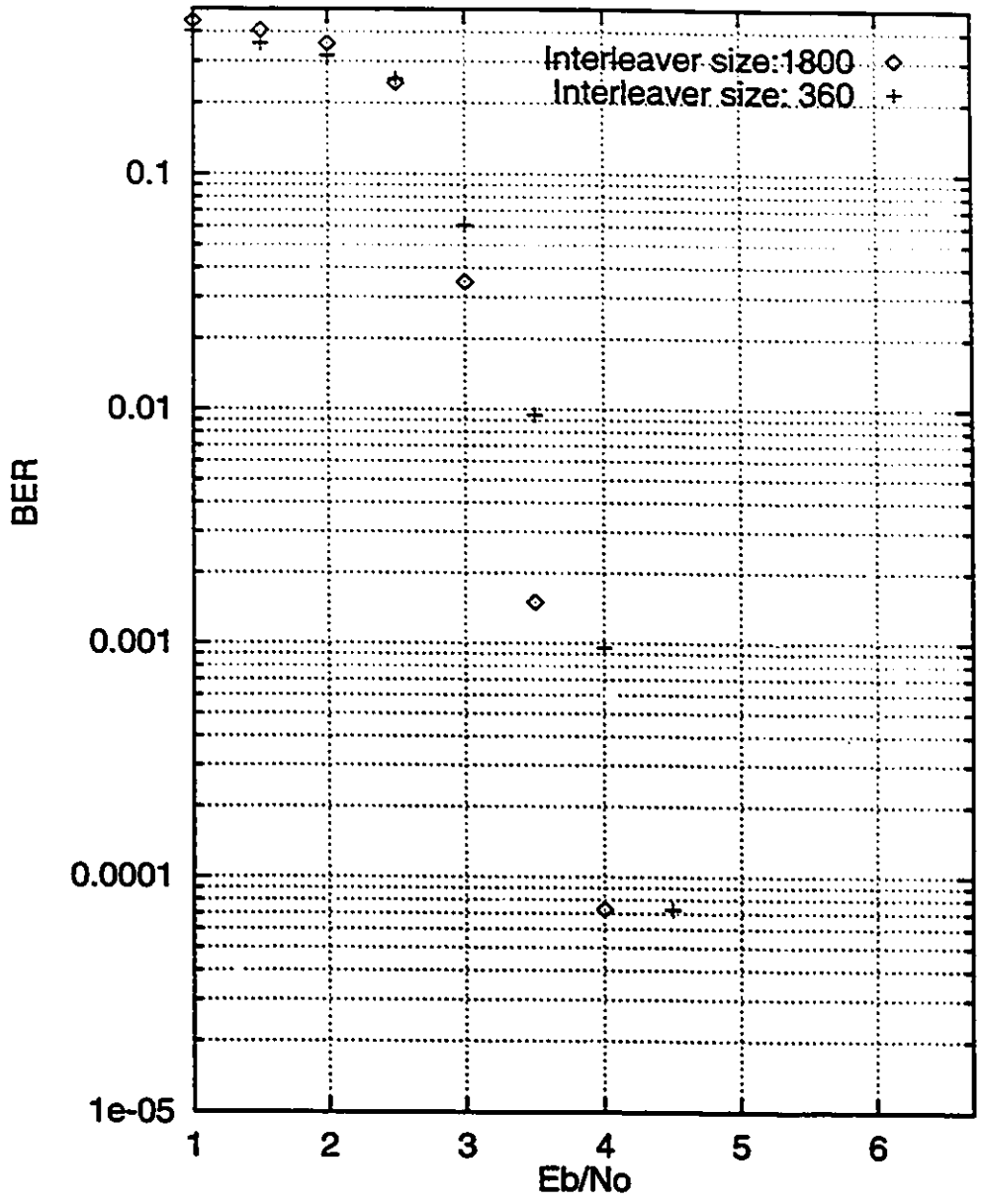

Figure 3.16: *Performance of 8APK constellation with* $r_1 = \frac{1}{3}$, $r_2 = \frac{2}{3}$, $r_3 = \frac{9}{10}$ *multilevel code with two iterations decoding and interleaver sizes 360 and 1800 in AWGN channel.*

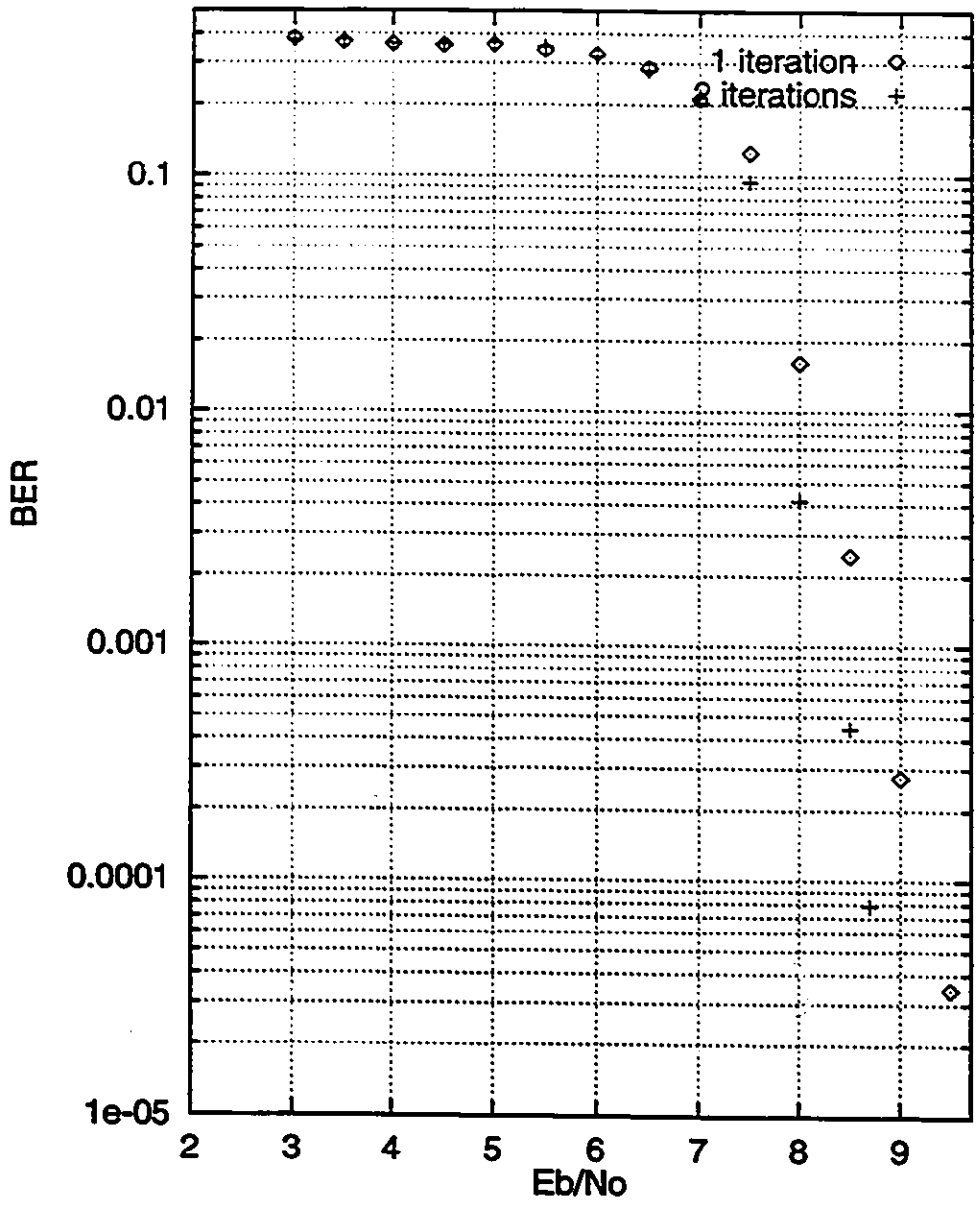

Figure 3.17: *Performance of 32APK constellation with* $r_1 = \frac{1}{3}$, $r_2 = \frac{3}{4}$, $r_3 = \frac{9}{10}$ $r_4 = \frac{11}{12}$ $r_5 = 1$ *code and interleaver size 360 in AWGN*

3.6 Conclusions

In this chapter, we have presented the multilevel codes: partitioning of the signal set, coding principle, interleaving and three decoding algorithms. Design rules have been given, based on minimum squared Euclidean distance, minimum Hamming distance and information theory arguments. Coded modulations with 1, 2 and 4 bits/symbol spectral efficiencies have been evaluated through computer simulations, the coding gains for BER = 10^{-3} are 4.0 dB, 4.1 dB and 5.0 dB, respectively.

Chapter 4

Modified multilevel coded modulation

In this chapter, a new approach to construct multilevel coded modulation is presented. With this coded modulation scheme spectral efficiency is decreased by a small amount (less than 0.25 information bit per symbol). Parallel concatenation is used; this concept was first presented by Berrou et al. in their now famous paper [4] on turbo codes. Since then, turbo codes and parallel concatenation have been widely studied. Turbo codes are simple codes that can achieve impressive error rate results which approach Shannon's theoretical limit.

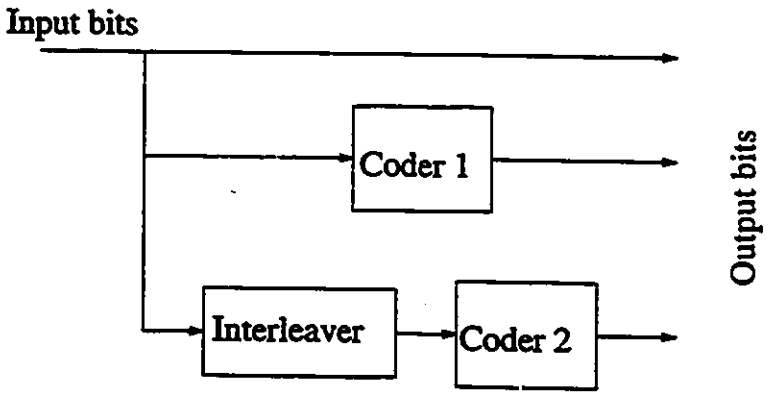

Figure 4.1: *Rate $r = \frac{1}{3}$ turbo encoder, coders 1 and 2 are RSC with rate 1.*

Parallel concatenation is done as follows: the same information bits are encoded

two or three times (using different encoder or the same) after interleaving. With turbo codes, a Recursive Systematic Convolutional (RSC) encoder[1] is used. The use of a RSC encoder has been justified by Battail in [3]: a finite weight input sequence produces an infinite weight output sequence and parallel concatenation makes very unlikely that an infinite weight input sequence will produce a finite weight output sequence.

Figure 4.1 presents a $r = \frac{1}{3}$ turbo encoder where encoders 1 and 2 are RSC encoders with a code rate 1. By puncturing the output of encoders 1 and 2, we can achieve any desired code rate between 1 and $\frac{1}{3}$ as is shown in Figure 4.2. The decoding algorithm is presented in [4]. A modified Bahl algorithm is used to compute the information bit probabilities. Iterative decoding can be done to improve the performance dramatically: for instance, a rate $\frac{1}{2}$ code built with two identical RSC of memory M= 4 followed by standard Gray mapping on QPSK signal set presented in [4] achieves BER= 10^{-5} for $\frac{E_b}{N_0}$=0.7 dB with 18 decoding cycles. This is only 0.7 dB away from the Shannon limit or 8.9 dB coding gain in AWGN.

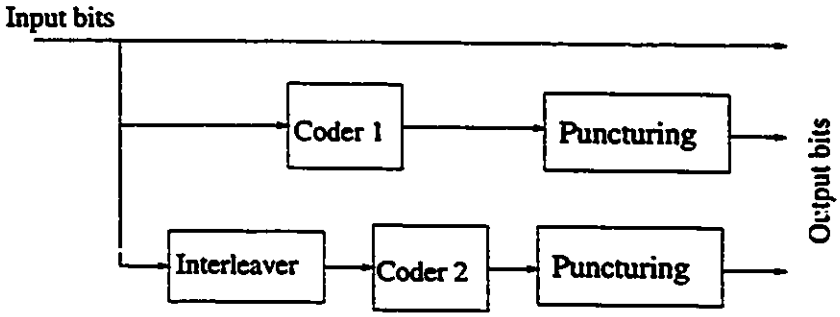

Figure 4.2: *Rate r turbo encoder* ($1 \geq r \geq \frac{1}{3}$).

4.1 Description of the modified multilevel codes

The encoder for a standard multilevel code using a three level coded modulation is depicted in Figure 4.3. Parallel concatenation is applied to modify the block diagram

[1] We use the words coder and encoder as synonimus

of the encoder. Some of the bits which enter encoder 2 also enter encoder 1. The inputs to encoder 1 and 2 are not exactly the same since, as we have seen previously, the code rates are not the same (as seen in the previous chapter the rates satisfy the condition $r_1 < r_2$). Then the three level coded modulation depicted in Figure 4.3 (where the code rates are r_1, r_2 and r_3 with $r_1 \leq r_2 \leq r_3 \leq 1$) can be modified to the one depicted in Figure 4.4. The mapping is done with set partitioning as explained in Section 3.1.1.

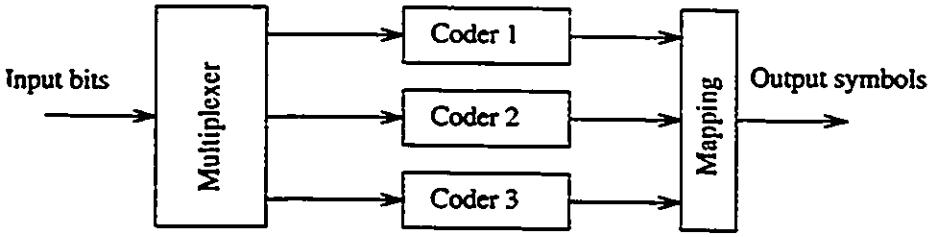

Figure 4.3: *Standard multilevel encoder with 8PSK.*

In Figure 4.4, the bits which enter encoder 2 are punctured and interleaved before entering encoder 1. Again, the 8PSK signal mapping used is the one obtained with set partitioning as presented in Section 3.2. In this case, levels 1 and 2 are in parallel, but modifications can be done for any partitioned signal set with any number of stages in parallel. Note that in our example, only the first two stages were concatenated in parallel, but this could have been done with the three stages in parallel, but this might not be interesting from the spectral efficiency point of view. In this example, the spectral efficiency is $\eta_1 = r_1 + r_2 + r_3$ for the standard procedure, $\eta_2 = r_2 + r_3$ and $\eta_3 = r_3$ for the two types of modifications. This means that, when the three levels are in parallel, the spectral efficiency is less than 1 bit/symbol.

With partitioning and "sufficient" interleaving (i.e. interleaver depth large enough), bit errors from each stage are assumed to be independent. If independence is closely approximated, the idea of parallel concatenation should be very efficient (as in turbo codes). The following example provides insight into the different code design aspects.

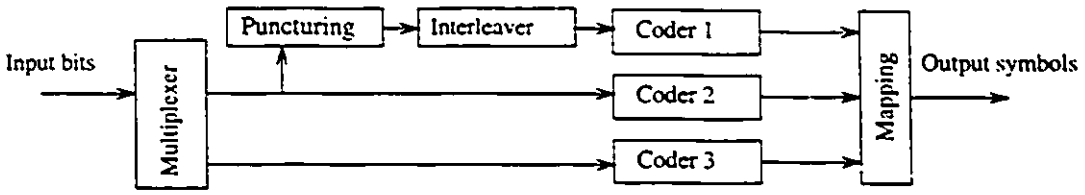

Figure 4.4: *Modified multilevel encoder*

Example:

In this example, modulation used is SPSK partitioned as described previously.

1. C_1: $r_1 = \frac{1}{5}$ M=4 convolutional code. $d_1 = 19$.

2. C_2: $r_2 = \frac{4}{5}$ M=4 convolutional code (punctured from C_1). $d_2 = 3$.

3. C_3: $r_3 = \frac{8}{9}$ M=4 convolutional code (punctured from C_1). $d_3 = 2$.

Recall that M is the memory length and d_i is the free distance of the convolutional code. All these codes are taken from [9] where powerful punctured codes are listed. The spectral efficiency is: $\eta[C] = \frac{4}{5} + \frac{8}{9} = 1.69$ bits/symbol. The energy per bit is then: $E_b = \frac{1}{1.69} = 0.592$.

4.1.1 Evaluation of the asymptotic coding gain

In this section, we show how the asymptotic coding gain evaluation is done using the previous example: the same procedure can be used for any parallel concatenation. The evaluation of the minimum squared euclidean distance of the code is done in two ways. For the first approach, we ignore that a parallel concatenation is done and find the minimum distance of the coded modulation as $d^2_{min}(1) = \min(0.586 \times 19, 2 \times 3, 4 \times 2) = 6$. This is a very pessimistic evaluation, since we ignored that bits from stage 1 are repeated in stage 2. The second approach considers the known bits from stage 1 that are inserted in the stage 2 information bits. Inserting known bits at the input of a convolutional encoder doesn't change anything in the nature of the code: it is still a

convolutional code but not a linear code anymore since "ones" may be inserted. We want to evaluate the minimum distance of this hybrid convolutional code; this is not a straightforward task since the convolutional encoder depends on the information bits that were already encoded in the previous stage (i.e., the known bits). The rate of this hybrid encoder is $r = r_2 - r_1 = \frac{4}{5} - \frac{1}{5} = \frac{3}{5}$.

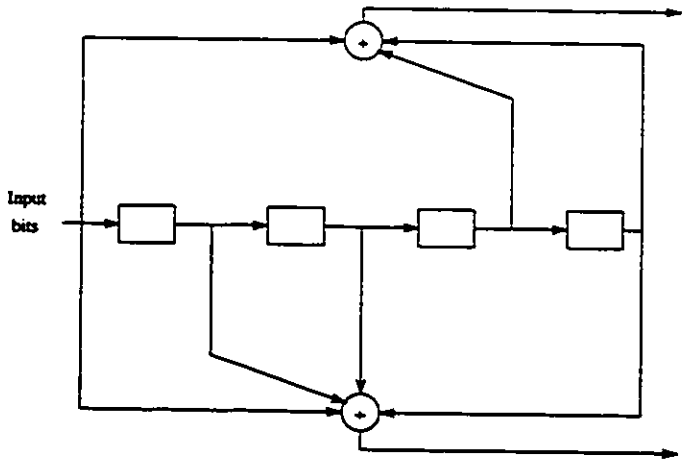

Figure 4.5: *Rate $\frac{1}{2}$ M=4 convolutional encoder*

The known inserted bits increase the minimum distance if they are "regularly" inserted between the bits. In the worst case, the minimum distance doesn't change and is equal to 3. We analyze in detail the rate $\frac{4}{5}$ convolutional encoder to find its free distance. The code is built from the rate $\frac{1}{2}$, $M = 4$ and maximum d_{free}, convolutional code presented in Figure 4.5. The puncturing pattern is 4 bits long and periodically repeated. Figure 4.6 describes the puncturing pattern, where a zero means that the output is deleted, a one means that the output is not deleted. The puncturing pattern is interpreted as follows: at time $t = 1$, the two bits coming out of the encoder are kept; at time $t = 2$, the first bit is deleted, the second is kept; at time $t = 3$, the first bit is kept, the second is deleted; at time $t = 4$, the first bit is deleted, the second one is kept. Therefore, four information bits are coming into the encoder, five encoded bits are going out; resulting in a rate $\frac{4}{5}$ convolutional encoder.

Let us assume $x^{(1)} = x_1^{(1)} x_2^{(1)} \ldots x_K^{(1)}$ and $x^{(2)} = x_1^{(2)} x_2^{(2)} \ldots x_K^{(2)}$ be two input se-

1	0	1	0
1	1	0	1

Figure 4.6: *Puncturing pattern for the rate $\frac{4}{5}$ convolutional code.*

quences with the same inserted bits from level 1. The output of the encoder are $y^{(1)} = y_1^{(1)} y_2^{(1)} \dots y_N^{(1)} = L(x^{(1)})$ and $y^{(2)} = y_1^{(2)} y_2^{(2)} \dots y_N^{(2)} = L(x^{(2)})$ (where $L(.)$ is the linear transformation of the coder). The Hamming distance between $y^{(1)}$ and $y^{(2)}$ is

$$\delta_H(y^{(1)}, y^{(2)}) = W(y^{(1)} \oplus y^{(2)}) = W(L(x^{(1)}) \oplus L(x^{(2)})) = W(L(x^{(1)} \oplus x^{(2)}))$$

where $W(.)$ is the Hamming weight of a sequence and $\oplus$ is the addition in GF[2] (Galois field with two elements). It results that, the free distance of the rate $\frac{3}{5}$ convolutional code is

$$d_{free} = \min\{W(L(x)) \text{ where } x \text{ is a } K \text{ bit sequence with zeros inserted}\}$$

The level 1 bits are inserted one every four bits, but the punctured code is not time invariant (because the puncturing is only periodic). To compute d_{free}, we need to know where level 1 bits will be inserted. There are four possibilities since the puncturing pattern is four bits long. By computer search, we find that in two cases, the free distance equals three. For the two other cases, the free distance is four. By inserting the bits at the right location, we improve the minimum free distance for stage 2. This leads to a minimum squared Euclidean distance $d^2_{min}(2) = \min(0.586 \times 19, 2 \times 4, 4 \times 2) = 8$. The corresponding asymptotic coding gains over 4PSK uncoded modulation are:

$ACG_1 = \frac{6}{0.592 \times 4} = 2.53 = 4.04$ dB and $ACG_2 = \frac{8}{0.592 \times 4} = 3.38 = 5.29$ dB.

As stated before, ACG_1 provides a pessimistic estimation of the coding gain, whereas ACG_2 gives a more realistic coding gain estimate. A 5.29 dB coding gain is very good for a spectral efficiency of 1.69 bit/symbol if it can be reached for a medium BER. We will later simulate the codes to find the actual coding gain values and the influence of the interleaving depth. But before doing any simulation, we need a decoding algorithm.

4.1.2 Decoding algorithm for the modified multilevel codes

The decoding algorithm presented in this section is a modification of the one for standard multilevel coded modulation. The algorithm is given for the previous multilevel code example which uses three stages, the first two being in parallel. The first decoding cycle is:

1. MAP filtering and decoding is done for the first stage bits: the outputs are probabilities on channel and information bits of the first stage.

2. Channel bits probabilities are computed for the second stage.

3. MAP filtering and decoding is done for the second stage bits with a modification: the a priori probabilities of the bits that were already decoded in the first stage are modified. This way, the probabilities on information bits from the first stage are used: the outputs are probabilities on channel and information bits of the second stage.

4. Channel bits probabilities are computed for the third stage.

5. MAP filtering is done for the third stage: the outputs are probabilities on channel bits only.

For the next cycles, a priori probabilities are used for the first decoding stage. For the last stage, MAP decoding is done at the third stage (because probabilities on information bits are needed). The cycles can be iterated as many times as we required, the performance should then improve as the number of cycles increases. This algorithm can be extended to decode any multilevel coded modulation with parallel concatenation. The computer simulations presented in the next section will show how this decoding algorithm performs.

4.2 Computer simulations of modified multilevel coded modulation

In this section, computer simulations are presented for the multilevel code example that was presented in section 4.1. There are two interleaver depths: a small interleaver depth that is matched with the small block size, and a much larger one to see how the BER performance depends on the interleaver size.

4.2.1 Simulation with interleaver size of 360 channel symbols

For the computer simulations done, the following graphs show the bit error rate versus $\frac{E_b}{N_0}$ for each stage of the code and the over all BER. Figure 4.7 shows the simulation results for an interleaver size of 360 bits and for a single decoding cycle. The BER curves for the three stages are not converging, at $\frac{E_b}{N_0} = 5$ dB, they are within a factor range of 5.3: $P_{b1} = 6.8 \times 10^{-4}$, $P_{b2} = 2.7 \times 10^{-3}$ and $P_{b3} = 5.1 \times 10^{-4}$. The best stage is stage 3, the worst is stage 2. The good performance of stage 1 didn't really improved the results for the second stage.

Figure 4.8 shows simulation results for interleaver size 360 channel symbols and two decoding cycles. The curves for the three stages are still not converging; at $\frac{E_b}{N_0} = 5$ dB, they are in a factor range of 8.0, but now stage 1 provides the best results: it has been improved by channel bit probabilities coming from the other stages. However, stage 2 is still the limiting stage. For some reason, the second cycle didn't improve stage 2 performances as much as for the other stages. The problem is that the second decoding cycle increases the factor between the bit error rate of the different stages.

Concerning the "pure" performances, a BER=10^{-3} is achieved at 5.2 dB and 4.7 dB signal-to-noise ratio for one and two decoding cycles, respectively. This is not efficient considering the spectral efficiency $\eta = 1.69$. The improvement provided by

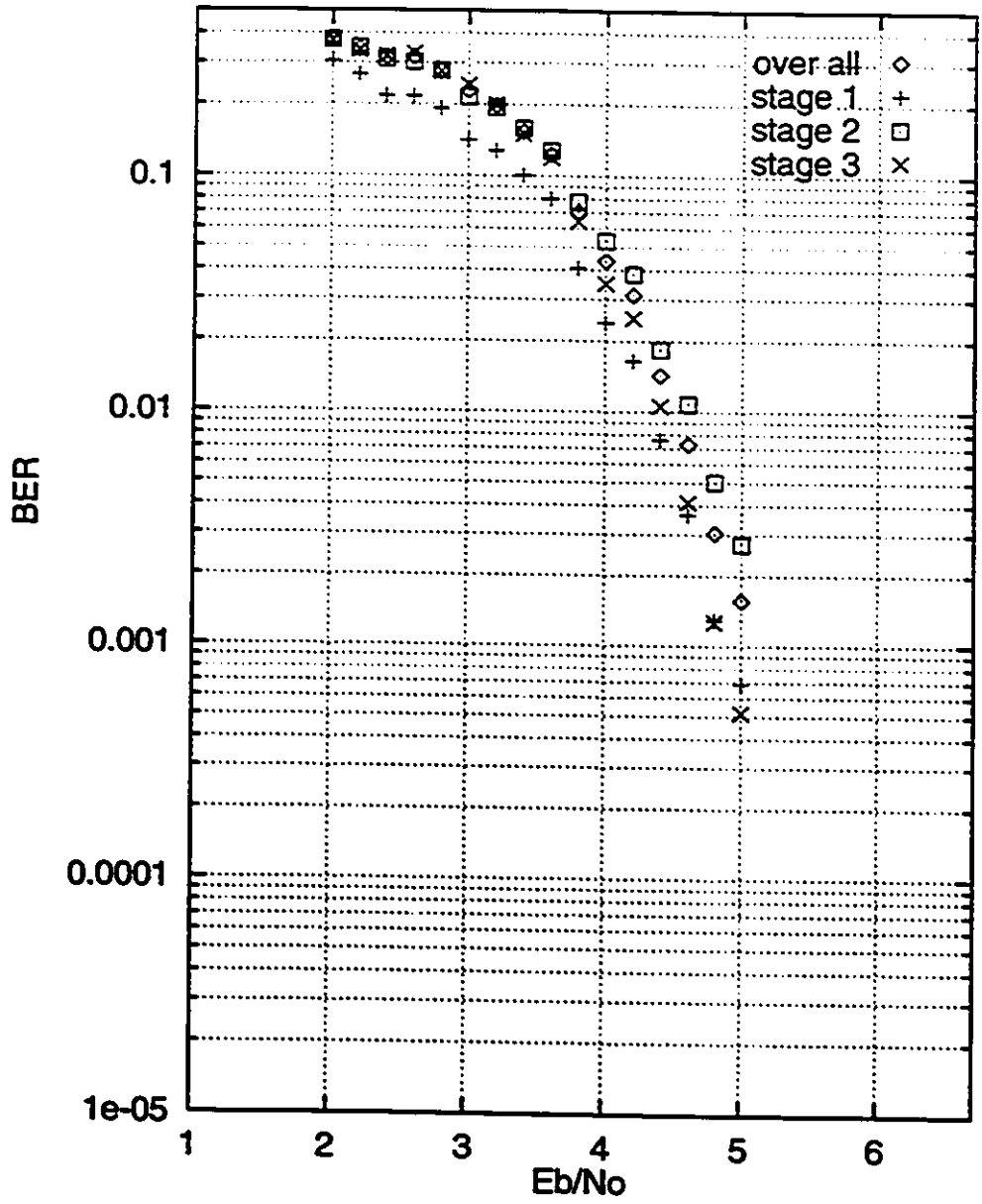

Figure 4.7: *BER performance of the three level modified code with an interleaver size of 360 symbols and a single decoding cycle in an AWGN channel.*

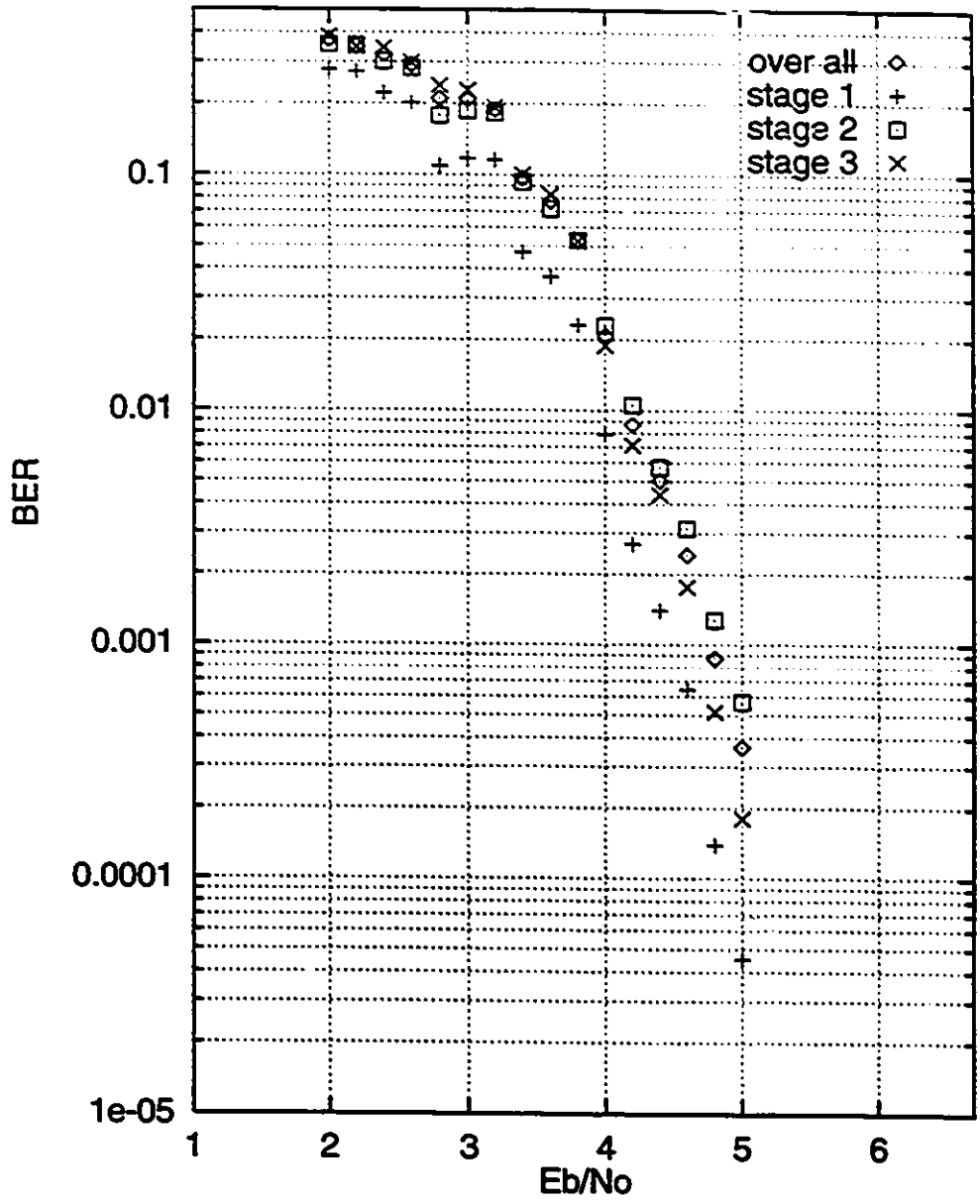

Figure 4.8: *BER performance of the three level modified code with an interleaver size of 360 symbols and 2 decoding cycles in an AWGN channel.*

the second decoding cycle is significant: 0.5 dB.

4.2.2 Simulation with an interleaver size of 3240 channel symbols

For an interleaver size of 3240 channel symbols, results are plotted in Figures 4.9 and 4.10 for one and two decoding cycles. The same bit error rate versus $\frac{E_b}{N_0}$ curves are plotted for each stage of the code.

With only one decoding cycle, the improvement due to extra interleaving depth is not clear: the interleaver additional coding gain at BER $= 10^{-3}$ is only 0.2 dB and the BER curves are within a 20.8 factor range, which is not acceptable: the code rates are not "balanced". Stage 1 is still the best one and stage 2 is the worst.

The performances are dramatically improved by the second decoding cycle: a BER $= 10^{-3}$ is achieved for $\frac{E_b}{N_0} = 3.6$ dB, which is considered a good performance with a spectral efficiency $\eta = 1.69$. The gain provided by the second cycle is 1.4 dB. The BER from the different stages are now very close: they are in a factor range of 1.65 which is very small. Having similar BER values for every stage is a good thing since it shows that the codes are "balanced", no stage really limits the performance of the system. This also shows that a larger interleaver can closely approximate independence between the three stages.

This example shows how efficient the modified multilevel coded modulation could be and under what type of conditions they are efficient: an interleaver of size 3240 channel symbols, or more, for convolutional codes with memory $M = 4$. We have seen that under this condition the second decoding cycle can provide significant extra coding gain. It is clear that with larger memory, we need a larger interleaver.

4.3 Interleaver size

In this Section, we describe a method to choose the size of the interleaver for a modified multilevel coded modulation (the results also apply partly to non-modified

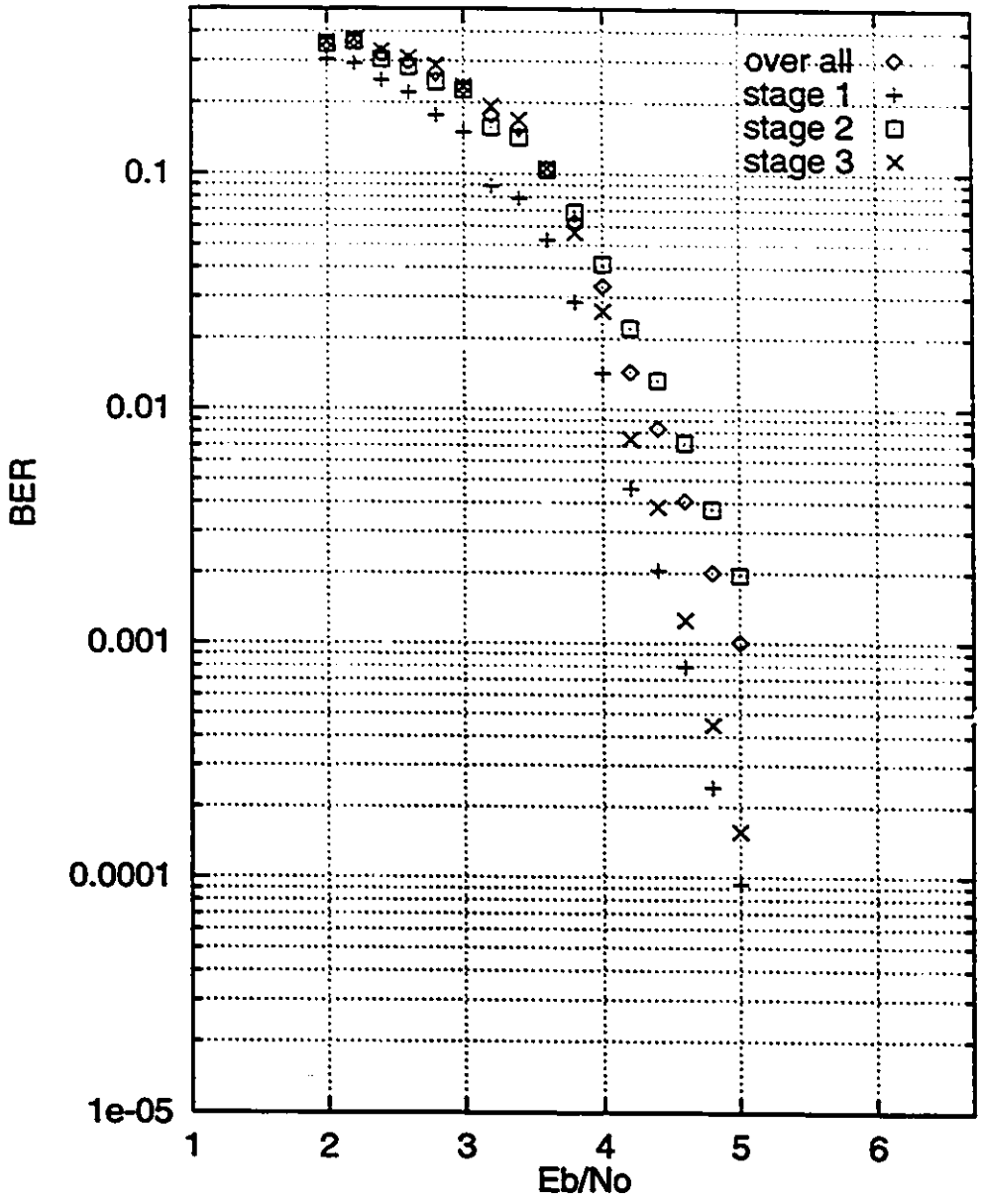

Figure 4.9: *BER performance of the three level modified code with an interleaver size 3240 and 1 decoding cycle in an AWGN channel.*

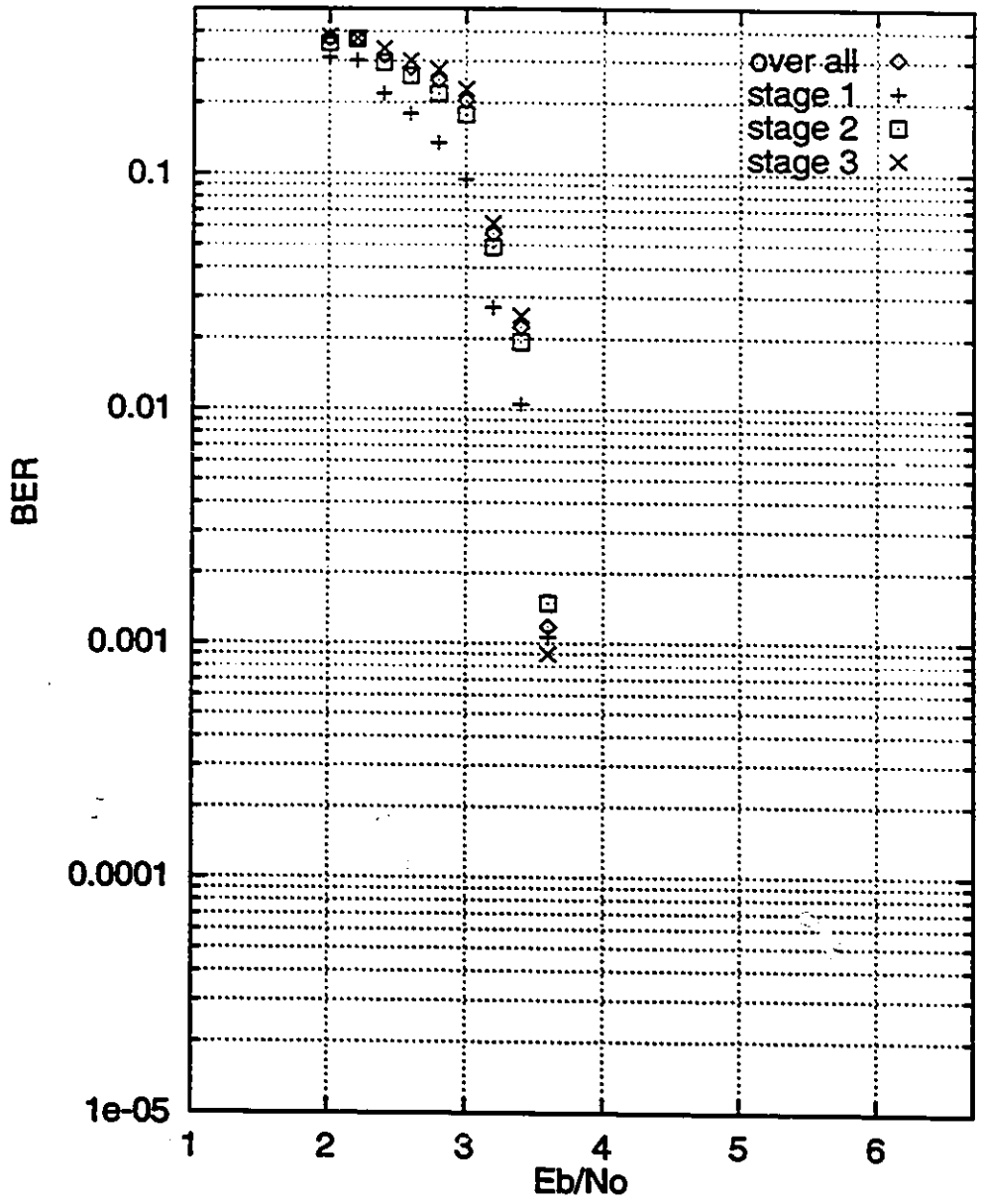

Figure 4.10: *BER performance of the three level modified code with an interleaver size 3240 and 2 decoding cycles in an AWGN channel.*

multilevel coded modulation). We use a three level coded modulation scheme as in the previous example, but the same procedure also apply with more levels. There are two interleavers: one for the information bits before entering encoder 1 (see Figure 4.4) and the other one for channel bits and channels symbols. Their sizes are related since the number of channel symbols vary with the number of information bits in stage 2.

4.3.1 Channel bits interleaver

The channel interleaver has two purposes:

1. Make the levels as uncorrelated as possible. The errors at the output of a convolutional decoder occur in bursts. A burst of errors has to be split into pseudorandom errors for the next stage decoders because convolutional codes are designed to cope with random errors.

2. For channels with memory, we want to eliminate any error bursts which may occur at the input of any decoder for the same reason as above.

The length of an error event produced by a convolutional decoder is less than six times the memory length of the code $6M$ (longer error events are very unlikely, so they can be neglected). We use six times the memory length, because punctured codes present long error events at the output of the decoder (see [9]). This is the length in terms of information bits. In terms of channel bits, the length of the error event depend on the code rate; it is less than $\frac{6M}{r}$, where r is the code rate.

For multilevel codes, where $r_1 \leq r_2 \leq_{\leq} r_3$ (r_i is the rate of C_i, the level i code), the longest error event we will consider is $\frac{6M}{r_1}$ long for the first level, and $\frac{6M}{r_2}$ long for the second level. The interleaver must be such that any $\frac{6M}{r_1}$ bits sequences (on the first level), and $\frac{6M}{r_2}$ (on the second level) must have one symbol (or less) in common. For block interleaver, this corresponds to a block of about $\frac{6M}{r_1}\frac{6M}{r_2}$ bits. If we look back at the example used for computer simulation, a good interleaver size would be about: $6 \times 4 \times 5 \times 6 \times 4 \times \frac{5}{4} = 3600$ channel symbols. We used 360 as the interleaving degree which is clearly not enough and 3240, which is of sufficient size.

Now, we consider the size of the interleaver considering a channel with memory. We need to split bursts of unreliable bits. If f_dT is the normalized Doppler frequency, the memory length of the channel is about $\frac{1}{f_dT}$ symbols. This means that any sequence of $\frac{1}{f_dT}$ symbols and any sequence of $\frac{6M}{r_1}$ bits of level 1 bits must have at most one bit in common. For a block interleaver, this corresponds to a block of $\frac{1}{f_dT}\frac{6M}{r_1}$ or more. For the coded modulation given in the example with a Rayleigh fading channel with $f_dT = 0.1$, the required interleaver size is $10 \times 6 \times 4 \times 5 = 1200$. The discussion on the channel bits interleavers applies to non-modified multilevel coded modulation.

4.3.2 Information bit interleaver

Note that, this interleaver is specific to modified multilevel coded modulation. The size of the information bits interleaver is expressed in terms of level 2 information bits since level 1 bits are chosen in level 2 bits. Level 1 information bit separated by $6M$, or less, must be separated in level 2 by at least $6M$ bits. That makes the block size at least $(6M)^2$ level 2 bits long. In terms of channel bits (to be compared with channel bits interleavers) this translates into $\frac{(6M)^2}{r_2}$ channel bits or more. For the previous example, the interleaver size is 576 level 2 bits or 720 channel symbols. The 360 channel symbols interleaver is just too small, and the 3240 channel symbols interleaver is easily sufficient.

4.4 Conclusions

In this chapter, we have seen a new method to construct multilevel coded modulation with parallel concatenation. An iterative decoding algorithm and asymptotic coding gain evaluation procedure have been given. Computer simulations have been carried out, the results are disappointing with a small interleaver and encouraging for a large one.

Chapter 5

2-dimensional Golay product code

The multilevel codes (modified or not) all have the same drawback; with the block size we want to use (between 150 and 600 information bits per block depending on the data rate used), there are not enough bits in the interleaver. We can't obtain a sufficient interleaving depth to achieve high coding gains for medium BER (i.e. 10^{-3} to 10^{-4} range). We have seen that the penalty due to small interleaver depths is about 1 to 1.5 dB, which is very significant.

This is the reason why we are now looking at powerful codes or coded modulations that work on a small number of symbols. We would like to avoid the use of an interleaver, if possible. In [17], product codes are presented with an iterative decoding algorithm. We are especially interested in codes that can be iteratively decoded. Product codes with a large minimum distance can be constructed.

The main difference here is that we will use QPSK; this implies that the spectral efficiency will be smaller than for coded modulation. But one can think about techniques with higher spectral efficiency using product codes. Product codes can be component codes in coded modulation[1].

In this chapter, we consider the 2-dimensional (24,12) extended Golay product code with iterative decoding. This powerful code has been suggested by Lodge et al. in [17] for iterative decoding. The (24,12) extended Golay code is a rate $\frac{1}{2}$ code, then

[1]This is called product coded modulation (see [26]).

the 2-dimensional product code is a rate $\frac{1}{4}$ code.

5.1 The (24,12) extended Golay code

This section is a short review of the (24,12) extended Golay code. The generator matrix of the code in a systematic form is:

$$G = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1
\end{bmatrix}$$

We note that the code is auto-orthogonal, i.e. the generator matrix is also the parity check matrix. First, we have to define the orthogonality of N-tuples in GF[2]: two N-tuples $(a_i)_{i=1}^{i=N}$ and $(b_i)_{i=1}^{i=N}$ are orthogonal if

$$\sum_{i=1}^{N} a_i b_i = 0 \text{ in GF[2]}$$

A parity check matrix generates the vector space orthogonal to the code. We check that each row of the matrix G is orthogonal to every row (including itself). Since the code is (24,12), every parity check matrix is a 12×24 matrix. Then G is a parity check matrix for the code.

The minimum distance of the (24,12) extended Golay code is 8, this is a classical result proved in [24]. Note that, the (24,12) extended Golay code is not cyclic, then

there is no advantage to use a polynomial representation for that code. The code is an extension of the (23.12) Golay code which is cyclic (the code is extended by adding extra parity bits such that each codeword contains an even number of "ones").

For the MAP algorithm, we will modify the parity check matrix by doing linear combination of the rows to obtain zeros in upper right corner and lower left corner (we know that the resulting matrix is also a parity check matrix for the code).

The parity check matrix, we will use for decoding is:

$$H = \begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1
\end{bmatrix}$$

This parity check matrix clearly shows the number of states in trellis at each instant:

- For $1 \leq t \leq K$, the number of states is $N_t = 2^t$.
- For $K \leq t \leq N$, the number of states is $N_t = 2^{N-t}$.

This is the easiest form for carrying computer simulations, since it minimize the number of states in the trellis at each instant. The states are listed easily at each instant. The matrix H is in a trellis-oriented form (see [14]); this means that all the codewords (recall that this is a parity check matrix and a generator matrix at the same time) start in a different position and end in a different position. These codewords are called "atomic" codewords.

5.2 Short review of product codes

5.2.1 Construction of 2-dimensional product codes

Let C_1 and C_2 be respectively an (N_1,K_1,d_1) and (N_2,K_2,d_2) binary block code, where N_i represents the block length, K_i is the number of information bits per block and d_i is the minimum distance. Again, when we speak about a block code, it can also be a convolutional code with reset bits at the end and be considered as a block code. The product code $C_1 \times C_2$ is a (N_1N_2,K_1K_2,d_1d_2) binary block code formed as follows: a message of K_1K_2 bits is arranged in an array of K_1 rows and K_2 columns. The K_1 rows of K_2 bits are encoded in K_1 N_2-bits codewords in C_2. Then the N_2 columns of K_1 bits are encoded in N_2 codewords in C_1.

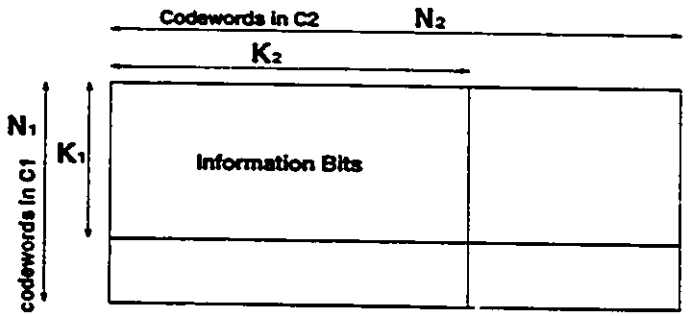

Figure 5.1: *2-dimensional product code with systematic codes.*

The result is clearly a (N_1N_2,K_1K_2) code, and its minimum distance is proved to be d_1d_2 in [15]. The $C_1 \times C_2$ code is linear and systematic if both C_1 and C_2 are linear and systematic. A systematic 2-dimensional product code is depicted in Figure 5.1. The construction of an n-dimensional product code ($n > 2$) is a straight forward extension of the same procedure.

5.2.2 Iterative decoding for product codes

The code can be viewed as a general (N_1N_2,K_1K_2) block code and decoded with a maximum likelihood decoder. The complexity of this decoding strategy is p.o-

hibitively high.

For low complexity decoding, it is natural to decode each column (and then each row) with a lower complexity decoder. Since we want to perform several iterations, we need a soft in/soft out algorithm which provides information on channel bits (not only on information bits). We use the modified Maximum A Posteriori (MAP) algorithm invented by Bahl et al. given in [2]: modified in the sense that the output is not the set of probabilities on information bits, but on the channel bits.

Any linear block code can be decoded with the Bahl algorithm since we can find a trellis associated with the code constructed with the parity check matrix (see Appendix B). For convolutional codes that are used as block codes, we use the usual trellis associated with convolutional codes.

The MAP filtering is first done on each column and then on each row: this corresponds to a single decoding cycle. For the last decoding cycle, we use MAP filtering on columns but MAP decoding on rows (since there is no more iteration to be done).

5.3 Decoding complexity of the 2-dimensional Golay product code

The multirate system uses three information rates: R, $\frac{R}{2}$ and $\frac{R}{4}$ (number of bits per second), while the transmission rate (number of channel symbol per second) is kept constant. A packet corresponds to the same number of channel symbols (it doesn't depend on the information rate), while the number of information bits varies. We have the same amount of time to decode less bits, we can perform more iterations and keep the overall decoding time of a packet constant (without increasing the complexity of the decoder).

Since for rates $\frac{R}{2}$ and $\frac{R}{4}$, there are less information bits, we can use more decoding cycles while keeping the overall decoding time the same: we use 1, 2 and 4 decoding cycles for rates R, $\frac{R}{2}$ and $\frac{R}{4}$, respectively. The complexity of the Bahl algorithm is essentially twice the complexity of the Viterbi algorithm (a forward and a backward

recursion). Here, in the case of the (24,12) extended Golay code, the maximum number of states is $2^{12} = 4098$, but the average number of states per information bit is:

$$\frac{1+2+\ldots+2^{11}+2^{12}+2^{11}+\ldots+2+1}{12} = \frac{2^{12}+2(2^{12}-1)}{12} = \frac{3.2^{12}-2}{12} = 2^{10}-\frac{1}{6} \approx 1024$$

That makes the code very complex to decode, but we know from Kschischang [13] that using parallel processing, we can speed up the decoding by a factor up to 43, which makes its implementation practical. The decoding time per bit becomes comparable to the time required for the convolutional code with $M = 6$. Concerning the decoding of the Golay code, one cycle corresponds to a convolutional code with $M = 7$.

5.4 Computer simulations of the 2-dimensional Golay product code for AWGN channel

Computer simulations are done with 1, 2 and 4 decoding cycles, corresponding to the decoding time available for the three different data rates.

The bit error rate for each of the three code rates are given in Fig. 5.2. These are not the best codes in terms of spectral efficiency and computational complexity, but they can provide coding gains of 5.6 dB, 5.3 dB and 4.3 dB for BER values between 10^{-3} and 10^{-4}, depending on the number of decoding cycles. These are high coding gains for medium BER. The decoding algorithm used is not optimum, but achieves good performance and can be improved with more decoding power.

Note that, as in most iterative decoding systems, the second decoding iteration gives the best improvement: 1 dB, which is significant. The third and fourth iterations only provide an additional 0.2 dB improvement.

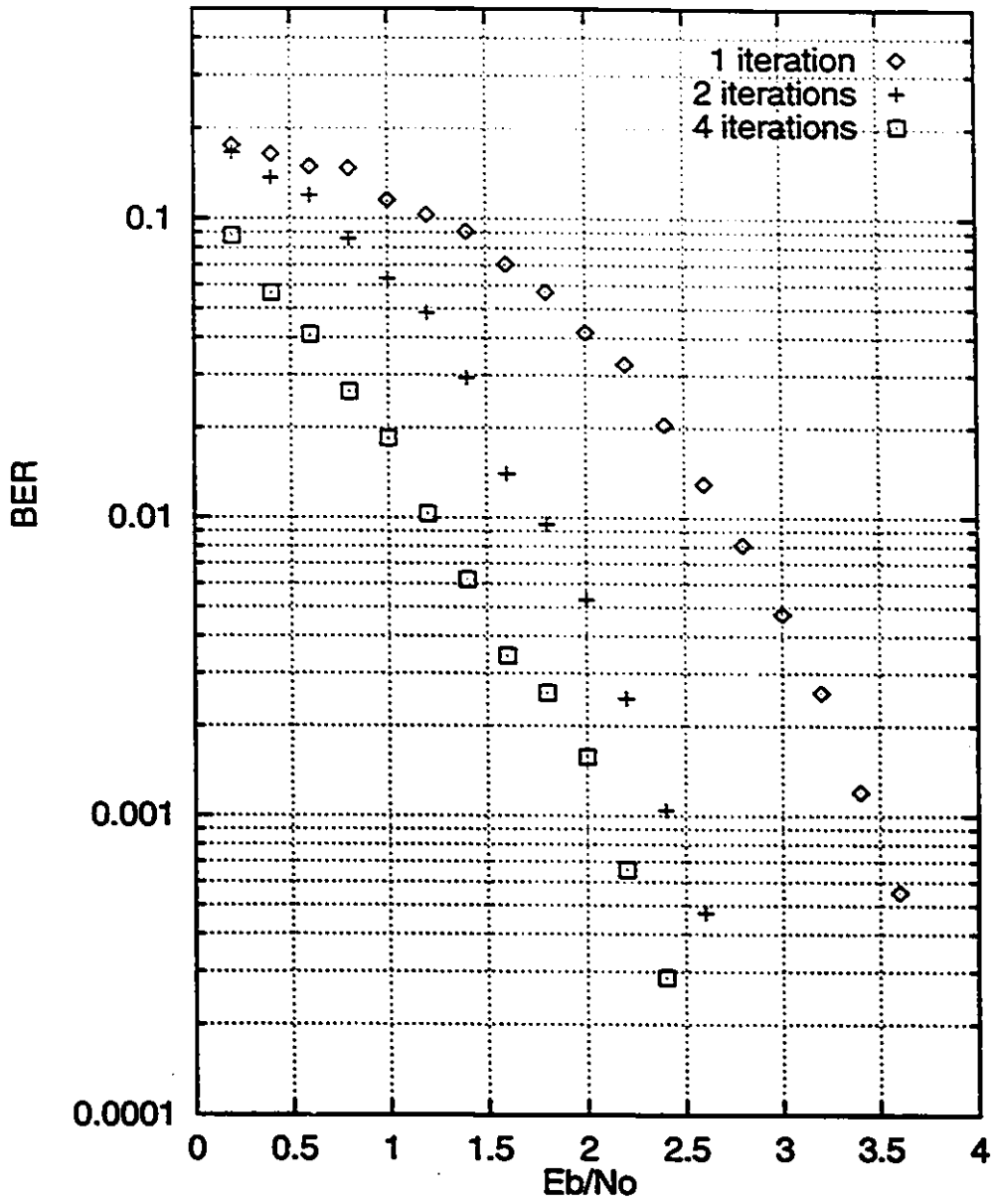

Figure 5.2: *Simulation results of the 2-dimensional Golay product codes on AWGN channel.*

5.5 2-dimensional Golay product code on Rayleigh fading channel

Before doing simulations for the Rayleigh fading channel, we have to know exactly what is the number of bits in a block because interleavering must be done to cope with fading. A three rate coding system will be designed with the 2-dimensional Golay product code.

5.5.1 Block size

The basic block size is determined by the code used: $12 \times 12 = 144$ information bits.

- Low rate $\frac{R}{4}$: 144 information bits (1 basic block).
- Medium rate $\frac{R}{2}$: $144 \times 2 = 288$ information bits (2 basic blocks).
- High rate R : $144 \times 4 = 576$ information bits (4 basic blocks).

Since the code rate is $\frac{1}{4}$, every encoded block is 2304, 1152 or 576 bits depending on the information rate used. The modulation is QPSK: 1152 symbols are sent for rate R. We choose to send the exact same number of symbols for the rates $\frac{R}{2}$ and $\frac{R}{4}$ by repeating the bits 2 or 4 times using an interleaver. This method can be seen as time diversity or as concatenation of a (4,1,4) or a (2,1,2) repetition code. This is one of the most efficient ways to combat fading.

5.5.2 Interleaving

- High rate R :

  The bits are read row by row in each of the 4 blocks. In the stream of bits sent through the channel, bit 1 is coming from block 1, bit 2 is coming from block 2, bit 3 is coming from block 3, bit 4 is coming from block 4, bit 5 is coming from block 1, bit 6 is coming from block 2, ...and so on (see Figure 5.3).

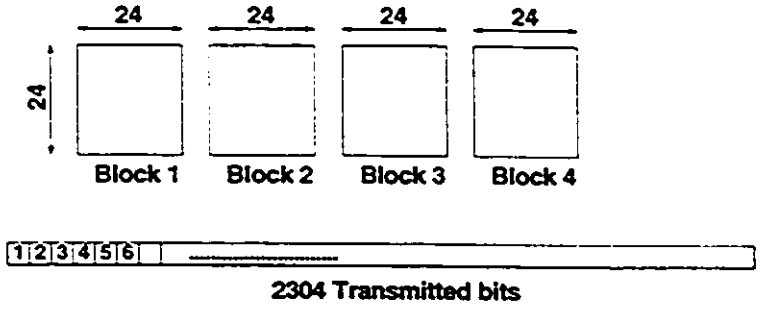

Figure 5.3: *Interleaver for rate R code.*

- Medium rate $\frac{R}{2}$:

  Each channel bit is sent twice: this can be seen as diversity or a (2,1) repetition code concatenated with the 2-dimensional product code. The interleaving is more complicated: instead of sending the bits from blocks 3 and 4, bits from blocks 1 and 2 are sent a second time, but now they are read column by column instead of row by row. This type of interleaving with diversity should be efficient against fading.

- Low rate $\frac{R}{4}$:

  Each channel bit is sent four times. The bits are read row by row, column by column, and in diagonal (in both directions). This type of interleaving should be more efficient against fading than the previous one.

We now take an example to show how the interleaver works. Figures 5.4 and 5.5 show how block 1 bits (from A_1 to A_{476}) and 2 bits (from B_1 to B_{476}) are arranged in the 24×24 array. Block 3 bits (C's) and 4 bits (D's) are arranged exactly the same way. Here, are the transmitted bits for each code rate:

- High rate: $A_1B_1C_1D_1A_2B_2C_2D_2 \ldots A_{476}B_{476}C_{476}D_{476}$.
- Medium rate: $A_1B_1A_1B_1A_2B_2A_{25}B_{25} \ldots A_{476}B_{476}A_{476}B_{476}$.
- Low rate: $A_1A_1A_1A_{453}A_2A_{25}A_{26}A_{430} \ldots A_{476}A_{476}A_{475}A_{23}$.

A_1	A_2	A_3	A_4	...	A_{22}	A_{23}	A_{24}
A_{25}	A_{26}	A_{27}	A_{28}	...	A_{46}	A_{47}	A_{48}
.	.	.	.		.	.	.
.	.	.	.		.	.	.
.	.	.	.		.	.	.
.	.	.	.		.	.	.
A_{429}	A_{430}	A_{431}	A_{432}	...	A_{450}	A_{451}	A_{452}
A_{453}	A_{454}	A_{455}	A_{456}	...	A_{474}	A_{475}	A_{476}

Figure 5.4: *Block 1: bit* A_1 *to* A_{476}

B_1	B_2	B_3	B_4	...	B_{22}	B_{23}	B_{24}
B_{25}	B_{26}	B_{27}	B_{28}	...	B_{46}	B_{47}	B_{48}
.	.	.	.		.	.	.
.	.	.	.		.	.	.
.	.	.	.		.	.	.
.	.	.	.		.	.	.
B_{429}	B_{430}	B_{431}	B_{432}	...	B_{450}	B_{451}	B_{452}
B_{453}	B_{454}	B_{455}	B_{456}	...	B_{474}	B_{475}	B_{476}

Figure 5.5: *Block 2: bit* B_1 *to* B_{476}

5.5.3 Simulations results for the Rayleigh fading channel

The channel under consideration is a frequency non-selective slow fading channel, we assume perfect phase recovery and perfect channel state information (CSI). Since we are working on relatively small blocks, the Doppler frequency is an important parameter of the channel. Simulations were done for 2 typical normalized Doppler frequencies: $f_dT = 0.1$ and $f_dT = 0.01$.

Figures 5.6 and 5.7 show the curves of the BER vs. $\frac{E_b}{N_o}$ for the different rates and normalized Doppler frequencies. A BER = 10^{-3} is achieved for $\frac{E_b}{N_o}$ as low as 3.9 dB with the low rate code and normalized Doppler frequency $f_dT = 0.1$. For the two other codes, a BER = 10^{-3} is achieved for $\frac{E_b}{N_o}$ = 4.7 dB (medium rate) and 7 dB (high rate). These results are very good considering that we are working on small blocks of 144, 288 and 576 information bits.

The results for $f_dT = 0.01$ are not as good as those for $f_dT = 0.1$. A BER = 10^{-3} is achieved for $\frac{E_b}{N_o}$ = 5 dB, 6.2 dB, 8.5 dB, depending on the data rate. When the Doppler frequency is increased by a factor 10, the performances are getting worse by roughly 1.5 dB. This indicates that we can have more gain by using a higher interleaving depth which requires longer blocks to be transmitted. With longer blocks, the encoding/decoding scheme introduces higher systematic delay, which is not tolerable for some applications such as speech transmission.

5.6 Conclusions

In this chapter, product codes have been reviewed: construction and iterative decoding. Computer simulations have been performed for the two-dimensional product code on the AWGN channel where the performances are considered as average; and on the Rayleigh fading channel for which the results are impressive: BER = 10^{-3} is achived for $\frac{E_b}{N_o}$ as low as 3.9 dB.

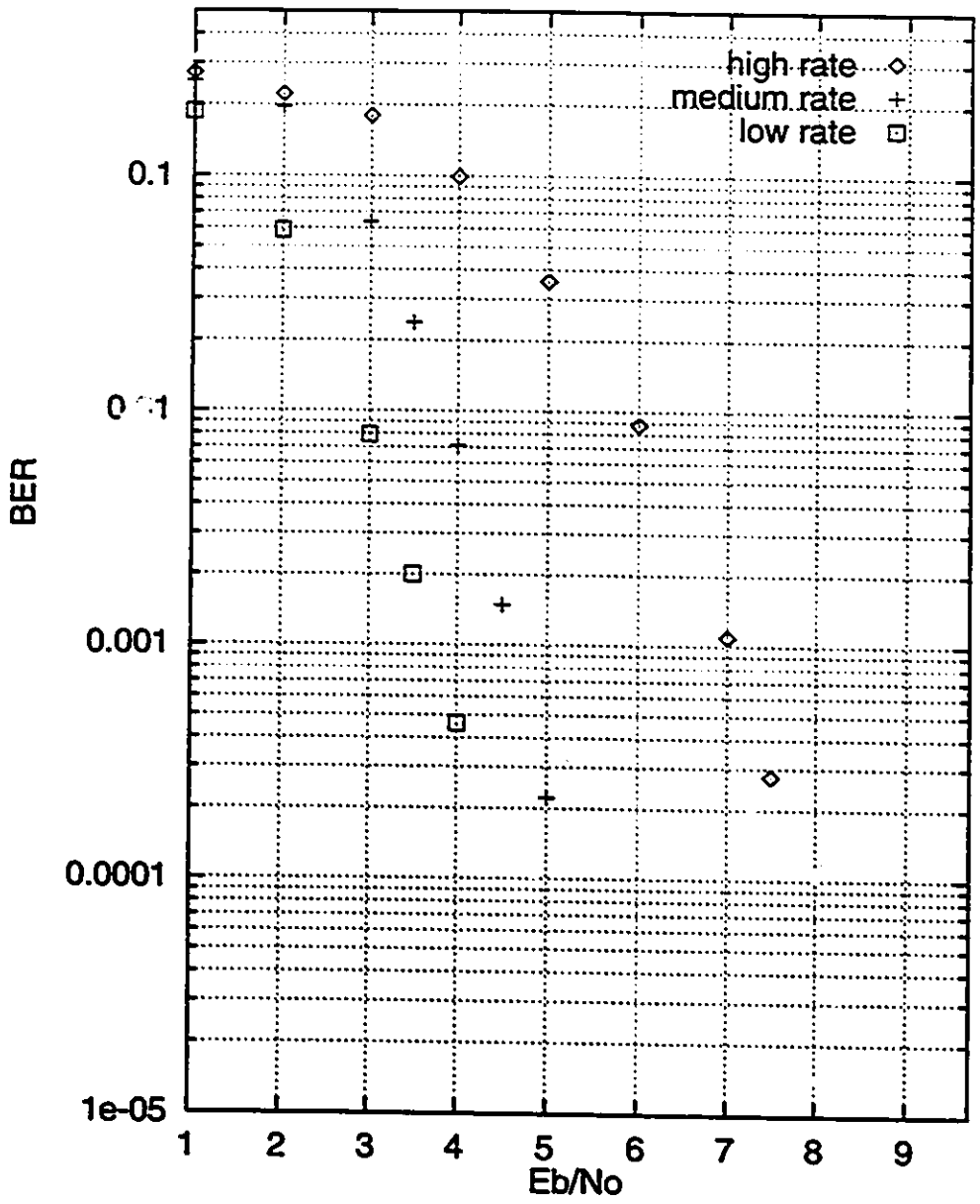

Figure 5.6: *Simulation results of the 2-dimensional Golay product code on Rayleigh fading channel with* $f_dT = 0.1$.

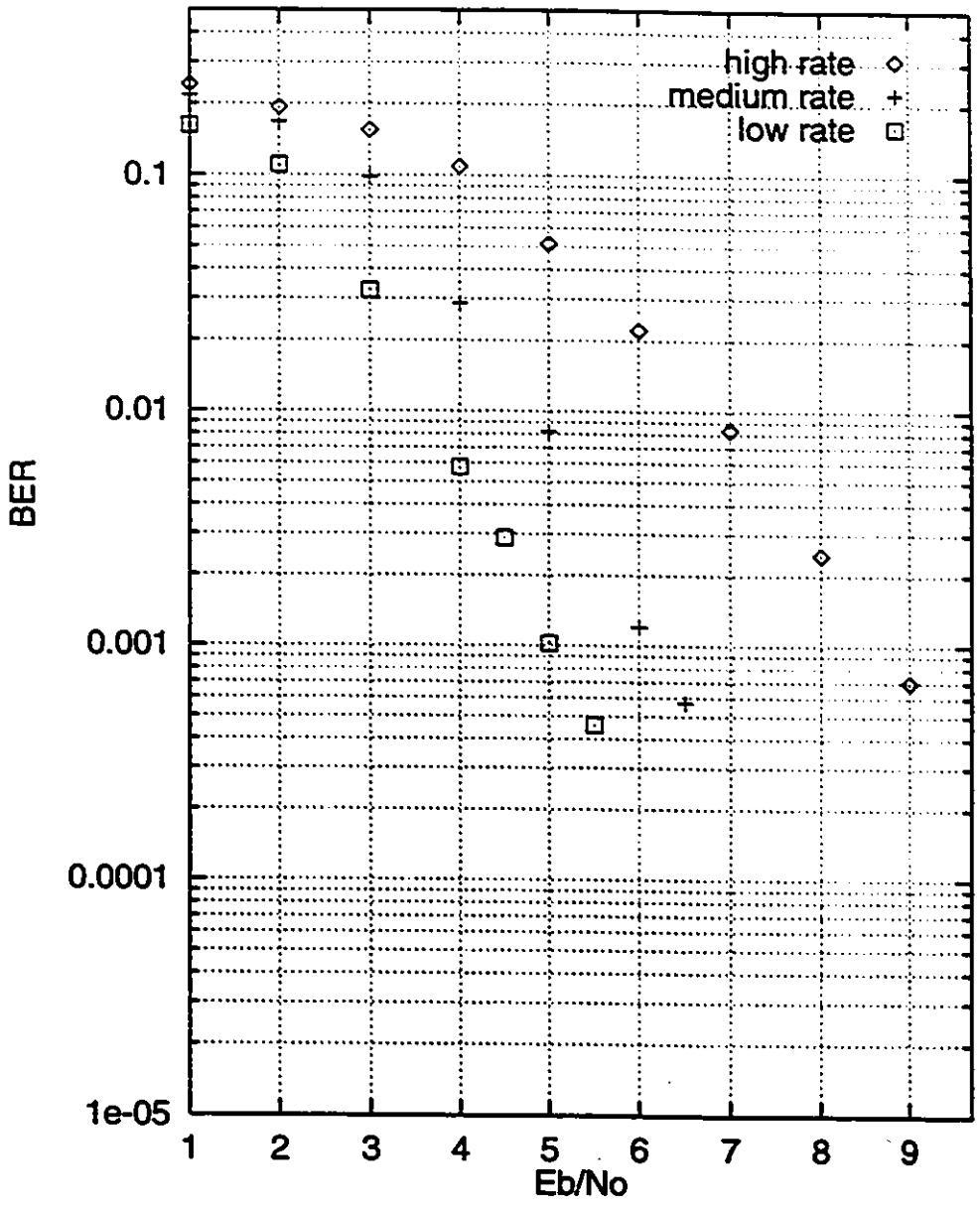

Figure 5.7: *Simulation results of the 2-dimensional Golay product code on Rayleigh fading channel with* $f_dT = 0.01$.

Chapter 6

Performance evaluation of rate adaptive systems for mobile satellite channels

6.1 Performance evaluation procedure for rate adaptive systems

For the comparison of the rate adaptive codes, we need to find a criterion (or several criteria) that indicates which system performs the best on a given channel. We refer back to the channel described in Chapter 2 and consider the three state channel model. To evaluate the performance of a coding system, we simulate it for the three states of the mobile satellite channel presented in Section 2.1.2. To simplify the problem, we assume that when the channel is in a bad state, low data rate is automatically selected, and for average and good channel states, medium and high data rates are used. Performances are evaluated using a cost function. Note that, the concept of quality or reliability, is very important: the idea of adaptive rate is justified by quality argument; it is "better" to send the maximum number of information bits instead of designing the system based on the worst channel state. A performance evaluation can be used to justify the use of a rate adaptive system over a non rate adaptive one.

We already noticed a high dispersion in the values of the various channel parameters depending on the type of environment and elevation angle (see Section 2.1.1). For example, in the literature (e.g. [19]) the Rice fading parameter vary from 5 dB to 17 dB, whereas the attenuation in average and bad states goes from 3 dB to 10 dB (average state) and 12 dB to 18 dB (bad state). Due to the high dispersion of channel parameters (depending on the elevation angle, environment, and even on rain attenuation in the case of K/Ka band) it is very likely that a power adaptive system has to be built.

6.1.1 Adaptive power control

The adaptive power control does not intend to track the fast variations of the channel due to multipath fading, but instead the slow varying shadowing losses due to blockage by buildings or hills which are assumed identical on the uplink and the downlink. Adaptive power control is usually studied on the uplink or reverse link, traditionally the weaker of the two transmission links. Open loop power adaptation has to be considered since even for a non-geostationary low earth orbit satellite (LEOS); the round trip delay is between 10 ms and 60 ms depending on the elevation angle. Power adaptation is done based on the estimation of the received signal power on the down link (or with a pilot tone) by averaging out the fast fading fluctuations on the downlink. In the case of non-geostationary satellites, the power adaptation is also necessary to compensate the path loss that vary with the distance between the mobile and the satellite.

Perfect channel state detection is achieved where the channel state is slowly varying compared to the packet duration. A packet corresponds to a speech spurt of duration up to 60 ms (see Section 2.2.1), compared to the smallest mean state duration for the "average" figures we used (see [19]) which is equal to 15 meters in the bad state on the highway. These 15 m correspond to a duration of 750 ms at a speed of 20 m/s. Then perfect channel state detection can be assumed; it is true for every channel that was measured in [19] since the smallest mean state duration corresponds

to 8 m which translates into 400 ms at a speed of 20 m/s.

We are aware of the fact that perfect power adaptation cannot be achieved (see [21]). A drawback of the simplified three state channel model is that it ignores the slow varying short time average of the received signal power which has to be tracked by the adaptive power control. Monk and Milstein have shown in [21] that the power control error (PCE) has a log-normal distribution with a 0 dB mean and standard deviation of 1 dB for unshadowed users and 2 to 4 dB for shadowed users, depending on the degree of shadowing. For shadowed users, a power margin has to be considered to cope with the large PCE. It has been suggested in [21] that shadowed user power is overcompensated by 3 dB for light shadowing (which corresponds to 2 dB standard deviation of the PCE) and by 6 dB for heavy shadowing (which correspond to 4 dB standard deviation of the PCE). In average and bad channel states, shadowing is present. As a result, 3 dB and 6 dB power margins are provided for average and bad channel states, respectively.

Note that, the problem of power adaptation is also closely related to the capacity of the system in terms of number of users when code division multiple access[1] (CDMA) is used: when we consider the performance of one user, the other users signal become interference. Overcompensated power of shadowed users increases the interference level and decreases the capacity of the system. This issue is only pointed out, it is not studied here. This is very similar to the near-far problem which is well known in cellular network.

6.1.2 Evaluation of the cost function

Since a BER = 10^{-3} or less is achieved in any case because of adaptive power, we will look at the cost in terms of power: we compute the minimum $\frac{E_b}{N_0}$ needed to achieve

[1]The choice between TDMA and CDMA has not been done yet for the third generation systems (see [1]).

a BER=10^{-3} in each state. The cost is related to the average $\frac{E_b}{N_0}$ required[2]. But we need to consider that the code rate is varying, this is the reason why we first compute the average symbol energy $(\frac{E_s}{N_0})_{average}$ with $E_s = \eta E_b$. E_s is used because the number of symbols sent does not depend on the code rate, whereas the number of information bits does depend on the code rate used. Then we normalize it to find the $(\frac{E_b}{N_0})_{average}$.

$(\frac{E_s}{N_0})_{high}$, $(\frac{E_s}{N_0})_{medium}$ and $(\frac{E_s}{N_0})_{low}$ are the ratios needed to achieve BER = 10^{-3} on the corresponding channel, while taking into account the 0, 3 and 6 dB power control error margins for high, medium and low bit rates, respectively.

$$(\frac{E_s}{N_0})_{average} = P(\text{good}) \times (\frac{E_s}{N_0})_{high} + P(\text{average}) \times (\frac{E_s}{N_0})_{medium} + P(\text{bad}) \times (\frac{E_s}{N_0})_{low}$$

$(\frac{E_b}{N_0})_{average}$ is the figure we will use to compare the three rate coding systems. The quality does not depend on the coding system, but only on the channel conditions. More precisely, it depends on the probabilities to be in each state, since low rate is used when the channel is in a bad state, medium and high data rates are used when the channel is in an average and good states, respectively. The higher is the data rate, the better is the quality.

If we want to compare the rate adaptive system to a single rate system, we have to compute $(\frac{E_b}{N_0})_{average}$ with new $(\frac{E_s}{N_0})_{low}$, $(\frac{E_s}{N_0})_{medium}$ and $(\frac{E_s}{N_0})_{high}$. When using a single data rate $(\frac{E_b}{N_0})_{average}$ increases, unless we are using medium or low data rates, in this case the quality decreases.

Note that, an overall bit error is not an interesting figure since the requirement is to achieve BER = 10^{-3} in each channel state. Having a state with a BER $\geq 10^{-3}$ is not acceptable, since it is equivalent to a cut link.

6.2 Presentation of three rate adaptive systems

The rate adaptive codes have to be precisely defined before being evaluated because the mobile satellite channel is a fading channel in each of the three states. On fading

[2]Note that the cost is also linked to the maximum E_s that is decreased when using an adaptive rate system. It will be neglected here.

channels, interleaving is fundamental to cope with bursts of errors. In this section, the codes are carefully described: packet size in terms of the number of information bits, number of channel symbols, interleaving between stages for multilevel codes and symbol interleaving.

6.2.1 Coding system using multilevel codes: first system

When using multilevel codes, we consider the codes described in Section 3.5. We need to be more precise on the description of the codes. $N = 360$ channel symbols are sent in the three cases. In convolutional codes, when the reset bits to end up in the zero state are considered, the actual number of information bits and the spectral efficiency slightly decrease.

Coded modulation scheme 1

This coded modulation is denoted by CM_{11}.

- Stage 1: $r_1 = \frac{1}{3}$, $M = 5$ convolutional code. The number of information bits is: $K_1 = r_1 N - M = 360 \times \frac{1}{3} - 5 = 115$.
- Stage 2: $r_2 = \frac{3}{4}$, $M = 5$ convolutional code. The number of information bits is: $K_2 = r_2 N - M = 360 \times \frac{3}{4} - 5 = 265$.
- Stage 3: $r_3 = \frac{9}{10}$, $M = 5$ convolutional code. The number of information bits is: $K_3 = r_3 N - M = 360 \times \frac{9}{10} - 5 = 319$.
- Stage 4: $r_4 = \frac{11}{12}$, $M = 5$ convolutional code. The number of information bits is: $K_4 = r_4 N - M = 360 \times \frac{11}{12} - 5 = 325$.
- Stage 5: $r_5 = 1$ uncoded bits. The number of information bits is: $K_5 = 360$.

Then $K_1 + K_2 + K_3 + K_4 + K_5 = 115 + 265 + 319 + 325 + 360 = 1384$ information bits are sent. A block interleaver is used: bits from stage 1 are written column by column, bits from stage 2 are written row by row, bits from stage 3 are written in

diagonal, bits from stage 4 are written in diagonal (second diagonal). The way bits from stage 5 are written is not relevant since they are uncoded. Symbols are read column by column before being sent over the channel such that an error burst can be distributed among the channel bits. We chose to read the symbols the same way stage 1 bits were written, because they are strongly protected against fading due to the high minimum distance of the code C_1.

We use Figure 6.1 to explain how the interleaver works:

- Stage 1 bits are written on $S_1, S_{21}, S_{41}, S_{61}, S_{81}, S_{101}\ldots$
- Stage 2 bits are written on $S_1, S_2, S_3, S_4, S_5, S_6\ldots$
- Stage 3 bits are written on $S_{341}, S_{322}, S_{303}, S_{284}, S_{265}, S_{246}, S_{227}\ldots$
- Stage 4 bits are written on $S_1, S_{22}, S_{43}, S_{64}, S_{85}, S_{106}\ldots$
- Symbols are written as follows: $S_1, S_{21}, S_{41}, S_{61}, S_{81}, S_{101}\ldots$

S_1	S_2	S_3	S_4	...	S_{17}	S_{18}	S_{19}	S_{20}
S_{21}	S_{22}	S_{23}	S_{24}	...	S_{37}	S_{38}	S_{39}	S_{40}
S_{41}	S_{42}	S_{43}	S_{44}	...	S_{57}	S_{58}	S_{59}	S_{60}
S_{61}	S_{62}	S_{63}	S_{64}	...	S_{77}	S_{78}	S_{79}	S_{80}
S_{81}	S_{82}	S_{83}	S_{84}	...	S_{97}	S_{98}	S_{99}	S_{100}
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
S_{281}	S_{282}	S_{283}	S_{284}	...	S_{297}	S_{298}	S_{299}	S_{300}
S_{301}	S_{302}	S_{303}	S_{304}	...	S_{317}	S_{318}	S_{319}	S_{320}
S_{321}	S_{322}	S_{323}	S_{324}	...	S_{337}	S_{338}	S_{339}	S_{340}
S_{341}	S_{342}	S_{343}	S_{344}	...	S_{357}	S_{358}	S_{359}	S_{360}

Figure 6.1: *Interleaver: symbol transmitted.*

Coded modulation scheme 2

This coded modulation will be denoted by CM_{12}.

- Stage 1: $r_1 = \frac{1}{3}$, $M = 5$ convolutional code. The number of information bits is: $K_1 = r_1 N - M = 360 \times \frac{1}{3} - 5 = 115$.
- Stage 2: $r_2 = \frac{2}{3}$, $M = 5$ convolutional code. The number of information bits is: $K_2 = r_2 N - M = 360 \times \frac{2}{3} - 5 = 235$.
- Stage 3: $r_3 = \frac{9}{10}$, $M = 5$ convolutional code. The number of information bits is: $K_3 = r_3 N - M = 360 \times \frac{9}{10} - 5 = 319$.

Then $K_1 + K_2 + K_3 = 115 + 235 + 319 = 669$ information bits are sent. A block interleaver is used: bits from stage 1 are written column by column, bits from stage 2 are written row by row, bits from stage 3 are written in diagonal and symbols are read in diagonal (second diagonal) before being sent over the channel such that an error burst can be distributed among the channel bits.

We use Figure 6.1 again to explain how the interleaver works:

- Stage 1 bits are written on S_1, S_{21}, S_{41}, S_{61}, S_{81}, S_{101}...
- Stage 2 bits are written on S_1, S_2, S_3, S_4, S_5, S_6...
- Stage 3 bits are written on S_{341}, S_{322}, S_{303}, S_{284}, S_{265}, S_{246}, S_{227} ...
- Symbols are written in the interleaver as follows: S_1, S_{22}, S_{43}, S_{64}, S_{85}, S_{106}...

Coded modulation scheme 3

This coded modulation is denoted by CM_{13}.

- Stage 1: $r_1 = \frac{1}{3}$, $M = 5$ convolutional code. The number of information bits is: $K_1 = r_1 N - M = 360 \times \frac{1}{3} - 5 = 115$.
- Stage 2: $r_2 = \frac{2}{3}$, $M = 5$ convolutional code. The number of information bits is: $K_2 = r_2 N - M = 360 \times \frac{2}{3} - 5 = 235$.

Then $K_1 + K_2 = 115 + 235 = 350$ information bits are sent. Channel symbols have to be interleaved because they are sent over fading channels. A block interleaver is used: bits from stage 1 are written column by column, bits from stage 2 are written row by row and symbols are read in diagonal before being sent over the channel such that an error burst can be distributed among the channel bits.

We use again Figure 6.1 to explain how the interleaver works:

- Stage 1 bits are written on S_1, S_{21}, S_{41}, S_{61}, S_{81}, S_{101}...
- Stage 2 bits are written on S_1, S_2, S_3, S_4, S_5, S_6...
- Symbols are written in the interleaver as follows: S_1, S_{22}, S_{43}, S_{64}, S_{85}, S_{106}...

6.2.2 Coding system using the 2-dimensional Golay product code: second system

This is exactly the system previously described in Section 5.5. The spectral efficiencies of these codes are: $\frac{1}{2}$, $\frac{1}{4}$ and $\frac{1}{8}$ bits/symbol. The codes are denoted by CM_{21}, CM_{22} and CM_{23}, respectively. The weakness of this system is the very low spectral efficiency compared to the spectral efficiency of the coded modulation schemes.

6.2.3 Coding system using Golay code and 2-dimensional Golay product code: third system

We know how powerful the 2-dimensional Golay product code can be. The system is designed using this code without time diversity for the low rate code.

- The high rate uses uncoded QPSK with 288 channel symbols and 576 information bits, denoted CM_{31}.
- The medium rate code uses Golay code followed by an interleaver to combat fading. 24 codewords are used and mapped on 288 QPSK channel symbols, denoted CM_{32}.

- The low rate code uses 2-dimensional Golay product code interleaved, denoted CM_{33}.

We need also to specify the two interleavers:

- The medium rate interleaver is a block interleaver: the 24 bit codewords are written in a 24×24 block row by row and the bits are read column by column before being mapped to a QPSK symbol. Two bits which form the same codeword are 24 bits away, which makes them 12 channel symbols away.
- The high rate interleaver is as follows: The codeword is already written in a 24×24 array, bits are read in diagonal before being sent on the channel.

The spectral efficiencies are 2, 1 and $\frac{1}{2}$ bits/symbols.

6.3 Performance evaluation of the systems

In this section, simulation results are presented for the codes described in previous section. Rice fading channel with $c = 10.6$ dB and 16.6 dB correspond to the good state channel in an urban environment and on highway environment, for the high rate code. Rayleigh fading channel for medium and low rate code corresponds to the average and bad state channels, corrections with the attenuation and power control error will be done afterward.

6.3.1 First system

CM_{11} is evaluated on the Rice fading channel through computer simulations, Figure 6.2 shows the results for $c = 10.6$ dB and $c = 16.6$ dB. These results are very similar and do not depend much on the normalized Doppler frequency and the Rice factor c. A BER $= 10^{-3}$ is achieved for $\frac{E_b}{N_0} = 8.8$ dB for $c = 10.6$ dB, and $\frac{E_b}{N_0} = 8.7$ dB for $c = 16.6$ dB.

As a general rule, normalized Doppler frequency does not strongly affect the performance on Rice fading channel with high Rice factor ($c \geq 10$ dB). In this case, the

direct path component is at least 10 dB above the multipath component. Fading with zero crossing is nonexistent and deep fades are very unlikely to happen. Therefore the symbols are never considered "erased" and no error bursts occur whatever is the normalized Doppler frequency. Usually on Rayleigh channels the length of the error bursts increases as the normalized Doppler frequency decreases. Then the performance on Rice fading channel with high Rice factor does not depend much on the normalized Doppler frequency as opposed to the Rayleigh fading channel.

CM_{12} and CM_{13} are simulated on the Rayleigh fading channel, Figure 6.3 shows the simulation results. CM_{13} is very efficient in Rayleigh fading channel: it can achieve a BER $= 10^{-3}$ with $\frac{E_b}{N_0} = 6.6$ dB or 7 dB (with $f_dT = 0.1$ and $f_dT = 0.01$, respectively). This is due to the large minimum Hamming distance of the coded modulation: $\delta_H[CM_{13}] = 6$. It is very surprising that the performances are better for $f_dT = 0.01$ than for $f_dT = 0.1$ in the range of interest (i.e., for a BER $\geq 10^{-4}$). For a BER $\leq 10^{-4}$, as expected, the slower channel ($f_dT = 0.01$) is worse.

CM_{12} is very efficient when $f_dT = 0.1$, a BER $= 10^{-3}$ is achieved for $\frac{E_b}{N_0} = 8.2$ dB. This is very good compared to 11 dB achieved by the coded modulation given in [22] or to 10 dB achieved in [32], both for 2 bits/symbol spectral efficiency on the Rayleigh fading channel. But for $f_dT = 0.01$, the performance deteriorates dramatically: by 4.5 dB for a BER $= 10^{-3}$. This demonstrates that the interleaving depth is not sufficient for a minimum Hamming distance as small as $\delta_H[CM_{12}] = 3$ when the fading process is slow. There are two solutions to improve the performance: either using a larger interleaver or selecting a stronger code for the last level bits.

6.3.2 Second system

CM_{21} is simulated on Rice fading channels, Figure 6.4 shows the results for $c =$ 10.6 dB and $c = 16.6$ dB and $f_dT = 0.1$ and $f_dT = 0.01$. Note that, the effect of Doppler frequency does not have a strong effect on the performances: less than 0.1 dB is lost for BER $\leq 10^{-3}$ when $f_dT = 0.01$. This was predictable since the channel fades are not very deep: most of the energy is coming from the direct path

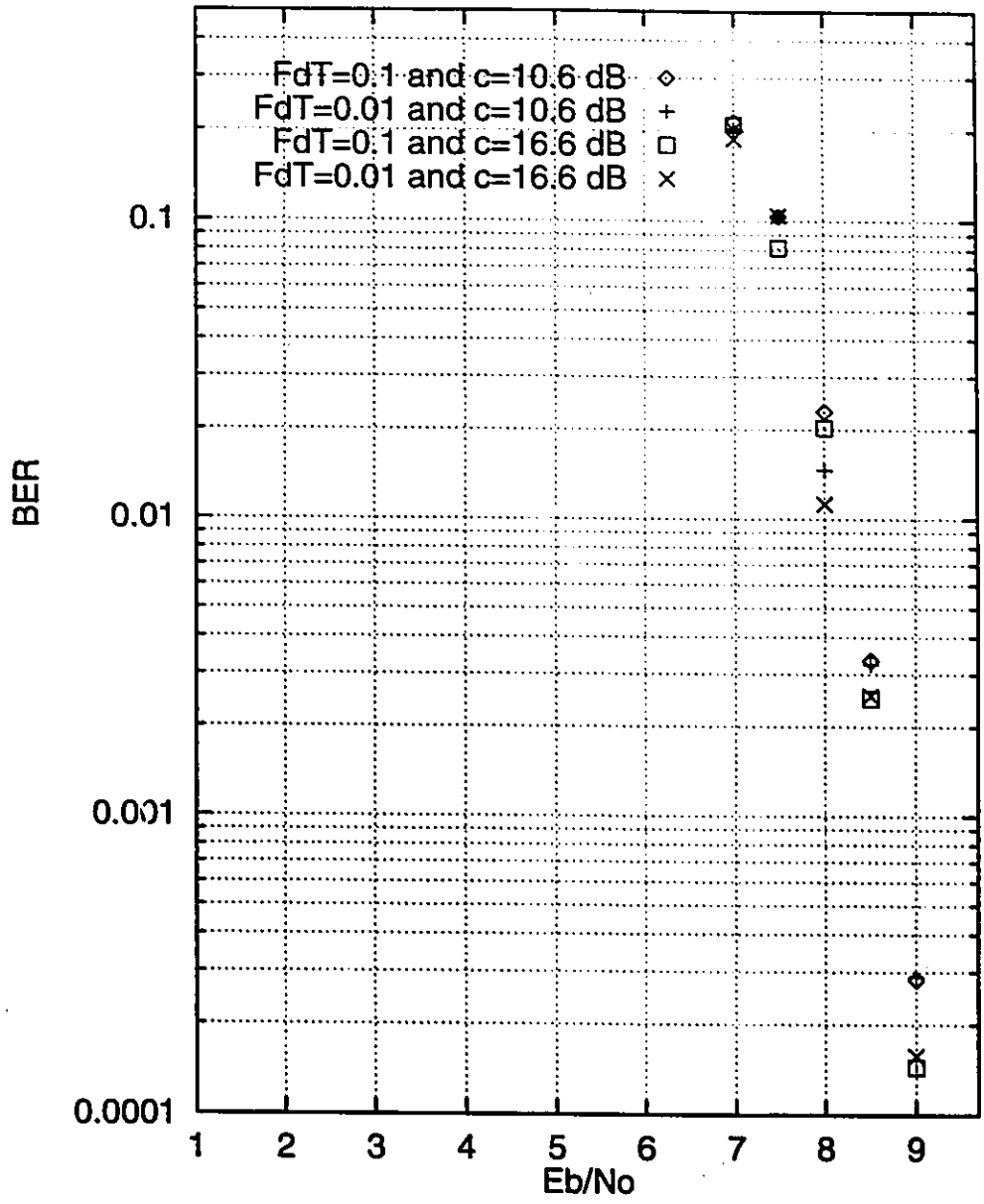

Figure 6.2: *CM_{11} (3.9 bit/symbol and 32APK modulation) simulated on Rice fading channel with $c = 10.6$ dB or $c = 16.6$ dB, $f_dT = 0.01$ or $f_dT = 0.1$.*

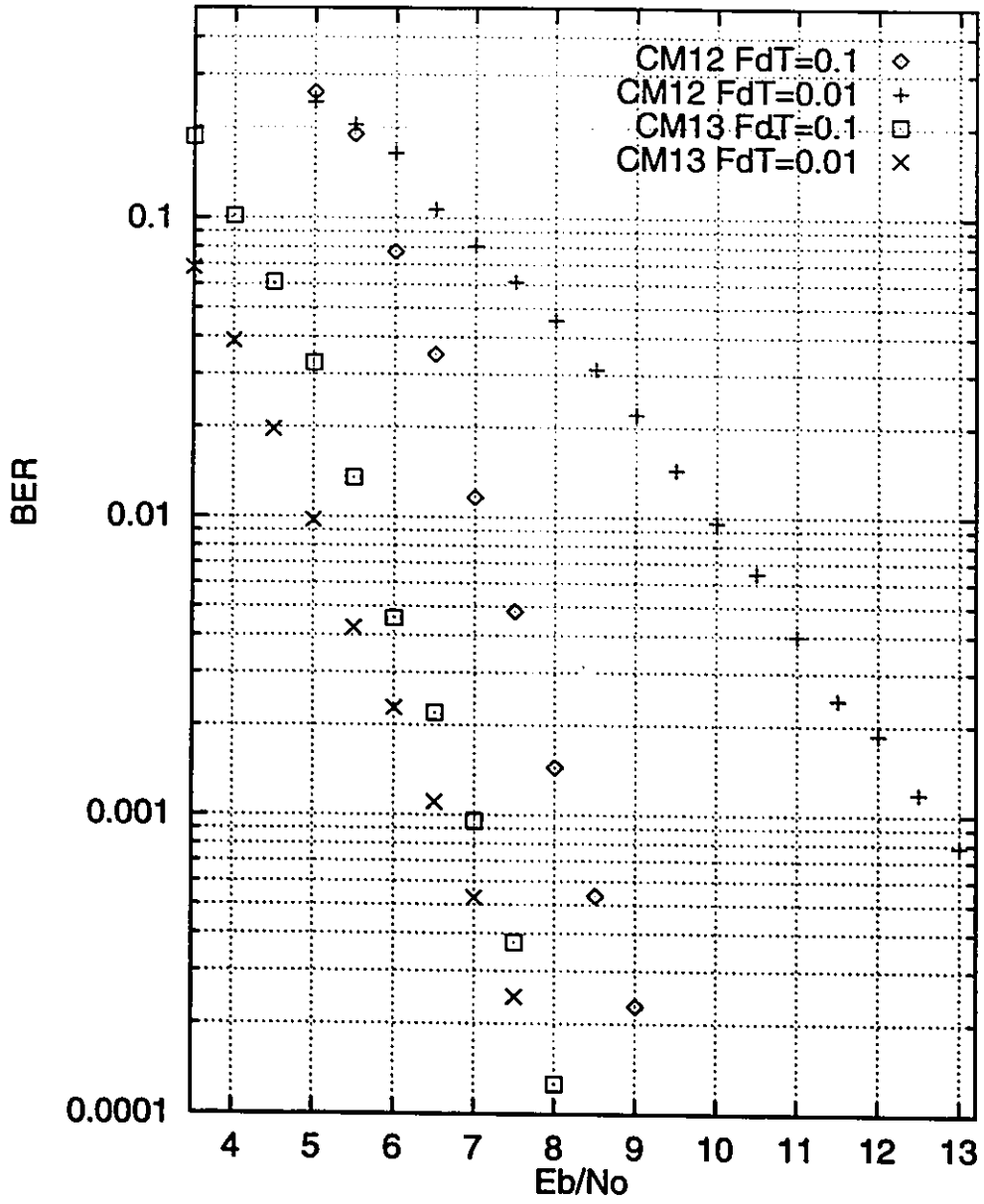

Figure 6.3: *CM_{12} (1.9 bit/symbol and 8APK constellation) and CM_{13} (1.0 bit/symbol and QPSK constellation) simulated on Rayleigh fading channel with $f_dT = 0.01$ or $f_dT = 0.1$.*

component. For c = 10.6 dB, the BER = 10^{-3} for $\frac{E_b}{N_0}$ = 3.3 to 3.4 dB depending on the normalized Doppler frequency. For c = 16.6 dB, a BER=10^{-3} is achieved for $\frac{E_b}{N_0}$ = 3.2 to 3.3 dB depending on the normalized Doppler frequency. Note that, the results are even better than those on the AWGN by 0.1 dB, such a result has already been observed in low signal to noise conditions for high Rice factor in [16].

Figure 6.5 shows simulation results for CM_{22} and CM_{23} on the Rayleigh fading channel that were already presented in Section 5.5. the BER = 10^{-3} at $\frac{E_b}{N_0}$ = 3.9 dB and 5 dB for CM_{23} for f_dT = 0.1 and f_dT = 0.01, respectively. For CM_{22}, a BER = 10^{-3} is achieved for $\frac{E_b}{N_0}$ = 4.7 dB and 6.2 dB depending on the normalized Doppler frequency.

6.3.3 Third system

CM_{31} is simulated on the Rice fading channel with c = 10.6 dB and c = 16.6 dB. Figure 6.6 shows the results of the computer simulations. The Doppler frequency does not affect the bit error probability since the bits are uncoded. The Rice factor c strongly affects the bit error probability: the BER = 10^{-3} when $\frac{E_b}{N_0}$ = 11.4 dB for c = 10.6 dB and $\frac{E_b}{N_0}$ = 7.6 dB for c = 16.6 dB.

Figure 6.7 presents simulation results for CM_{32} and CM_{33} on the Rayleigh fading channel with f_dT = 0.1 and f_dT = 0.01. Here, the results are very sensitive to the Doppler frequency: it shows that a larger interleaver could improve the performance. For CM_{32}, a BER = 10^{-3} is achieved for $\frac{E_b}{N_0}$ = 9.7 dB and 12 dB depending on the normalized Doppler frequency. For CM_{33}, BER = 10^{-3} at $\frac{E_b}{N_0}$ = 6.1 dB and 7.9 dB depending on the normalized Doppler frequency. These are not bad results when we consider that the low decoding complexity of the codes: there is only 1.5 dB and 1.0 dB difference between CM_{32} and CM_{12} (both with spectral efficiency of 1 bit/symbol).

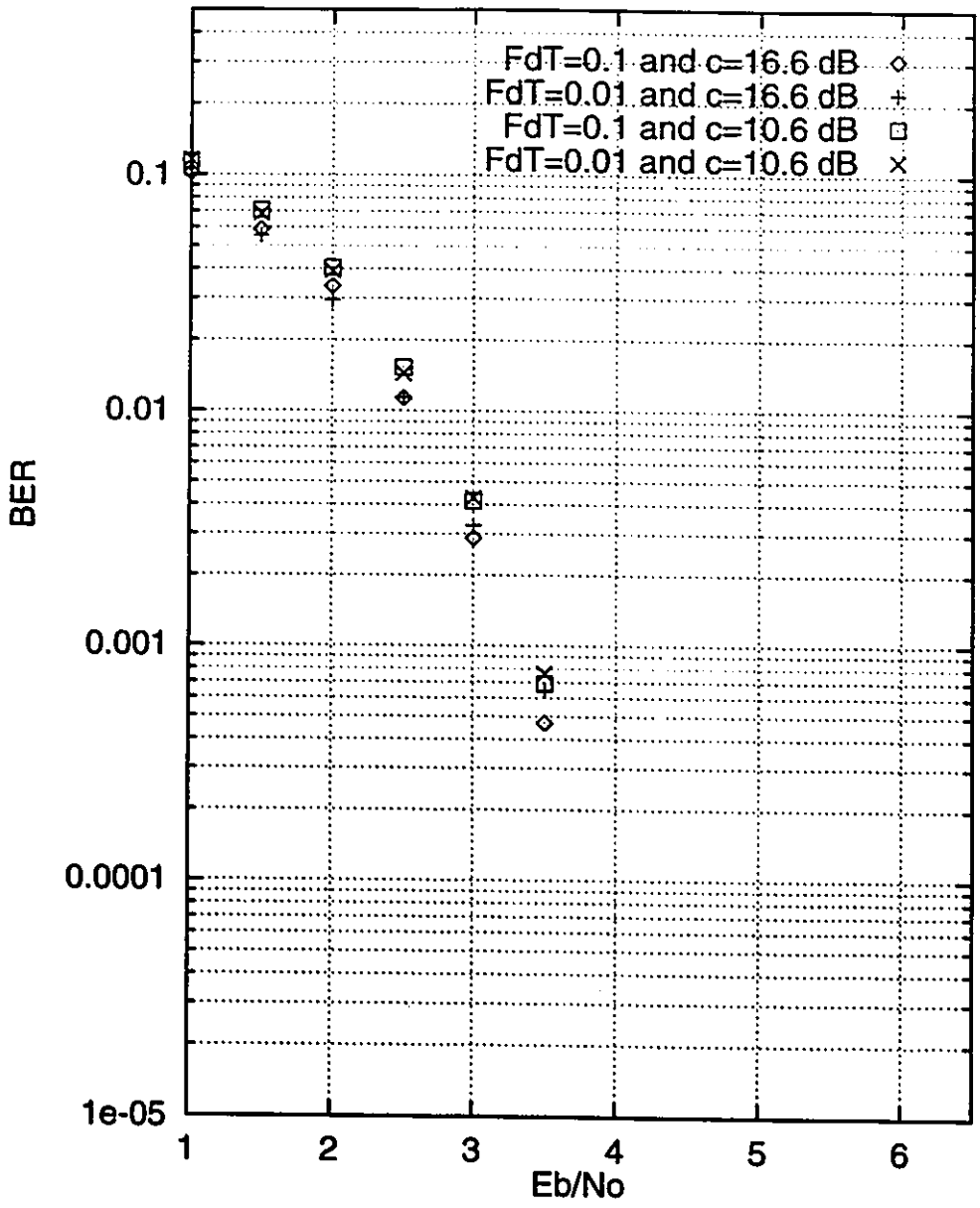

Figure 6.4: *CM_{21} (0.5 bit/symbol and QPSK constellation) simulated on Rice fading channel with* $c = 10.6$ *dB or* $c = 16.6$ *dB,* $f_dT = 0.01$ *or* $f_dT = 0.1$.

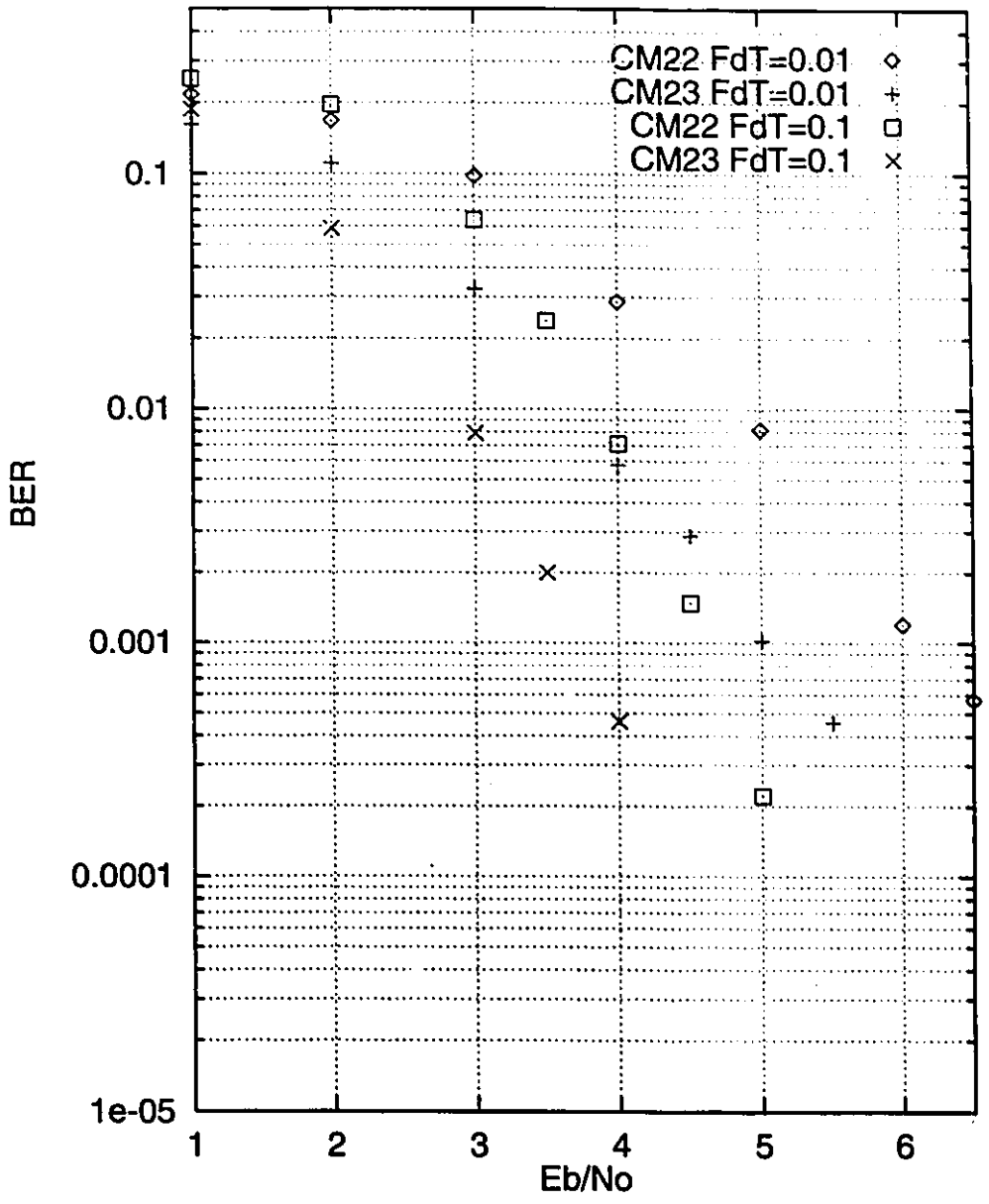

Figure 6.5: *CM_{22} (0.25 bit/symbol and QPSK constellation) and CM_{23} (0.125 bit/symbol and QPSK constellation) simulated on Rayleigh fading channel with $f_dT =$* 0.01 *or* $f_dT = 0.1$.

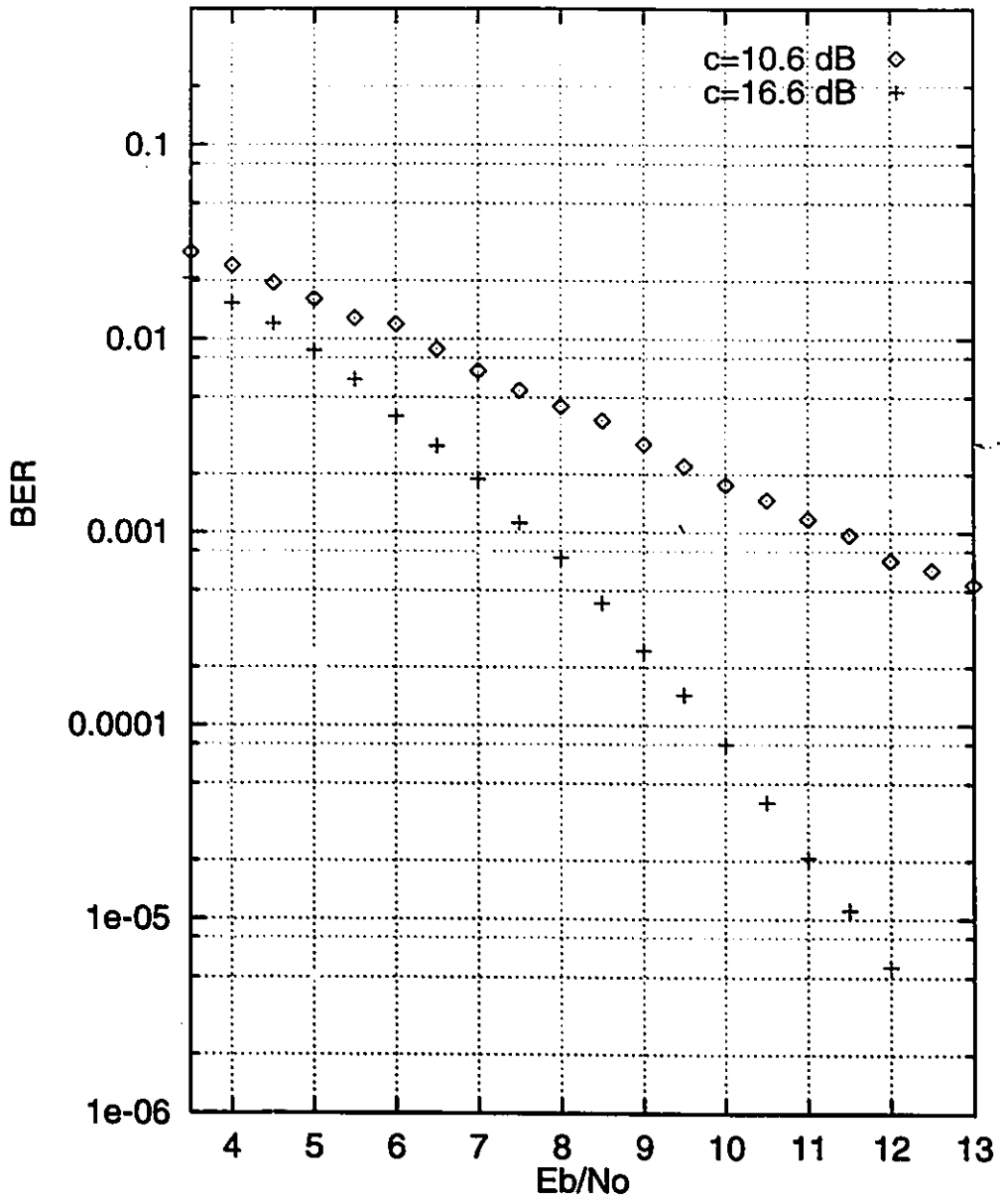

Figure 6.6: *CM_{31} (2 bit/symbol uncoded QPSK constellation) simulated on Rice fading channel with $c = 10.6$ dB or $c = 16.6$ dB.*

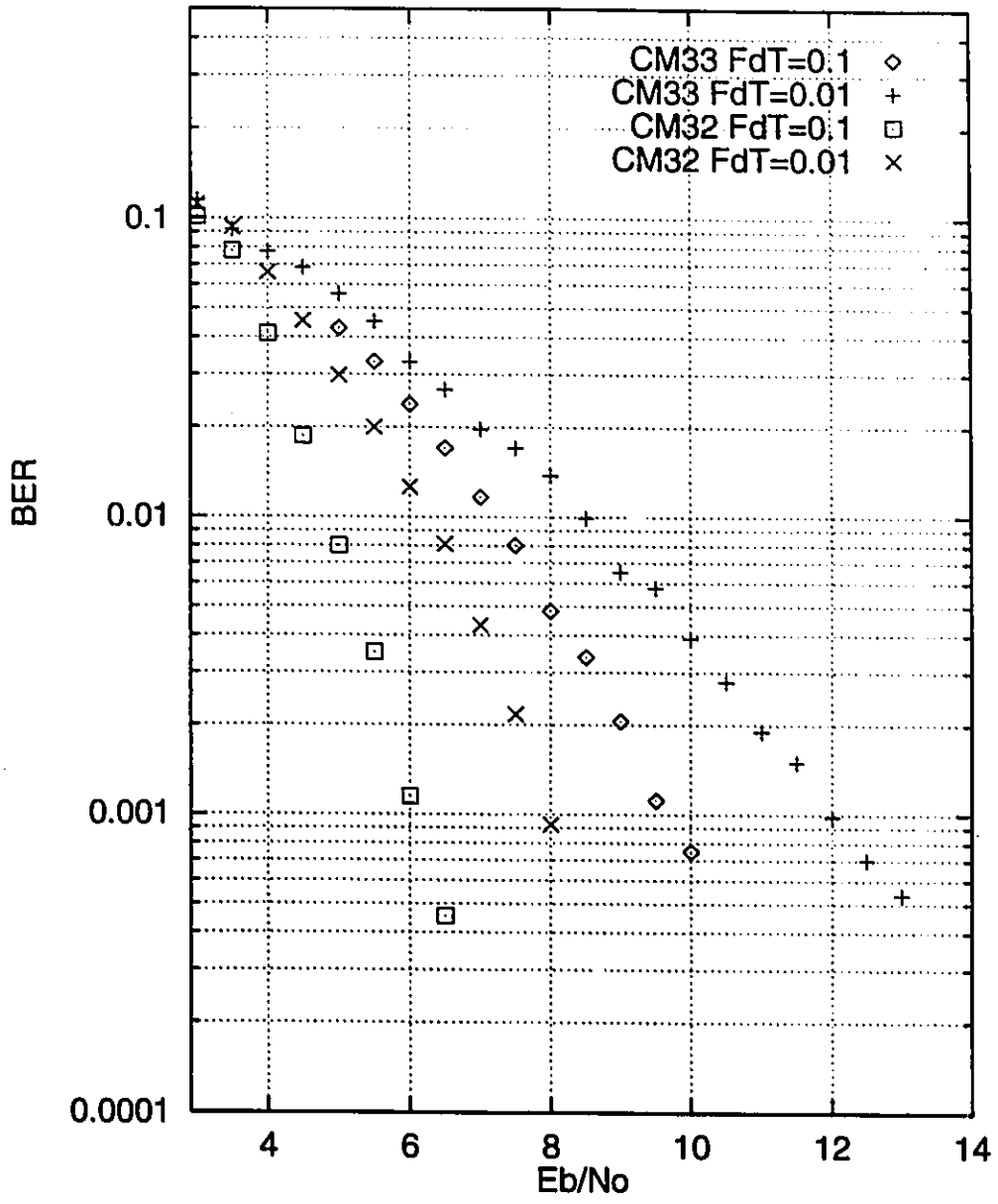

Figure 6.7: *CM_{32} (1.0 bit/symbol and QPSK constellation) and CM_{33} (0.5 bit/symbol and QPSK constellation) simulated on Rayleigh fading channel with $f_dT = 0.1$ and $f_dT = 0.01$.*

6.3.4 Computation of the cost function for each system

Table 6.1 presents the $\frac{E_b}{N_0}$ ratio needed to achieve a BER of 10^{-3} for each channel state, normalized Doppler frequency and coding system.

Table 6.1: *Recapitulation of the simulation results:* $\frac{E_b}{N_0}$ *(in dB) needed to achieve a* $BER = 10^{-3}$.

Coded Modulation	$CM_{\cdot 1}$				$CM_{\cdot 2}$		$CM_{\cdot 3}$	
Channel	Rice fading				Rayleigh fading			
	$c = 10.6$ dB		$c = 16.6$ dB					
f_dT	0.1	0.01	0.1	0.01	0.1	0.01	0.1	0.01
First system	8.8	8.8	8.7	8.7	8.2	12.7	6.7	6.6
Second system	3.3	3.4	3.2	3.3	4.7	6.2	3.9	5.0
Third system	11.4	11.4	7.6	7.6	9.7	12.0	6.1	7.9

Recall that the spectral efficiencies of first, second and third systems are (4,2,1), ($\frac{1}{2}$,$\frac{1}{4}$,$\frac{1}{8}$) and (2,1,$\frac{1}{2}$) bits/symbol, respectively. The performances should improve as the spectral efficiency decreases. Results of coded modulation CM_{13} (1 bit/symbol) on the Rayleigh fading channel is very impressive: the results are better than CM_{33} ($\frac{1}{2}$ bit/symbol) and only 1.9 dB or 0.5 dB (depending on f_dT) away form CM_{12} ($\frac{1}{4}$ bit/symbol).

Note that, the performance of the third system are about the same as the one as the first in spite of a lower spectral efficiency. It can be explained by a reduced complexity and small interleaver size (288 instead of 360 channel symbols) that affect the efficiency of the system.

Then we can put the results of Table 6.1 in terms of the required $\frac{E_s}{N_0}$ and take into account the power margins to cope with power control error (0, 3 and 6 dB). This gives Table 6.2 where the losses due to Rayleigh fading are not taken into account yet (Table 6.2 can't be used directly to compare the performances of the codes since E_s depends on the spectral efficiency).

Table 6.2: *Recapitulation of the simulation results:* $\frac{E_s}{N_0}$ *(in dB) needed to achieve a* $BER = 10^{-3}$ *with power control error overcompensation.*

Coded Modulation	$CM_{\cdot 1}$				$CM_{\cdot 2}$		$CM_{\cdot 3}$	
Channel	Rice fading				Rayleigh fading			
	$c = 10.6$ dB		$c = 16.6$ dB					
f_dT	0.1	0.01	0.1	0.01	0.1	0.01	0.1	0.01
First system	14.7	14.7	14.6	14.6	14.0	18.5	12.7	12.6
Second system	0.3	0.4	0.2	0.3	1.7	3.2	0.9	2.0
Third system	14.4	14.4	10.6	10.6	12.7	15.0	9.1	10.9

We compute the $(\frac{E_s}{N_0})_{average}$ for the two typical channels: highway environment or urban environment and the two different normalized Doppler frequencies. We should keep in mind that we have to consider the attenuation due to fading for the average and bad states (see Section 2.1.2).

The results for the highway environment are:

1. Highway environment, $f_dT = 0.1$ with the first system:
   $(\frac{E_s}{N_0})_{average}$= 15.3 dB. $(\frac{E_b}{N_0})_{average}$= 9.5 dB.

2. Highway environment, $f_dT = 0.01$ with the first system:
   $(\frac{E_s}{N_0})_{average}$=15.4 dB. $(\frac{E_b}{N_0})_{average}$=9.6 dB.

3. Highway environment, $f_dT = 0.1$ with the second system:
   $(\frac{E_s}{N_0})_{average}$= 1.4 dB. $(\frac{E_b}{N_0})_{average}$= 4.4 dB.

4. Highway environment, $f_dT = 0.01$ with the second system:
   $(\frac{E_s}{N_0})_{average}$= 1.8 dB. $(\frac{E_b}{N_0})_{average}$= 4.8 dB.

5. Highway environment, $f_dT = 0.1$ with the third system:
   $(\frac{E_s}{N_0})_{average}$= 10.2 dB. $(\frac{E_b}{N_0})_{average}$= 11.4 dB.

6. Highway environment, $f_dT = 0.01$ with the third system:
   $(\frac{E_s}{N_0})_{average}$= 10.6 dB. $(\frac{E_b}{N_0})_{average}$= 11.8 dB.

The results with a city (i.e. urban) environment are:

1. City environment, $f_dT = 0.1$ with the first system:
   $(\frac{E_s}{N_0})_{average}$= 25.1 dB. $(\frac{E_b}{N_0})_{average}$= 19.2 dB.

2. City environment, $f_dT = 0.01$ with the first system:
   $(\frac{E_s}{N_0})_{average}$= 26.0 dB. $(\frac{E_b}{N_0})_{average}$= 20.1 dB.

3. City environment, $f_dT = 0.1$ with the second system:
   $(\frac{E_s}{N_0})_{average}$= 13.4 dB. $(\frac{E_b}{N_0})_{average}$= 16.4 dB.

4. City environment, $f_dT = 0.01$ with the second system:
   $(\frac{E_s}{N_0})_{average}$= 14.5 dB. $(\frac{E_b}{N_0})_{average}$= 17.5 dB.

5. City environment, $f_dT = 0.1$ with the third system:
   $(\frac{E_s}{N_0})_{average}$= 22.0 dB. $(\frac{E_b}{N_0})_{average}$= 21.9 dB.

6. City environment, $f_dT = 0.01$ with the third system:
   $(\frac{E_s}{N_0})_{average}$= 23.8 dB. $(\frac{E_b}{N_0})_{average}$= 23.7 dB.

We could expect the performances to get better as the spectral efficiency decreases, but system 1 is better than system 3 even though its spectral efficiency is higher. In highway environment, system 1 is 1.9 dB and 2.2 dB better than system 3 (depending on the normalized Doppler frequency). In city environments (severe fading channel), the difference between systems 1 and 3 increases: 2.7 dB and 3.6 dB. Then the high spectral efficiency system becomes very attractive. In highway conditions, the Doppler frequency does not have a strong effect on the required energy: the second and third systems present the biggest performance deterioration, only 0.4 dB. It can be explained since the good state is "dominant" (that is much more probable than the other states) and in Rice fading channel with a Rice factor $c \geq 10$ dB, the effect of the Doppler frequency is almost negligible: this can be seen in Table 6.1 or in

Figures 6.2, 6.4 and 6.6. On the contrary, in a city environment, the performances strongly depend on Doppler frequency. The penalties are 0.9 dB, 1.1 dB and 1.8 dB depending on the system used.

The advantage of using a multirate will become clear here. For a single rate error control coding system using CM_{21} (with information rate R) in each channel state, for $f_dT = 0.1$ with a highway environment, $(\frac{E_s}{N_0})_{average}$ = 6.4 dB compared to the 1.4 dB when using an adaptive rate system: adaptive rate here saves 5.0 dB of power and does affect the quality of the transmission only 3% of the time. In a city environment, a single rate error control coding system using CM_{21} as the unique code and information rate R, using CM_{21} in each channel state: $(\frac{E_s}{N_0})_{average}$ = 22.2 dB instead of 13.4 dB for the rate adaptive system. The gain is very large: 8.8 dB, but the quality is lower 57% of the time, this is the price to pay for this 8.8 dB improvement. As usual, the best improvement is achieved on the most severe channel.

What is the advantage of iterative decoding? Does a multirate system without iterative decoding perform as well as the system presented here? We have simulated CM_{22} and CM_{23} on the Rayleigh fading channel with $f_dT = 0.1$ and only 1 decoding iteration, the $\frac{E_s}{N_0}$ required to achieve a BER = 10^{-3} are 3.7 and 2.9 dB, respectively. Then we can compute $(\frac{E_s}{N_0})_{average}$ for the non-iterative system. In city environment, $(\frac{E_s}{N_0})_{average}$ = 15.2 dB, in highway environment, $(\frac{E_s}{N_0})_{average}$ = 2.0 dB ($f_dT = 0.1$ in both cases). This means that the improvement is 1.8 dB and 0.6 dB. This is not as spectacular as the multirate improvement, but significant since this gain due to iterative codes is "free", in other words, it does not make the decoding longer or more complex. Again the largest improvement is achieved on the worst channel.

6.4 Conclusions

In this chapter, we have presented a new method to evaluate the performance of a rate adaptive coding system on the land mobile satellite channel for speech transmission application. Three rate adaptive systems, using coding technique presented in the

previous chapters, have been detailed and evaluated on the land mobile satellite channel: multilevel codes' results are very encouraging. It has been shown that the gain achieved by multirate techniques over single rate techniques can be as high as 5.0 dB with highway channel conditions with practically no quality loss, and 8.8 dB with urban environment.

Chapter 7

Conclusion

7.1 Thesis summary

In this thesis, rate adaptive forward error correction techniques for mobile satellite communications have been studied. We were especially interested in codes that can be iteratively decoded. In Chapter 2, a model was chosen for the mobile satellite channel. The channel was divided into three different states with state probabilities that depend on the environment. It turned out that two cases have to be separated: when the mobile is in the city or on the highway.

Chapter 3 gave a review of multilevel coded modulation and design rules were presented using the asymptotic coding, the minimum Hamming distance and information theory arguments. Performance results have been obtained through computer simulations over AWGN with iterative decoding.

In Chapter 4, a new method to construct multilevel coded modulation schemes using parallel concatenation and its iterative decoding algorithm were proposed. The asymptotic coding gain of the new coded modulation was computed and simulations have been performed for the AWGN channel case.

The two-dimensional (24,12) Golay product code was studied in Chapter 5. An iterative decoding algorithm and simulation results over AWGN and Rayleigh fading channels were given.

In Chapter 6, three rate adaptive forward error correction systems were presented. A performance criterion for such a system has been found for speech transmission applications. A systematic evaluation of the three systems has been conducted. We have found that the gain that a rate adaptive system can achieve over a non adaptive rate system can be up to 5.0 dB in highway environments without significant losses in quality and 8.8 dB in urban environments.

7.2 Suggestions for further research

As we have seen, multilevel coded modulation techniques are very powerful when the component codes are carefully chosen. The coded modulation schemes we considered in this work use convolutional codes: they achieve their best performance with large interleavers. One may ask: what is the effect of the interleaver size on product coded modulation? Can one construct powerful codes with shorter blocks?

From the practical point of view, the simulation of the adaptive rate systems can include power control algorithms, rate selection algorithms and use the more accurate two-state channel. Thanks to this two-state channel model, it could be interesting to compute the bit error rate distribution and to deduct the availability of service: the probability that the bit error rate is under a given threshold (here 10^{-3}).

As indicated in Chapter 2, the channel model we considered is applicable to the L-band communication systems. Performance evaluation can be done with a K/Ka band channel model, as soon as accurate K/Ka band channel model becomes available.

Appendix A

The Bahl algorithm

The Bahl algorithm can find a posteriori probabilities on information and/or channel bits for any code with a trellis.

The state at instant t is denoted S_t. Let $(i_1^\tau) = i_1 i_2 \ldots i_\tau$ represent the information bits. The $(x_1^\tau(j))$ are the channel bits for $j = 1 \ldots n_b$, where n_b is the number of channel bits per information bit. $(y_1^\tau(j))$ are the complex valued symbols received by the demodulator for which we know the conditional probability density functions $p(y_t(j)|x_t(j))$.

The decoder needs the following probabilities $Pr(S_t = m|S_{t-1} = m')$, that depend on the trellis structure and the a priori probabilities on information bits. The probability $Pr(S_t = m|S_{t-1} = m') = 0$ if there is no transition from state m' to state m in the trellis. When using convolutional code, for which a state transition corresponds to an information bit (when bits are fed to the coder one at a time), $Pr(S_t = m|S_{t-1} = m') = Pr(i(m' \rightarrow m))$, which is the a priori probability of the information bit associated with the transition $m' \rightarrow m$. We introduce the a priori probabilities on information bits through these transition probabilities.

We define the α's, β's, γ's, λ's and σ's as follows:

$$\alpha_t(m) = p(S_t = m, y_1^t)$$

$$\beta_t(m) = p(y_{t+1}^\tau | S_t = m)$$

$$\gamma_t(m', m) = p(S_t = m, y_t | S_{t-1} = m')$$

$$\lambda_t(m) = p(S_t = m, y_1^\tau)$$

$$\sigma_t(m', m) = p(S_{t-1} = m', S_t = m', y_1^\tau)$$

The following equalities hold:

$$\lambda_t(m) = \alpha_t(m)\beta_t(m) \tag{A.1}$$

$$\sigma_t(m', m) = \alpha_{t-1}(m)\gamma_t(m', m)\beta_t(m) \tag{A.2}$$

$$\alpha_t(m) = \sum_{m'} \alpha_{t-1}(m)\gamma_t(m', m) \tag{A.3}$$

$$\beta_t(m) = \sum_{m'} \beta_{t+1}(m)\gamma_{t+1}(m, m') \tag{A.4}$$

$\gamma_t(m', m)$ is the branch probability, it can be computed directly with:

$$\gamma_t(m', m) = \prod_{j=1}^{n_b} p(y_t(j)|x_t(j)(m', m)) \tag{A.5}$$

The trellis starts and ends in state 0 at instant $t = 0$ and $t = \tau$. As the symbols are received, the values of the γ's can be computed. A forward recursion algorithm computes the α parameters using equation A.3 with initial values: $\alpha_0(0) = 1$ and $\alpha_0(m) = 0$ for $m \neq 0$. Then, a backward recursion algorithm computes the β parameter using equation A.4 with these initial values: $\beta_\tau(0) = 1$ and $\beta_\tau(m) = 0$ for $m \neq 0$. The λ's and σ's can now be easily computed with equations A.1 and A.2.

If we want to obtain the information bit probabilities, we perform the MAP decoding procedure: information bits depend on the states for convolutional codes. Let's define the set $A_t(b)$ for $b = 0, 1$

$$A_t(b) = \{S_t \text{ such that the last bit fed to the encoder is b}\}$$

We then get:

$$p(i_t = b, y_1^\tau) = \sum_{m \in A_t(b)} \lambda_t(m)$$

For systematic linear block codes, information bits are also channel bits and their probabilities are computed the same way as channel bit probabilities. To determine

channel bits probabilities, we apply the MAP filtering computation: channel bits depend on state transitions. Let's define the set $B_t(b)$ for $b = 0, 1$

$$B_t^{(j)}(b) = \{(S_t, S_{t+1}) \text{ such that the } j^{th} \text{ output bit is b}\}$$

Then:

$$p(x_t(j) = b, y_1^\tau) = \sum_{(m,m') \in B_t^{(j)}(b)} \sigma_t(m, m')$$

We then normalize by dividing by $\lambda_\tau(0) = p(y_1^\tau)$ to find the a posteriori probabilities $Pr(i_t = b|y_1^\tau)$ and $Pr(x_t(j)|y_1^\tau)$ for information and channel bits. The complexity of this algorithm is essentially twice the complexity of the Viterbi algorithm since a single forward and a single backward recursions have to be done.

Appendix B

Trellis for a linear block code

The trellis of a linear block code can be constructed from the parity check matrix. Let C be a (n, k) linear block code, then the parity check matrix H is a $(n-k) \times n$ matrix. H has the following property: $c \in GF[2]^n$ is a codeword in C if and only if $H^T c = 0$. Let h_i be the i^{th} column of matrix H. The condition becomes $\sum_{i=1}^{n} c_i h_i = 0$ if and only if $c = (c_1, c_2, \ldots, c_n)$ is a codeword in C.

Then, we can define the states S_t for $t = 0, 1 \ldots, n$ as follows $S_0 = 0$ and

$$S_t = S_{t-1} + c_t h_t = \sum_{i=1}^{t} c_i h_i \qquad \text{for } t = 1, 2, \ldots, n$$

Then obviously the trellis ends up in state $S_n = 0$. The number of states in the trellis is 2^{n-k}. This method can be used when $n - k \leq k$.

A simple example consists of a parity code of length 7. Matrix H equals $\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$ for the special case. Figure B.1 gives the trellis of the (7,6) parity check code.

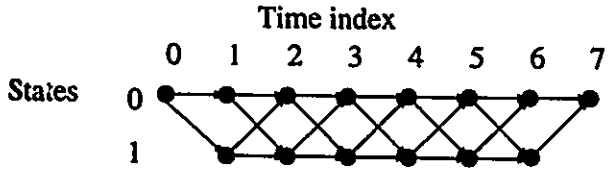

Figure B.1: *Trellis of the (7,6) parity check code*

Bibliography

[1] F. Ananasso and F. Delli Priscoli. The role of satellite in personal communication services. *IEEE J. Select. Areas Commun.*, 13:180–196, Feb. 1995.

[2] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv. Optimal decoding of linear codes for minimizing symbol error rate. *IEEE Trans. Inform. Theory*, 20:284–287, Mar. 1974.

[3] G. Battail. Pseudo-random recursive convolutional coding for near-capacity performances. In *Int. Symp. on Commun. Theory and Applications*, Jul. 1993.

[4] C. Berrou, A. Glavieux, and P. Thitimajshima. Near Shannon limit error-correcting coding: turbo codes. In *ICC'93*, pages 1064–1070, 1993. Geneva.

[5] R. Blahut. *Principles and practice of information theory.* Addison Wesley, 1987.

[6] G. Cohen, J.-L. Dornstetter, and P. Godlewski. *Codes correcteurs d'erreurs: une introduction au codage algébrique.* Masson, Paris, CNET-ENST, 1992.

[7] D. Cygan and E. Lutz. A concatenated two-stage adaptive error control scheme for data transmission in time-varying channel. *IEEE Trans. Commun.*, 43:795–803, Apr. 1995.

[8] E. O. Gilbert. Estimates of error rates for codes on burst noise channel. *Bell Syst. Tech. J.*, 42:1977–1997, Sep. 1963.

[9] J. Hagenauer. Rate compatible punctured convolutional codes (RCPC codes) and their applications. *IEEE Trans. Commun.*, 36:389–400, Apr. 1988.

[10] J. Hagenauer, P. Hoeher, and J. Huber. A Viterbi algorithm with soft decision outputs and its applications. In *GLOBECOM'89*, Nov. 1989.

[11] H. Imai and S. Hirakawa. A new multilevel coding method using error correcting codes. *IEEE Trans. Inform. Theory*, 23:371-377, 1977.

[12] Y. Kofman, E. Zehavi, and S. Shamai. Performance analysis of a multilevel coded modulation system. *IEEE Trans. Inform. Theory*, 42:299-312, Feb. 1994.

[13] F. Kschischang and R. Chi-Kong Lee. Applications of nonminimal block code trellises. In *Canadian Workshop on Information Theory*, 1995.

[14] F. Kschischang and V. Sorokine. On the trellis structure of block codes. In *Canadian Workshop on Information Theory*, 1993.

[15] S. Lin and D. Costello. *Error control coding: fundamentals and applications.* Prentice-Hall, Inc., Englewood Cliffs, N.J., 1983.

[16] J. Lodge and R. Young. Separable concatcnated codes with iterative map decoding for Rician fading channel. In *International Mobile Satellite Conference*, pages 467–472, 1993.

[17] J. Lodge, R. Young, P. Hoeher, and J. Hagenauer. Separable map "filters" for the decoding of product and concatenated codes. In *ICC'93*, pages 1740–1745, 1993. Geneva.

[18] C. Loo. A statistical model for a land mobile satellite link. *IEEE Trans. Veh. Technol.*, 34:122–127, Aug. 1985.

[19] E. Lutz, D. Cygan, M. Dippold, F. Dolainsky, and W. Papke. The land mobile satellite communication channel-recording, statistics and channel model. *IEEE Trans. Veh. Technol.*, 40:375–385, May. 1991.

[20] Z. Markovic. Satellite in non-geostationary orbits. *Satellite communications, mobile and fixed service*, pages 345–391, 1993.

[21] A. Monk and L. Milstein. Open-loop power control error in a land mobile satellite system. *IEEE J. Select. Areas Commun.*, 13:205–212, Feb. 1995.

[22] R. Morelos-Zaragoza, T. Kasami, and S. Lin. Multilevel block coded SPSK modulations using unequal error protection codes for the Rayleigh fading channel. In *PIMRC'95*, pages 486–490, 1995.

[23] A. Pedersen. MSAT wide-area fleet management: end-user requirements andapplications. In *International Mobile Satellite Conference*, pages 158–163, 1995.

[24] V. Pless. *Introduction to the theory of error-correcting codes.* Wesley Intersciences, 1989.

[25] G. Pottie and D. Taylor. Multilevel codes based on partitioning. *IEEE Trans. Inform. Theory*, 35:89–98, Jan. 1989.

[26] S. Rajpal and S. Lin. Product coded modulation. In *GLOBECOM'93, Communication theory mini-conference*, pages 7–11, 1993.

[27] S. Sayegh. A class of optimum block codes based on partitioning. *IEEE Trans. Commun.*, pages 1043–1045, Oct. 1986.

[28] W. H. W. Tuttlebee. *Cordless communications in Europe.* Springer-Verlag, London, 1990.

[29] G. Ungerboeck. Channel coding with multilevel/phase signals. *IEEE Trans. Inform. Theory*, 28:55–67, Jan. 1982.

[30] A. Viterbi, J. Wolf, E. Zehavi, and R. Padovani. A pragmatic approach to trelli-coded modulation. *IEEE Communications Magazine*, 27:11–19, Jul. 1989.

[31] T. Woerz and J. Hagenauer. Iterative decoding for multilevel codes using reliability information. In *GLOBECOM'92*, pages 1779–1784, 1992.

[32] E. Zehavi. 8-PSK trellis codes for a Rayleigh channel. *IEEE Trans. Commun.*, 40:873–884, May 1992.