N° d'ordre

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE CIENTIFIQUE

Université Djillali Liabes de Sidi- Bel-Abbes
Faculté de génie électrique
Département d'électronique
Laboratoire de télécommunications et de traitement numérique du signal

THESE DE DOCTORAT

Pour l'obtention du Diplôme de Doctorat en Sciences
Spécialité: Electronique
Option : Signal et Télécommunications
Présentée par

M. BENMAHMOUD Slimane

On Bit-Interleaved Coded Modulation with Iterative Decoding

Soutenu le : **19/04/2018.**

Devant le jury composé de :

Professeur	**BOUZIANI Merahi**	**UDL-SBA**	**Président**
Professeur	**DJEBBARI Ali**	**UDL-SBA**	**Rapporteur**
Docteur	**BENAISSA Mohammed**	**C.U d'Ain Témouchent**	**Examinateur**
Docteur	**BOUCHAMA Idris**	**Univ-de M'sila**	**Examinateur**

Année Universitaire: 2017-2018

DJILLALI LIABES UNIVERSITY

DOCTORAL THESIS

On Bit-Interleaved Coded Modulation with Iterative Decoding

By
Slimane BENMAHMOUD
A thesis submitted in partial fulfillment of the requirements
for the degree of ***'Doctorat en Sciences'***

in

Electronic Engineering
Department of Electronic Engineering

19/04/2018.

Declaration of Authorship

I, Slimane BENMAHMOUD, declare that this thesis titled, “On Bit-Interleaved Coded Modulation with Iterative Decoding” and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a ""Doctorat en Sciences" degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

"Don't let anyone ever make you feel like you don't deserve what you want."

Heath Ledger

"Your time is limited, so don't waste it living someone else's life."

Steve Jobs

DJILLALI LIABES UNIVERSITY

Abstract

Faculty of Electrical Engineering
Department of Electronic Engineering

'Doctorat en Sciences'

On Bit-Interleaved Coded Modulation with Iterative Decoding

by Slimane BENMAHMOUD

To increase the diversity order of coded modulation, Zehavi proposed Bit Interleaved Coded Modulation (BICM). Compared to Trellis Coded Modulation (TCM), BICM is very flexible in the design of the encoder and the modulator. BICM outperforms Ungerboeck's TCM over fully interleaved Rayleigh fading channel but it suffers degradation over AWGN channel.
BICM with soft-decision feedback significantly outperforms TCM and performs closely to Turbo-TCM over both AWGN and Rayleigh fading channel.
This thesis considers BICM with iterative decoding (BICM-ID) in its soft-decision feedback version. In this work, we've studied the Soft-Input Soft-output (SISO) decoding algorithm and we've investigated the modulation block to improve the adaptability and the performance of this system: first, after a detailed theoretical study on BICM, its performance has been characterized. Then, to get a clear idea on iterative decoding and to fully explore the BCJR algorithm used later in BICM-ID, turbo codes have been studied. After this, the focus was on the joint optimization of signal constellation points and their mappings where a new improved symbol mapper/8-ary constellation was introduced. Finally, we've investigated the MAP algorithm's variants. In particular, we've proposed a new approximation for the Jacobian logarithm and we've demonstrated its suitability for MAP decoding in BICM-ID.

DJILLALI LIABES UNIVERSITY

Abstract

Faculty of Electrical Engineering
Department of Electronic Engineering

'Doctorat en Sciences'

On Bit-Interleaved Coded Modulation with Iterative Decoding

by Slimane BENMAHMOUD

Résumé:

Pour augmenter l'ordre de diversité de la modulation codée, Zehavi a proposé la modulation codée à bits entrelacés (BICM). Par rapport à la modulation codée en treillis (TCM), la modulation BICM est très flexible en ce qui concerne la conception du codeur et du modulateur. La modulation BICM surpasse la modulation TCM d'Ungerboeck à travers un canal à évanouissement de Rayleigh complètement entrelacé mais elle souffre une dégradation à travers un canal AWGN.

La modulation BICM avec un retour de décision souple surpasse, d'une manière considérable, la modulation TCM et elle a une performance proche à celle de la modulation Turbo-TCM à travers les canaux AWGN et à évanouissement de Rayleigh.

Cette thèse traite la modulation BICM avec décodage itératif (BICM-ID) dans sa version de décision souple. Dans ce travail on a étudié l'algorithme de décodage à entrée-souple et sortie-souple (SISO) et on a examiné le bloc de modulation pour améliorer l'adaptabilité et la performance de ce système. En premier lieu, après une étude théorique détaillée sur la modulation BICM, sa performance a été caractérisée. Ensuite, afin d'avoir une idée claire sur le décodage itératif et de bien explorer l'algorithme BCJR, qui va être utilisé plus tard pour la modulation BICM-ID, les turbo codes ont été étudiés.

Ensuite, on s'est concentré sur l'optimisation conjointe des points de la constellation et leurs étiquettes où on a introduit un nouvel mappeur/constellation 8-aire. Enfin, on a examiné les différentes variantes de l'algorithme MAP. En particulier, on a proposé une nouvelle approximation pour le logarithme Jacobien et on a démontré qu'elle est bien convenable au décodage MAP pour une modulation BICM-ID.

DJILLALI LIABES UNIVERSITY

Abstract

Faculty of Electrical Engineering
Department of Electronic Engineering

'Doctorat en Sciences'

On Bit-Interleaved Coded Modulation with Iterative Decoding

by Slimane BENMAHMOUD

ملخص:

بغرض زيادة رتبة التنوع في التضمين المشفر، اقترح Zehavi تضمين BICM .

مقارنة مع تضمين TCM ، فان تضمين BICM جد مرن فيما يتعلق بالمشفر و المضمن. اداء تضمين BICM يتفوق على اداء تضمين TCM خلال قناة الخفوت ل Rayleigh المشذرة كليا و لكنه يعاني تراجعا عبر قناة AWGN .

تضمين BICM المصحوب بقرار رجعي مرن يتفوق بجدارة على تضمين TCM و يقارب في ادائه تضمين Turbo-TCM عبر كلتا القناتين AWGN و قناة الخفوت ل Rayleigh.

هذه الدراسة تهتم بتضمين BICM المصحوب بفك تشفير تكراري (ID) في نسخته ذات القرار الرجعي المرن (Soft-decision feedback).

في هذا العمل ، درسنا خوارزمية فك التشفير SISO و الجزء الخاص بالتضمين بغرض تحسين تكيف و اداء النظام. بداية، بعد دراسة نظرية مفصلة حول تضمين BICM، تم توصيف ادائه. بعدها و بغرض زيادة فهم فك التشفير التكراري و الاستكشاف الكلي لخوارزمية BCJR، التي ستستخدم فيما بعد في نظام BICM-ID، تم دراسة تشفيرات تربو (Turbo Codes), بعدها، تم التركيز على التحسين المشترك لنقاط كوكبة الاشارة و نعييناتها اين تم اقتراح تحسين مشترك لمعين رموز/كوكبة ثمانية, في النهاية، استثمرنا في صيغ خوارزمية MAP، و بالخصوص اقترحنا تقريبا للوغاريتم الجاكوبي و برهنا انه متوافق مع فك تشفير MAP المستخدم في تضمين BICM-ID

Acknowledgements

My reserved praises and thankfulness are for Allah, the most compassionate and the most merciful. He blesses me with his ever-enduring mercies.

I would like to thank my thesis' advisor, Prof. Djebbari Ali. I also thank all the committee members (Prof. BOUZIANI Merahi, Dr. BENAISSA Mohammed and Dr. BOUCHAMA Idris) for their time. My thanks to all my friends. Last but not least, I'm thanking my parents, my sister, and my brothers, for their endless love and support. . . . . .

Contents

List of Figures

List of Tables

List of Abbreviations

BICM	Bit Interleaved Coded Modulation
TCM	Trellis Coded Modulation
AWGN	Additive White Gaussian Noise
BICM-ID	BICM with Iterative Decoding
SISO	Soft-Input Soft-Output
MAP	Maximum A Posteriori
PSD	Power Spectral Density
PDF	Probability Density Function
SNR	Signal-to-Noise Ratio
BER	Bit Error Rate
MIMO	Multiple-Input Multiple-Output
FIR	Finite Impulse Response
FSM	Finite State Machine
ML	Maximum Likelihood
SOVA	Soft-Output Viterbi Algorithm
BCJR	Bahl, Cocke, Jelinek, and Raviv algorithm
PSK	Phase Shift Keying
PEP	Pairwise Error Probability
MSED	Minimum Squared Euclidian Distance
NSC	Non-Systematic Non-Recursive
LLR	Log-Likelihood Ratio
SISO	Soft-Input Soft-Output
SP	Set Partitioning
SSP	Semi SP
QPSK	Quaternary PSK
MSEW	Maximum Squared Euclidian Weight
RSC	Recursive Systematic
BPSK	Binary PSK
MI	Mutual Information
CE	Cross-Entropy
MOR	Measurement of Reliability
PE	Probability of Error
HAD	Hard Decision Aided
SCR	Sign Change Ratio
HMMSED	Harmonic Mean of the Minimum Squared Euclidian Distance

To good people all over the world…

Chapter 1

Introduction and Background

In this chapter, We'll talk about some basic concepts in digital communication. First, we'll present a digital communication system and we'll describe the role of its basic elements. Then, we'll present two statistical channel models namely AWGN and fading that will be used in this work. After that, we'll talk about diversity techniques used to mitigate fading over the second type of channels (i.e. fading channels). Shannon's channel coding theorem will also be highlighted. After that, error-control coding techniques will be briefly discussed, and we'll particularly discuss convolutional codes which represent a basic element in our study. Trellis coded modulation will be then briefly introduced. The outline and contribution of this thesis will be presented at the end of this chapter.

1.1 Digital communication systems

Basically, a digital communication system contains three major elements: a transmitter, a communication channel, and a receiver (Fig. 1.1). In its most simple form a transmitter contains only a

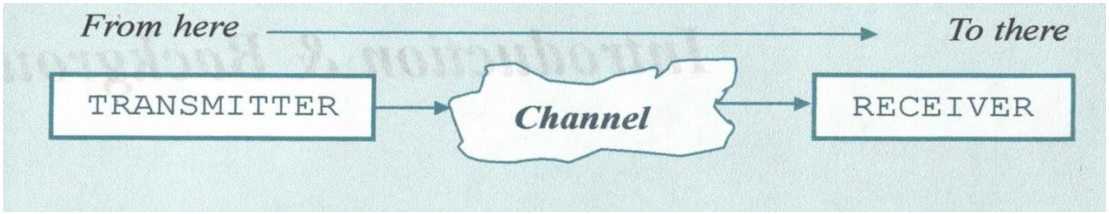


FIGURE 1.1: The basic elements of a digital communication system [1].

binary source and a modulator. This latter performs the operation of binary bits' mapping to signals (modulation) that can be transmitted over the channel. The channel has negative effects on the transmitted signal such as attenuating its amplitude, adding noise, and changing its phase [1]. To overcome the effects of an impaired channel on the transmitted signal, some error correction/ control capability is needed. For this purpose, an entity called a channel encoder is needed. The use of this later results in a coded communication system. The channel encoder introduces a controlled redundancy to the binary data bits, that will be mapped into transmitted signals, to make the data transmission more reliable. Error control/correction coding or channel coding is used to achieve high transmission reliability at the cost of more bandwidth and an increase in the receiver complexity. At the receiver, the inverse operations, namely demodulation and channel decoding, are performed [2].

1.2 The wireless communication channel

In any communication system, the channel is used to carry the signal to be transmitted from the transmitter to the receiver. An electromagnetic wave representing the transmitted signal may travel from the transmitter to the receiver on the direct line-of-sight or my reach the receiver with different reflected, diffracted and/or scattered versions [1].
For line-of-sight space and satellite communications, the Additive White Gaussian Noise (AWGN) channel model is considered as an appropriate model [2]. It provides a minimum complex starting model [2].
In mobile radio channels, the line-of-sight is rarely available and we have a multi-path propagation of the transmitted signal which leads to the phenomena of fading [2]. In this case, the AWGN channel model must be extended to introduce the fading assumption.

1.2.1 AWGN channel model

Let x_k be the transmitted discrete-time signal and r_k be the received discrete-time signal at time k. In an AWGN discrete-time channel model, the received signal (Fig. 1.2) is given by

$$r_k = x_k + n_k \tag{1.1}$$

where n_k is the value, at time k , of a discrete-time Gaussian random variable (r.v) N having a mean $\mu_N = 0$ and a variance σ_N^2

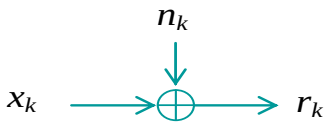


FIGURE 1.2: Discrete-time AWGN channel model [2].

1.2.2 Fading channel model

For a discrete-time fading channel model, the received signal r_k (Fig. 1.3) at time k is given by

$$r_k = a_k.x_k + n_k \tag{1.2}$$

where a_k is the scalar fading gain. We consider, in this work, a flat fading where a_k is considered constant over the symbol time. The fading gain a_k can be Rayleigh, Racian, or Nakagami distributed [4].
Rayleigh fading:
Rayleigh fading channel model is considered when there is a large number of independent scatters and no line-of-sight signal. Urban areas in mobile communication, for instance, mostly possess this kind of channel. The fading amplitude A is modeled as a Rayleigh random variable whose pdf is

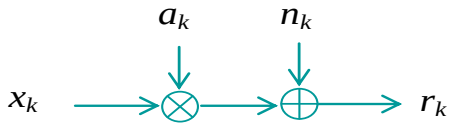


FIGURE 1.3: Discrete-time fading channel model [2].

given by [4]

$$f_A(a) = \frac{2a}{\sigma_A^2} e^{\frac{-a^2}{\sigma_A^2}} \tag{1.3}$$

where σ_A^2 is its variance.

Rician fading:

Rician fading channel model is used when there is a strong line-of-sight signal and additional diffuse signals from multi-path reflections [4]. For instance, rural areas in mobile communication, usually have this kind of channel. The fading amplitude A, in this case, is modeled as a Rician random variable whose pdf is given by [4]

$$f_A(a) = \frac{a}{\sigma_A^2} e^{\frac{-(a^2+v^2)}{2\sigma_A^2}} . I_0(\frac{av}{\sigma_A^2}) \tag{1.4}$$

where v^2 represents the power in the line-of-sight path and $2\sigma_A^2$ denotes the diffuse power. $I_0(.)$ is the zeroth-order modified Bessel function of the first kind.

If we put

$$k = \frac{v^2}{2\sigma_A^2} \tag{1.5}$$

and,

$$E[A^2] = v^2 + 2\sigma_A^2 = 1 \tag{1.6}$$

Equation (1.4) can be rewritten as

$$f_A(a) = 2a(1+k).e^{-(k+a^2(1+k))} . I_0(2a\sqrt{k(k+1)}) \tag{1.7}$$

From equation (1.7), we can observe that as the line-of-sight power v^2 is reduced $(k \to 0)$, the term $I_0(2a\sqrt{k(k+1)}) \to 1$, and the Rician pdf becomes a Rayleigh pdf with $\sigma_A^2 = 1$.

Nakagami fading:

Nakagami-m fading channel model is a generalization of the Rayleigh fading channel model with two parameters. The fading amplitude A is then modeled as a Nakagami-m random variable whose pdf is given by [4]

$$f_A(a) = \frac{1}{\Gamma(m)} . \left(\frac{m}{\sigma_A^2}\right)^m . a^{2m-1} . e^{\frac{-ma^2}{\sigma_A^2}} , \; m \geq 0.5 \tag{1.8}$$

where $\Gamma(.)$ is the Gamma function and $m = \sigma_A^4 / E[(A^2 - \sigma_A^4)]$ is the ratio of moments. for $m = 1$, equation (1.8) gives the pdf of a Rayleigh distributed r.v.

With a two parameter pdf, Nakagami distribution can describe different fading environments. Because of this feature, it is widely used in performance analysis. It was suggested as the best fit for

the distribution of the signals received in urban radio multipath channels.

1.3 Diversity techniques over the wireless channel

Transmission over fading channels results in a sever degradation of the system's BER performance. An efficient solution to mitigate the effect of fading is to provide copies of the same transmitted signal that are affected by independent fading gains, at the receiver side and appropriately process them to make the detection more reliable. This technique is called diversity. The following diversity techniques are of particular interest.

1.3.1 Time Diversity

In time diversity, the information signal is transmitted in different time intervals in such a way that it experiences independent fades over time. Repetition code is an example of time diversity where the information signal is repeated exactly the same over a number of time intervals. However, the use of repetition codes decreases the system's bandwidth efficiency. Channel codes can be considered as a time diversity technique, where redundancy is used by the receiver to overcome the effects of fading. In this thesis two types of channel codes are considered: convolutional codes (CC) and their parallel concatenation (Turbo codes)[8].

1.3.2 Frequency Diversity

In frequency diversity, the information signal is transmitted using different frequencies to provide independently faded versions of the information signal at the receiver side. Frequency diversity can be used over frequency selective channels [8].

1.3.3 Spatial Diversity

In spatial diversity, also known as antenna diversity, multiple transmitting or receiving antennas (or both) are used. When the space between adjacent antennas is sufficient, the signals transmitted between different antenna pairs will be affected by independent fading gains. Three types of antenna diversity exist. Transmit diversity when multiple antennas are used at the transmitter, receive diversity when multiple antennas are used at the receiver, and transmit-receive diversity when multiple antennas are used at both the transmitter and the receiver which results in the so-called Multiple-Input Multiple-Output (MIMO) system [8].

1.4 Shannon's channel coding theorem

Consider a discrete memoryless source with an alphabet l having the entropy $H(l)$ and produces symbols over T_s seconds. Consider also a discrete memoryless channel that has a capacity C that it is used once every T_c seconds.

The channel-coding theorem also known as the noisy coding theorem states that if [9]

$$\frac{H(l)}{T_s} \leq \frac{C}{T_c} \tag{1.9}$$

there exists a coding scheme for which the source output can be reliably transmitted over the channel. The term reliably doesn't imply zero probability of error, but it means that the error probability can be made arbitrarily small.
If the condition in equation (1.9) isn't satisfied (i.e. if $H(l)/T_s > C/T_c$), we can't transmit and reconstruct the source output with an arbitrarily small error probability. The channel coding theorem specifies the channel capacity as an upper limit on the rate at which we can make reliable transmission. The original proof of this theorem was given by Shannon [9]. In chapters 9 and 10 of [10], Hamming presented a detailed proof of the theorem. The channel coding theorem doesn't show how to construct a code that assures a reliable transmission. It doesn't also have a precise result for the probability of error after decoding the channel output.

1.5 Error control coding

Error control coding is a technique used to detect and correct erroneous symbols resulting from transmission over unreliable channels. A channel encoder adds redundancy, to the data symbols, according to a prescribed rule. It proceeds as follows: divides the sequence of data bits into data words each of k bits. Each data word will be mapped to a code word of n bits $(n > k)$. Out of 2^n possible bit words of length n, there are only 2^k valid code words. The set C of valid code words is called the channel code. The code rate is $R = k/n$. Historically, channel codes have been classified into block and convolutional codes according to the absence or the presence of memory in the encoder [1]. This thesis considers only convolutional codes and their parallel concatenation [9].

1.5.1 Convolutional Codes

Convolutional codes introduced by Elias [11] in 1955 are now widely used in wireless communications. A convolutional encoder can be viewed as a $k-$input, $n-$output Finite Impulse Response (FIR) filter with operations performed over the binary Galois field, and then the encoding process will be a discrete-time convolution of its input sequence with its impulse response. Alternatively, it may be viewed as a Finite State Machine (FSM). So it can be represented by trellis and state diagrams or by graphs [9].
A convolutional code is specified by: the data word length n, the code word length k, and the constraint length K defined as the number of previous data bits involved plus 1. A (n, k, K) convolutional code involves the current data bit and $(K-1)$ previous data bits. $M = K-1$ refers to the code's memory. The number of code's states is 2^M. The code rate is $R = k/n$. A $(n, 1, K)$ binary convolutional code at each time t receives $1-$bit data word d_t and produces an $n-$bit code word $c_t = (c_t^{(1)}, c_t^{(2)}, ..., c_t^{(n)})$ as follows [9]

$$c_t^{(j)} = \sum_{i=0}^{k-1} g_i^{(j)} d_{t-1} \tag{1.10}$$

where $j = 1, 2, ..., n$. $g_i^{(j)} \in 0, 1$ are the coefficients of the polynomial of the convolutional code. The convolutional code (2,1,3) shown in Fig. 1.4 is defined by [9]

$$\begin{cases} c_t^{(1)} = d_t \bigoplus d_{t-2} \\ c_t^{(2)} = d_t \bigoplus d_{t-1} \bigoplus d_{t-2} \end{cases} \tag{1.11}$$

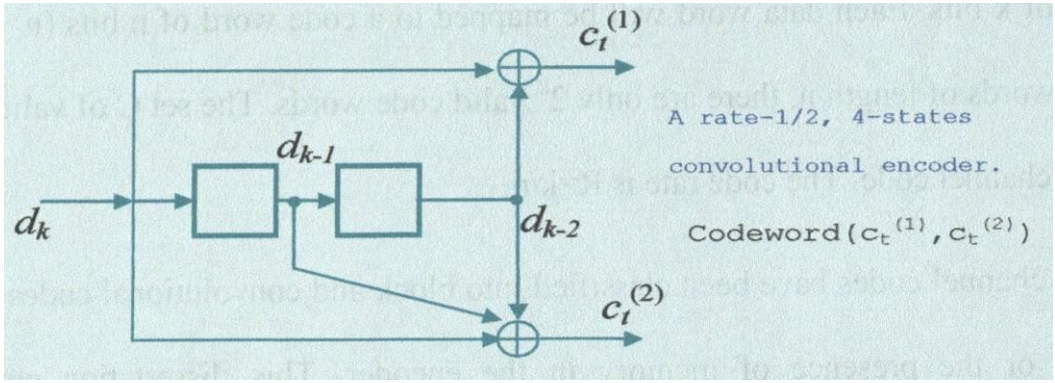


FIGURE 1.4: a (2,1,3) convolutional encoder [12].

1.5.2 Trellis diagram of Convolutional Codes

The trellis diagram is the most frequently used representation of convolutional codes. The trellis diagram for the code shown in Fig. 1.4 is depicted in Fig. 1.5. The upper branch coming out of a

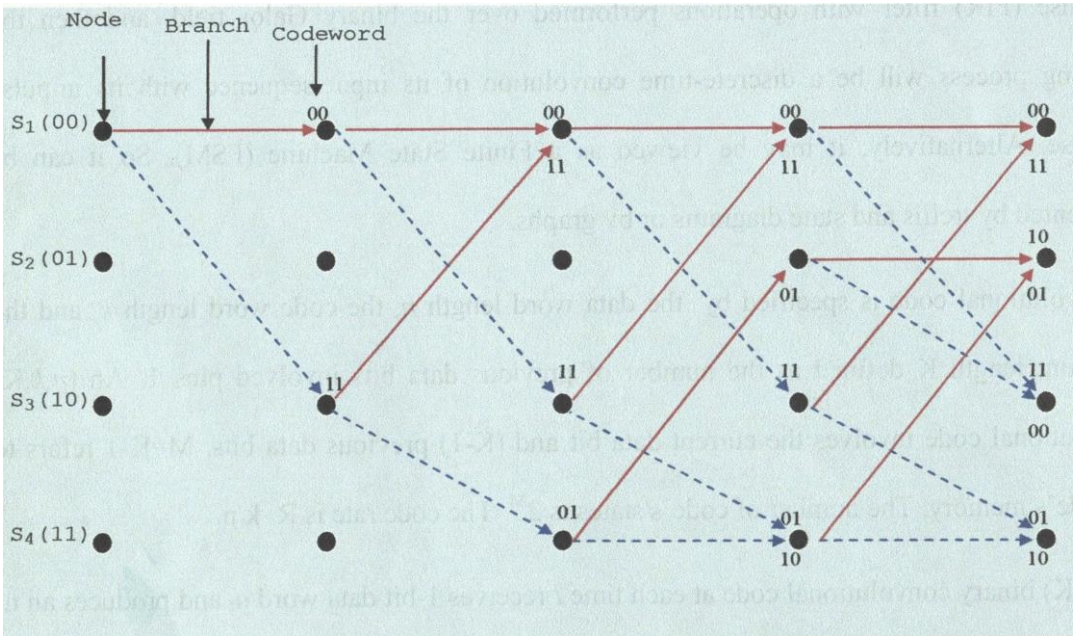


FIGURE 1.5: Trellis diagram for (2,1,3) convolutional encoder [9].

state corresponds to an input data bit equal to 0, and the lower branch corresponds to an input data bit equal to 1. Each node represents a particular state s at a particular time *t* [9].

1.5.3 Distance properties of a convolutional code

The free distance d_{free} defined as the smallest Hamming distance between all possible code-words of the code, determines the error correction capability of this convolutional code. It is given by [9]

$$d_{free} = min_{C_A \neq C_B} d_H(C_A, C_B) = min_{C \neq 0} d_H(C, 0) = min_{C \neq 0} W_H(C) \tag{1.12}$$

where C_A and C_B are two different code words. $d_H(.)$ is the Hamming distance and $W_H(.)$ is the Hamming weight.

1.5.4 Convolutional codes' decoding

Optimal decoding of convolutional codes consists on Maximum Likelihood (ML) decoding. A ML decoder searches, in the trellis, the most likely sequence to the received one. If L is the code bit sequence's length, then all the 2^L possible code sequences must be examined one by one which is impractical for real applications. Low complexity algorithms, to decode convolutional codes, were developed. The Viterbi algorithm (VA) [11] is the most famous one. The basic concept behind the Viterbi algorithm is the sequential computation of the metrics and the tracking of the survivor paths in the trellis. An extended version of the Viterbi algorithm called the Soft-Output Viterbi Algorithm (SOVA) was proposed in [12]. The Bahl, Cocke, Jelinek, and Raviv (BCJR) algorithm that performs Maximum A Posteriori (MAP) decoding using a recursive forward and backward calculation of state probabilities over the trellis was proposed in [13].

1.6 Coded modulation

Channel codes introduce certain constraints on the possible combinations of the bits to be transmitted to help the receiver correct errors introduced by the channel. The previous operation is performed at the expense of adding redundancy which results in reducing the system's bandwidth efficiency. If we use channel encoder and we follow it by a high-order modulator that takes in more bits we can compensate the loss introduced by redundancy. However, the use of high-order modulation has a negative effect on the error-performance of the system because it results in a more crowded constellation. To overcome this unsatisfactory performance, Massey proposed to jointly design the channel encoder and the modulator [14]. By this idea Massey founded a new field called coded modulation (CM). Another very successful solution is the one called Trellis Coded Modulation (TCM) proposed by Ungerboeck [15].

1.6.1 Trellis Coded Modulation

TCM is a coding technique that improves the system's error performance without requiring more bandwidth. TCM has the following basic features [15]:

- The number of constellation points used is larger than what is required for the modulation format of interest with the same data rate; the additional points allow redundancy for error-correction coding without requiring more bandwidth.
- Convolutional coding is used to introduce certain dependency between successive signal points, such that only certain patterns are permitted.
- Soft-decision decoding is performed at the receiver side, in which the permissible sequence of signals is modeled as a trellis structure, hence, the name trellis codes.

In the design of Ungeboeck's TCM, we emphasize on maximizing the Euclidian distance between code words rather than maximizing the Hamming distance of the error-correcting code [15].
To achieve a bandwidth efficiency of m bits/sec/Hz, an $m/(m+1)-$rate convolutional code and a constellation of 2^{m+1} points are used.
Fig. 1.6 shows a TCM with $4-$states, rate$-2/3$ code and $8-$PSK constellation $(m = 2)$. In Unger-

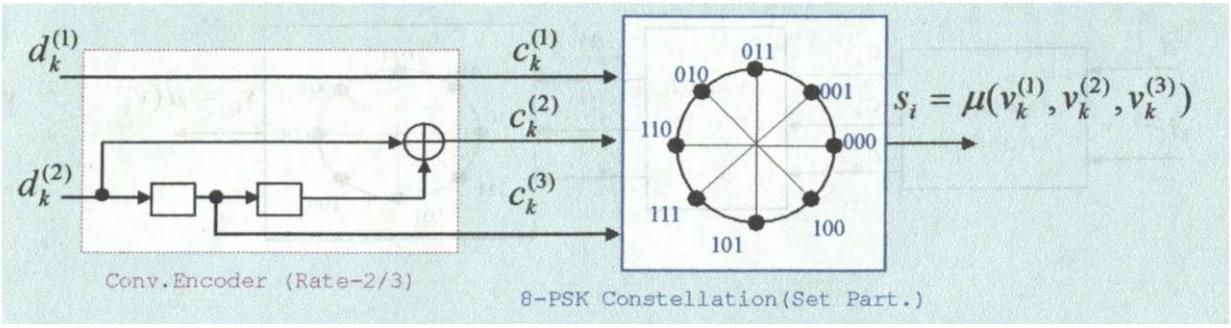


FIGURE 1.6: TCM with 4-states, rate-2/3 code and 8-PSK constellation (m=2) [15].

boeck's TCM, the $M-$ary signal constellation is partitioned hierarchically into constellation subsets of M/2, M/4, M/8, ... point that have progressively larger increasing intra-subsets minimum Euclidian distance. This operation called set partitioning (SP) forms the key idea in the construction of TCM. In Fig. 1.7, the partitioning procedure for an 8-PSK constellation is shown. The resulting subsets share the common property that the minimum Euclidian distance between their individual points increased at every partitioning step [15].

1.6.2 Coded Modulation over Fading Channels

Authors of [16] and [17] used symbol interleaving for TCM to provide time diversity over fading channels. Fig. 1.8 shows a symbol interleaved TCM system. The performance of symbol interleaved TCM was evaluated analytically in [18] for the first time, where it was shown that the Pairwise Error Probability (PEP) $P(x \to \hat{x})$ can be given by

$$P(x \to \hat{x}) = E_a[P(x \to \hat{x}/a)] = E_a\left[Q\sqrt{\frac{\sum_{i=1}^{N} a_i^2 ||x_i - \hat{x}_i||^2}{2N_0}} P(x \to \hat{x}/a)\right] \tag{1.13}$$

where $a = [a_1 a_2 ... a_N]$ is the sequence of channel fades and N is the set of all i for which $x_i \neq \hat{x}_i$. Authors of [19] derived an exact analytical expression of the PEP.

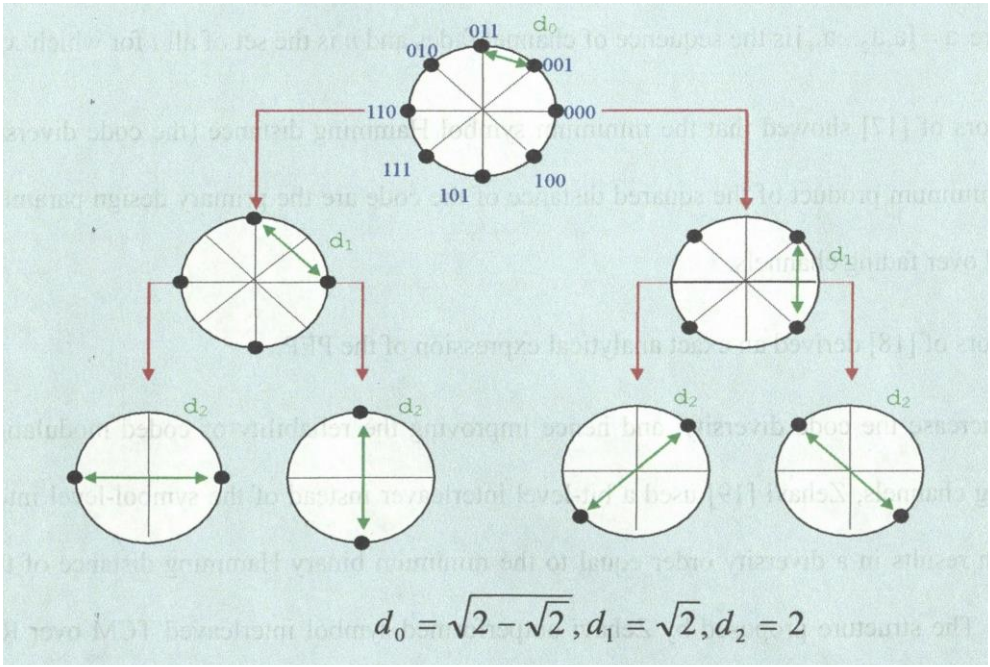


FIGURE 1.7: Partitioning of an 8-PSK constellation ($d_0 < d_1 < d_2$). [16].

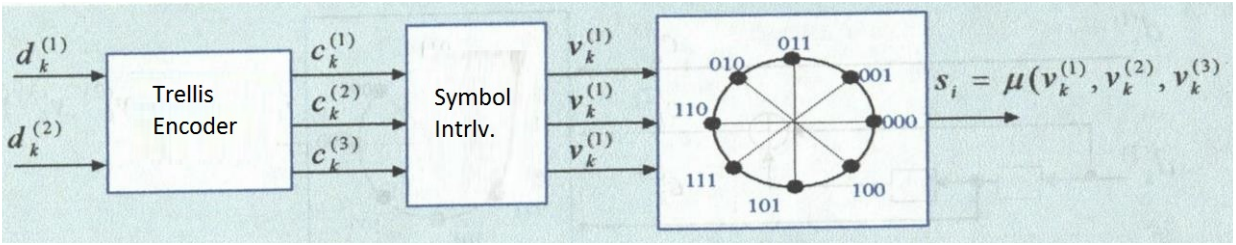


FIGURE 1.8: A block diagram for symbol interleaved TCM [17].

To increase the code diversity, and hence improving the reliability of coded modulation over fading channels, Zehavi [20] used a bit-level interleaver instead of the symbol-level interleaver which results in a diversity order equal to the minimum binary Hamming distance of the used code. The structure proposed by Zehavi outperformed symbol interleaved TCM over Rayleigh fading channels. This scheme called later Bit Interleaved Coded Modulation (BICM) will be the topic of chapter 2.

1.7 Outline and contribution of this thesis

This thesis is organized as follows:
In chapter two, we provide a detailed theoretical study of BICM accompanied by some Matlab® scripts to show how one can simulate a given block in BICM.
We present also simulation results that have been carried out to evaluate the BICM's BER performance and to analyze the effects of various parameters on its performance over both Rayleigh fading and AWGN channels.

Chapter three considers parallel concatenated convolutional codes (Turbo codes) and their iterative decoding technique. It details the BCJR algorithm used in their decoding and finally performs a comprehensive simulation to characterize their BER performance. This material sheds a special light on our understanding of iterative decoding.
Chapter four covers the basic theory underlying iteratively decoded BICM.
The main contribution of this thesis is contained in chapter Five, where firstly the effects of various parameters on BICM-ID's BER performance were investigated. Then a joint optimization of signal constellation points and their mappings were proposed to introduce a new improved symbol mapper/8-ary constellation for BICM-ID, and finally a detailed study on MAP algorithms used in BICM-ID's decoding have been done which led to propose a new approximation for the Jacobian logarithm and to demonstrate that it is suitable for MAP decoding in BICM-ID.
Finally, we conclude this work and we give suggestions for future research.

Chapter 2

Bit-Interleaved Coded Modulation's fundamentals

In this chapter, We'll study, in details, Bit Interleaved Coded Modulation (BICM). We'll provide some Matlab scripts to show how a given block, in BICM, can be simulated. We'll carry out simulations to characterize its Bit Error Rate (BER) performance and analyze the effect of various parameters on its BER performance over both Additive White Gaussian Noise (AWGN) and Rayleigh fading channels.

2.1 Introduction

In 1982, Ungerboeck proposed Trellis Coded Modulation (TCM), a joint design of a convolutional encoder and a multi-level/phase modulator [1]. TCM maximizes the Minimum Squared Euclidian Distance (MSED) to provide a significant improvement over AWGN channels. For fading channels, the diversity order is the key parameter in determining the error performance of a coded system [2]. For TCM using a symbol-based interleaver, the diversity order is the minimum number of distinct symbols between any two code words. By replacing symbol-level interleaving by bit-level interleaving in TCM, Zehavi in [3] succeeded to increase the diversity order to the minimum number of distinct bits between any two code words, which is in fact the minimum binary Hamming distance of the coded system. This scheme, called later in [4] Bit Interleaved Coded Modulation (BICM), achieves better performance than Ungerboeck's TCM over Rayleigh fading channels. For example, a gain of 2 dB at a BER level of 10^{-5} can be attained by using BICM with $8-$states, rate$-2/3$ convolutional encoder, a Gray mapped $8-$PSK constellation and a random bit-level interleaver over $8-$PSK Ungerboeck's symbol-based interleaved TCM [3]. The BICM's diversity order can be increased by using a convolutional code that provides the largest minimum Hamming distance for a given code rate and constraint length.

2.2 BICM's System Model

2.2.1 BICM's transmitter

The BICM's encoder, as introduced in [3], is depicted in Fig. 2.1. It consists of a rate-$2/3$ convolutional encoder serially concatenated with three parallel random bit-level interleavers followed by a Gray mapped 8-PSK constellation. Each two information bits $(d_k{}^{(1)}, d_k{}^{(2)})$ are encoded using

the rate-2/3 convolutional code to produce three coded bits $(c_k{}^{(1)}, c_k{}^{(2)}, c_k{}^{(3)})$ which will be interleaved separately to produce three interleaved bits $(v_k{}^{(1)}, v_k{}^{(2)}, v_k{}^{(3)})$. Each three interleaved bits $(v_k{}^{(1)}, v_k{}^{(2)}, v_k{}^{(3)})$ represent a non-binary complex (channel) symbol, where Gray labeling is used for the sake of optimizing the system's performance. The purpose of the interleavers is to disperse

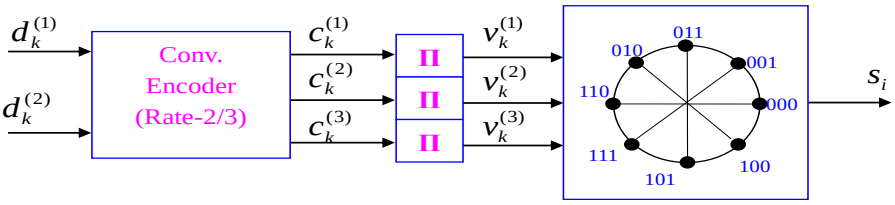


FIGURE 2.1: BICM's encoder schematic employing 3 independent bit interleavers [3].

the bursty errors introduced by the correlated fading and to maximize the diversity order. In [4], authors proposed to use a single bit-interleaver (Fig. 2.2) instead of the three bit-interleavers. The reason for this is to increase the system's flexibility, reduce its complexity and simplify the analysis. Under Matlab environment this random bit-level interleaver can be implemented simply using the

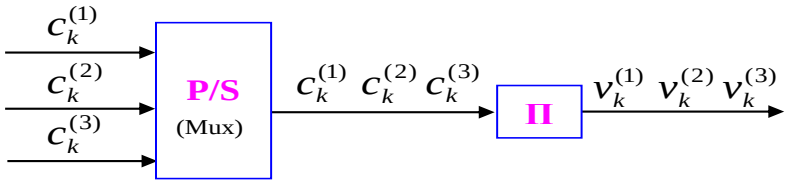


FIGURE 2.2: The single interleaver that replaces the 3 independent interleavers.

following user-defined function:

```
function [output] = BitInterleaver(input,alpha)
% BitInterleaver.m
% Description:  Interleave one block of data according to the interleaving
pattern alpha
% calling syntax:
% [output] = BitInterleaver(input,alpha)
% output = interleaved data
% input = un-interleaved data
% alpha = interleaver pattern
% Interleaving
output=input(alpha);
end
```

Its inverse function (i.e. the bit de-interleaver) can also be implemented as follows

```
function output=BitDeInterleaver(input,alpha)
% BitDeInterleaver.m
output(alpha)=input;
end
```

The BICM's encoder in Fig. 2.1 uses Paaske's Non-Systematic Non-Recursive (NSC) convolutional code chosen (see table 11.1 on page 331 of [5]) as the optimum convolutional code in the sense it gives the highest possible minimum Hamming distance for a rate-2/3 and a constraint length $k = 3$, hence attaining optimum performance over Rayleigh fading channel.
In Fig. 2.3, we show this NSC code. Its free bit-based Hamming distance is $d_{free} = 4$. This code has the generator polynomials $g_1 = [4, 2, 6]$ and $g_2 = [1, 4, 7]$. At each cycle of encoding a two-bits data word $({d_k}^{(1)}, {d_k}^{(2)})$ is encoded to generate a three-bits code word $({c_k}^{(1)}, {c_k}^{(2)}, {c_k}^{(3)})$ and to generate a new state S. This encoder has 8 states. Each state is given by

$$S = \{s^0, s^1, s^2\} \tag{2.1}$$

where s^0, s^1, and s^2 are the contents of the shift registers constructing the encoder in Fig. 2.3.
The trellis diagram for this code is shown in Fig. 2.4. Paaske's NSC code in Fig. 2.3, can be

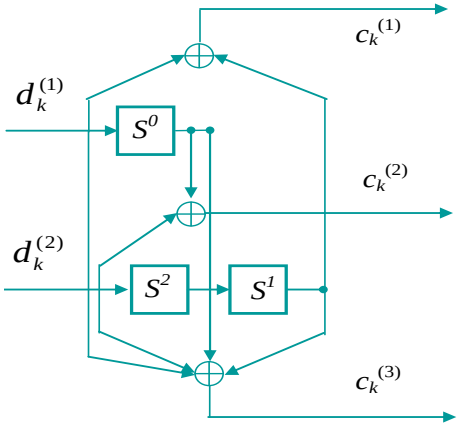


FIGURE 2.3: Paaske's NSC code of 8-states, rate-2/3, constraint length k=4, and generator polynomials $g_1 = [4, 2, 6]$ and $g_2 = [1, 4, 7]$ [5].

implemented using the following user-defined Matlab function

```
   function [encded_bits] = paaske_code(dk1,dk0)
% paaske_code.m
% Description:  Paaske's NSC code shown in figure 3
% calling syntax :
% [output] = paaske_code(input1,input2)
% output = encoded bits
% input1 = first message bit
% input2 = second message bit
% Initialisation of Shift registers and encoded bits
s0=0;s1=0;s2=0;
ck3=zeros(1,length(dk1));
ck2=zeros(1,length(dk1));
ck1=zeros(1,length(dk1));
for bit_index=1:length(dk1);
% Calculation of encoded bits
```

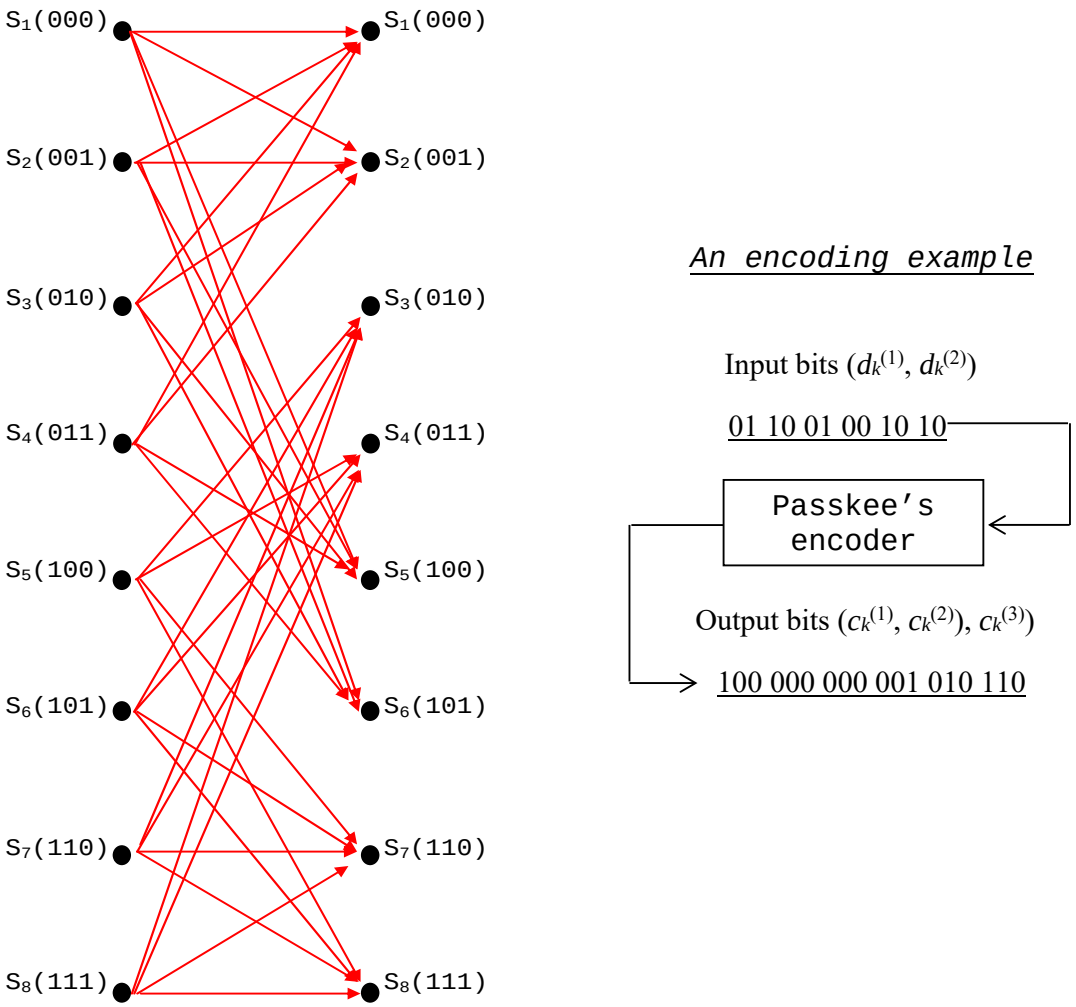


FIGURE 2.4: The trellis diagram for Paaske's code shown in Fig. 2.3.

```
ck3(bit_index)=mod((dk0(bit_index)+s1),2);
ck2(bit_index)=mod(dk1(bit_index)+s0,2);
ck1(bit_index)=mod(s0+dk1(bit_index)+dk0(bit_index)+s1+s2,2);
% Next states of convolutional encoder
if (bit_index  =length(dk1))
s0=dk0(bit_index);
s1=s2;
s2=dk1(bit_index);
end% if
end% for
% reshaping of the encoded bits
bmat=[ck3;ck2;ck1];
enocdedbits=reshape(bmat,1,3*length(ck1));
end% function
```

Each three interleaved bits $({v_k}^{(1)}, {v_k}^{(2)}, {v_k}^{(3)})$ are to be mapped into a complex symbol $s_i \in \Psi$ according to the rule

$$\mu : s_i = \mu(({v_k}^{(1)}, {v_k}^{(2)}, {v_k}^{(3)})), s_i \in \Psi, i = 1, 2, ..., M \tag{2.2}$$

where M is the number of symbols in the constellation set Ψ . For an 8-PSK constellation, $M = 8$, $\Psi = \{s_0, s_1, ..., s_7\}$ and $s_i = exp(-j2\pi i/8)$. The symbol μ in (2.2) represents the operation of signal mapping. A Gray mapper for an 8-PSK constellation, can be implemented using the following user-defined Matlab function.

```
function output=8PSK_GrayMapper(input)
% 8PSK Gray mapper
% calling syntax :
% output = 8PSK_GrayMapper(input)
% input :  binary input vector
% output :  complex output vector
complex_points =exp(j*([1 4 2 3 8 5 7 6]*2*pi/8));
reshaped_input = reshape(input,3,length(input)3);
y = complex_points(bi2de(reshaped_input)'+1);
end% function
```

2.2.2 BICM's receiver

The resulting complex symbols (from the mapping block) are then passed through a discrete-time channel. The received discrete-time signal, for a fully-interleaved flat Rayleigh fading channel model is given by [3]

$$r = g.s_i + n \tag{2.3}$$

where g is the fading amplitude, which can be modeled as a Rayleigh random variable and n is a sample of an AWGN with a zero mean and a variance $\sigma_N^2 = N0/2$, where $N0/2$ is the unilateral Power Spectral Density (PSD) of a white Gaussian noise. For AWGN channel, we take $g = 1$. The received signal r will be used to calculate the conditioned Log-Likelihood Ratio (LLR) for each interleaved bit v_k in the tuple (v_1, v_2, v_3) . This LLR is given by [3]

$$L(v_k/r) = ln\,(p(v_k = 1/r)/p(v_k = 0/r)) \tag{2.4}$$

where

$$\lambda(v_k/r) = ln(p(v_k = b/r)) = ln \sum_{s_i \in \Psi_b^k} p(r/s_i, g) \tag{2.5}$$

is called the bit-metric.
From equations (2.4) and (2.5), it follows that

$$L(v_k/r) = \lambda(v_k = 1) - \lambda(v_k = 0) \tag{2.6}$$

In (2.5), Ψ_b^k , $b \in \{0, 1\}$, $k = 1, 2, 3$, denotes the subset of Ψ that contains all symbols s_i whose labels (v_1, v_2, v_3) have the value b in the k^{th} position. The function $p(r/s_i, g)$ is the probability density

function (PDF) of receiving r conditioned by the transmitted signal s_i and the fading amplitude g. It is given by

$$p(r/s_i, g) = \frac{1}{\sqrt{2\pi\sigma^2}} \times e^{\left(-\frac{\|r - g.s_i\|^2}{2\sigma^2}\right)} \tag{2.7}$$

where

$$d_{s_i} = \|r - g.s_i\| = \sqrt{(r_1 - g.s_{i_1})^2 + (r_2 - g.s_{i_2})^2} \tag{2.8}$$

is the Euclidean distance between r and the faded version of s_i. (r_1, r_2) and (s_{i_1}, s_{i_2}) are the real and imaginary components of r and s_i respectively.
By substituting (2.7) in (2.5), we find that

$$\begin{aligned} \lambda(v_k = b) &= ln \sum_{s_i \in \Psi_b^k} \frac{1}{\sqrt{2\pi\sigma^2}} \times e^{\left(-\frac{\|r - g.s_i\|^2}{2\sigma^2}\right)} \\ &= ln\frac{1}{\sqrt{2\pi\sigma^2}} + ln \sum_{s_i \in \Psi_b^k} e^{\left(-\frac{\|r - g.s_i\|^2}{2\sigma^2}\right)} \end{aligned} \tag{2.9}$$

By using the approximation $ln \sum_i z_i \cong max_i(lnz_i)$, (2.9) can be re-written as follows

$$\lambda(v_k = b) = ln\frac{1}{\sqrt{2\pi\sigma^2}} + max_{s_i \in \Psi_b^k}\left(-\frac{\|r - g.s_i\|^2}{2\sigma^2}\right) \tag{2.10}$$

By substituting (2.10) in (2.6), we find that

$$\begin{aligned} L(v_k/r) &= max_{s_i \in \Psi_1^k}\left(-\frac{\|r - g.s_i\|^2}{2\sigma^2}\right) - max_{s_i \in \Psi_0^k}\left(-\frac{\|r - g.s_i\|^2}{2\sigma^2}\right) \\ &= \left[-min_{s_i \in \Psi_1^k}\left(-\frac{\|r - g.s_i\|^2}{2\sigma^2}\right)\right] - \left[-min_{s_i \in \Psi_0^k}\left(-\frac{\|r - g.s_i\|^2}{2\sigma^2}\right)\right] \end{aligned} \tag{2.11}$$

From (2.11), we can observe that the bit-metric $\lambda(v_k = b)$ and therefore the conditioned LLR $L(v_k/r)$ are based on the MSED between the received signal r and the faded version $g.s_i$ of the transmitted signal s_i.
After being interleaved, the three conditioned LLRs ($L(v_1/r)$, $L(v_2/r)$ and $L(v_3/r)$) will be fed to a standard Soft-Input Hard-Output (SIHO) Viterbi algorithm.
An example of $L(v_k/r)$ computation [7], [9]:
The conditioned LLR of the interleaved bit v_k is given by

$$L(v_k/r) = \left[-min_{s_i \in \Psi_1^k}\left(-\frac{d_{s_i}^2}{2\sigma^2}\right)\right] - \left[-min_{s_i \in \Psi_0^k}\left(-\frac{d_{s_i}^2}{2\sigma^2}\right)\right] \tag{2.12}$$

if we take the case of a Gray mapped conventional 8-PSK constellation shown in Fig. 2.5. For the 1^{st} bit position (i.e. $k = 1$), we have

$$\Psi_1^1 = \{s_5, s_6, s_7, s_8\}; \Psi_1^0 = \{s_1, s_2, s_3, s_4\} \tag{2.13}$$

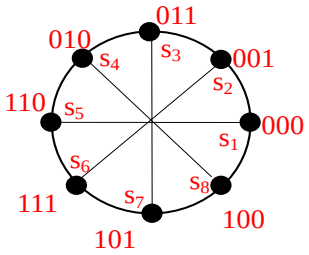


FIGURE 2.5: A Gray mapped conventional 8-PSK constellation.

For the 2^{nd} bit position (i.e. $k = 2$), we have

$$\Psi_1^1 = \{s_3, s_4, s_4, s_6\}; \Psi_1^0 = \{s_1, s_2, s_7, s_8\} \tag{2.14}$$

For the 3^{rd} bit position (i.e. $k = 3$), we have

$$\Psi_1^1 = \{s_2, s_3, s_6, s_7\}; \Psi_1^0 = \{s_1, s_4, s_5, s_8\} \tag{2.15}$$

This operation called Sub-Set Partitioning is illustrated in Fig. 2.6. The conditioned LLR of the

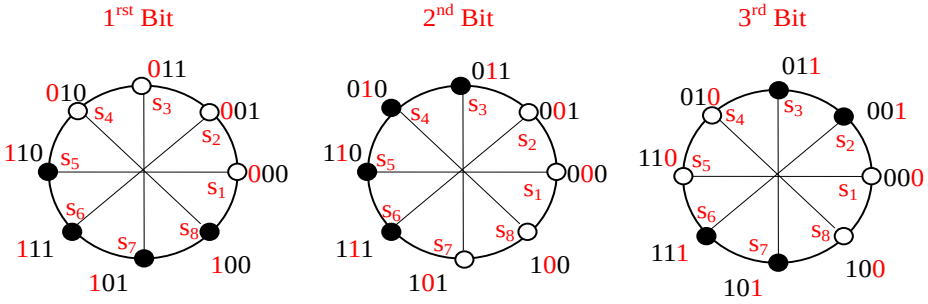


FIGURE 2.6: Sub-Set Partitioning of a Gray mapped 8PSK constellation.

interleaved bit v_1 is given by

$$\begin{aligned} L(v_1/r) &= \left[-min_{s_i \in \{s_5,s_6,s_7,s_8\}} \left(-\frac{d_{s_i}^2}{2\sigma^2}\right)\right] - \left[-min_{s_i \in \{s_1,s_2,s_3,s_4\}} \left(-\frac{d_{s_i}^2}{2\sigma^2}\right)\right] \\ &= \left[-min\left(\frac{d_{s_5}^2}{2\sigma^2}, \frac{d_{s_6}^2}{2\sigma^2}, \frac{d_{s_7}^2}{2\sigma^2}, \frac{d_{s_8}^2}{2\sigma^2}\right)\right] - \left[-min\left(\frac{d_{s_1}^2}{2\sigma^2}, \frac{d_{s_2}^2}{2\sigma^2}, \frac{d_{s_3}^2}{2\sigma^2}, \frac{d_{s_4}^2}{2\sigma^2}\right)\right] \end{aligned} \tag{2.16}$$

For the case shown in Fig. 2.7, $L(v_1/r)$ can be given by

$$L(v_1/r) = \left[-min\left(\frac{d_{s_5}^2}{2\sigma^2}\right)\right] - \left[-min\left(\frac{d_{s_4}^2}{2\sigma^2}\right)\right] \tag{2.17}$$

To simplify the computation of $L(v_k/r)$ the previous operation can be divided into two steps:

- Step 1: bit-metric computation using equation (2.5).
- Step 2: conditioned LLR computation using equation (2.6).

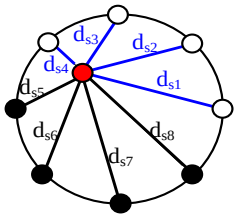


FIGURE 2.7: An example of a received signal r and its Euclidean distances from different constellation symbols (case of 8-PSK constellation).

The following user-defined Matlab® functions can be used for this purpose. In these scripts, we consider a Gray mapped QPSK constellation.

```
function [ bit_metrics ] = bit_metrics( R,S,EsNo,fade_coef )
% bit_metrics transforms received vector R of received symbols (r1,r2,....rn)
into a vector of bit metrics
% calling syntax:
% [output] = bit_metrics( input, S, EsNo, fade_coef)
% output = Vector of bit metrics
% Required inputs:
% R= received complex Vector
% S= complex vector representing the constellation points
% EsNo= the symbol SNR (converted back from dB to linear)
% EsNo = log2(M)*en*rate;
% en = 10^(EbN0db/10);
% where "EbN0db" is the SNR in dB.
% rate is the code rate.
% M is the number of symbols in the constellation.
% fade_coef = 1 by N vector of (complex) fading coefficients (1 by N ones-vector
for an AWGN channel)
%-------------------------------
% begin
for n=1:length(R) bit_metric1(n)=-EsNo*((real(R(n))-fade_coef*real(S(1)))^2+
(imag(R(n))-fade_coef*imag(S(1)))^2);
bit_metric2(n)=-EsNo*((real(R(n))-fade_coef*real(S(2)))^2+
(imag(R(n))-fade_coef*imag(S(2)))^2);
bit_metric3(n)=-EsNo*((real(R(n))-fade_coef*real(S(3)))^2+
(imag(R(n))-fade_coef*imag(S(3)))^2);
bit_metric4(n)=-EsNo*((real(R(n))-fade_coef*real(S(4)))^2+
(imag(R(n))-fade_coef*imag(S(4)))^2);
bit_metrics=[bit_metric1,bit_metric2,bit_metric3,bit_metric4];
end % for
end % function
```

```
function [ bits_LLRs ] = LLRb( bit_metrics)
% LLRb performs bits' LLRs computation
% calling syntax:
% [output] = Compute_LLRb( input )
% % output=vector of bits' LLRs values
% Required inputs:
% input= 1 by M vector of symbol likelihoods ratios
%------------------------------
bit_metric1=bit_metrics(1:4:length(bit_metrics));
bit_metric2=bit_metrics(2:4:length(bit_metrics));
bit_metric3=bit_metrics(3:4:length(bit_metrics));
bit_metric4=bit_metrics(4:4:length(bit_metrics));
for n=1:length(bit_metrics)/4
LLRb1(n)=max(bit_metric2(n),bit_metric3(n))-max(bit_metric1(n),bit_metric4(n));
LLRb2(n)=max(bit_metric3(n),bit_metric4(n))-max(bit_metric1(n),bit_metric2(n));
bits_LLRs=[LLRb2;LLRb1];
end % for
end % function
```

2.3 Performance evaluation of BICM

We present here some simulation results that have been carried out to evaluate the BICM's BER performance and to analyze the effects of various parameters on its performance. The BER vs. SNR curves are provided and commented. The convolutional encoder shown in Fig. 2.8, is used in all the next simulations. A random bit-level interleaver and a conventional $8-$PSK constellation are considered.

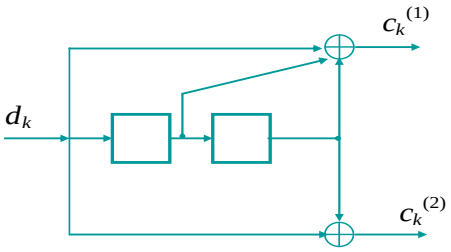


FIGURE 2.8: The NSC convolutional encoder $(7,5)_8$, rate-1/2 and constraint length k=3. [6].

2.3.1 The influence of the block length

In Fig. 2.9 and 2.10, BER vs. SNR curves are shown parameterized by the block length. The simulation parameters, for these two figures, are provided in Table 2.1.

These figures show that, the block length doesn't have a significant effect on the system's BER performance over both Rayleigh and AWGN channels.

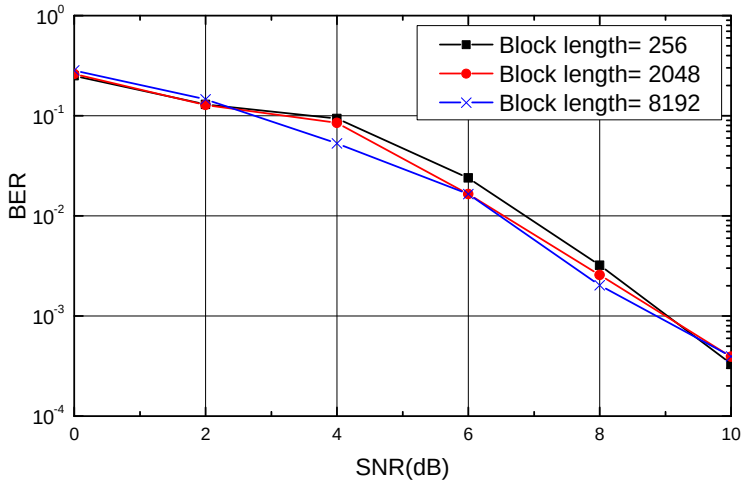


FIGURE 2.9: BER vs. SNR as parameterized by the block length (over Rayleigh fading channel).

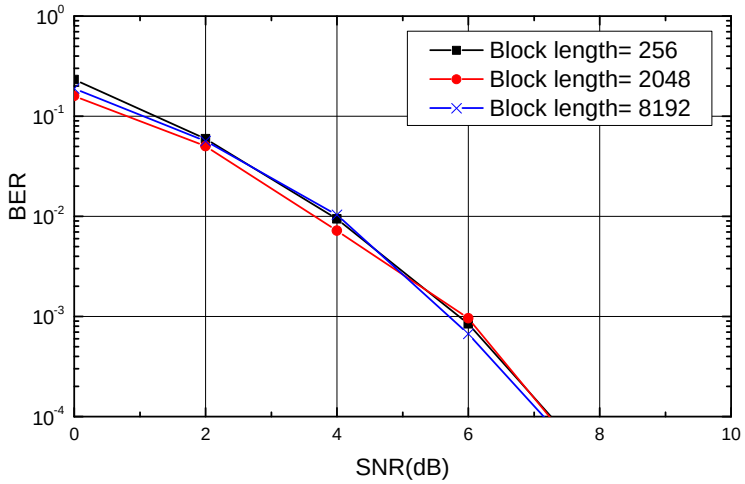


FIGURE 2.10: BER vs. SNR as parameterized by the block length (over AWGN fading channel).

TABLE 2.1: Code Simulation parameters.

Figure #	**Channel**	**Code generator**	**Code rate**	**Mapping**	**Block length**
9	Rayleigh	$(7,5)_8$	$1/2$	Gray	256, 2048, and 8192
10	AWGN	$(7,5)_8$	$1/2$	Gray	256, 2048, and 8192

2.3.2 The influence of signal mapping

In Fig. 2.11 and 2.12, BER vs. SNR curves are shown parameterized by the signal mapping method. Five different signal mappings namely Gray, SP, SSP, MIXED [7] and MSEW [8] are considered. Fig. 2.11 and 2.12 compares the performance of these mappings over Rayleigh and AWGN channels. The simulation parameters, for these two figures, are provided in Table 2.2. From these two figures, we can clearly see that Gray mapping gives the best performance over all other mappings. SSP and MSEW mappings give the worst performance while SP and Mixed mappings' performances are between those of Gray and SSP.

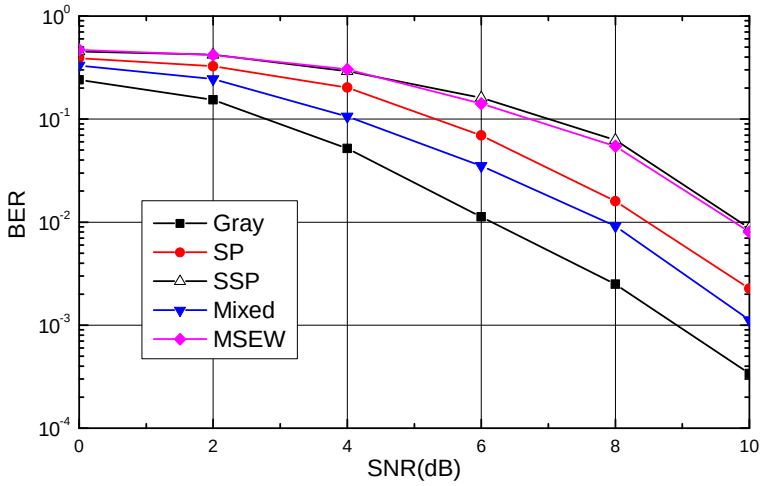


FIGURE 2.11: BER vs. SNR as parameterized by the mapping method (over Rayleigh fading channel).

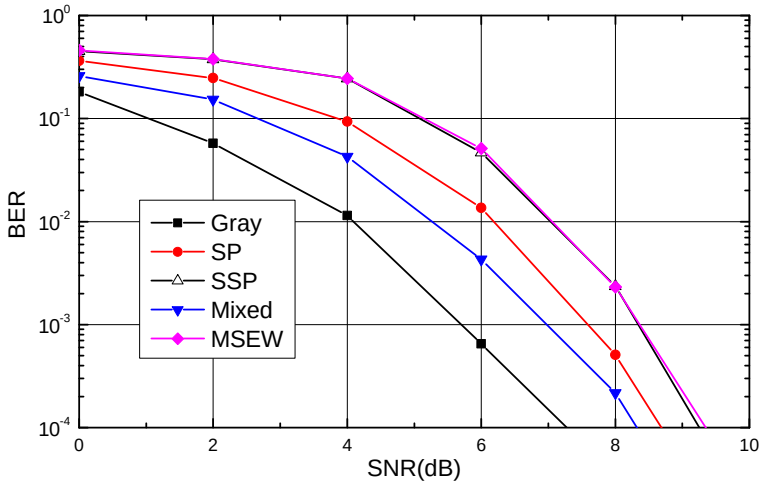


FIGURE 2.12: BER vs. SNR as parameterized by the mapping method (over AWGN fading channel).

TABLE 2.2: Code Simulation parameters.

Figure #	Channel	Code generator	Code rate	Mapping	Block length
11	Rayleigh	$(7,5)_8$	$1/2$	Gray, SP, SSP, MIXED, and MSEW	8192
12	AWGN	$(7,5)_8$	$1/2$	Gray, SP, SSP, MIXED, and MSEW	8192

2.3.3 The influence of the encoder's type

In Fig. 2.13, BER vs. SNR curves are shown parameterized by the encoder's type (NSC or RSC). Gray mapping is considered. The simulation parameters, for this figure, are provided in Table 2.3. This figure shows that, for small SNR values (i.e., when SNR $\leq$ 3dB) NSC code outperforms RSC code, but for high SNR values (i.e. when SNR $>$ 3dB) the opposite happens.

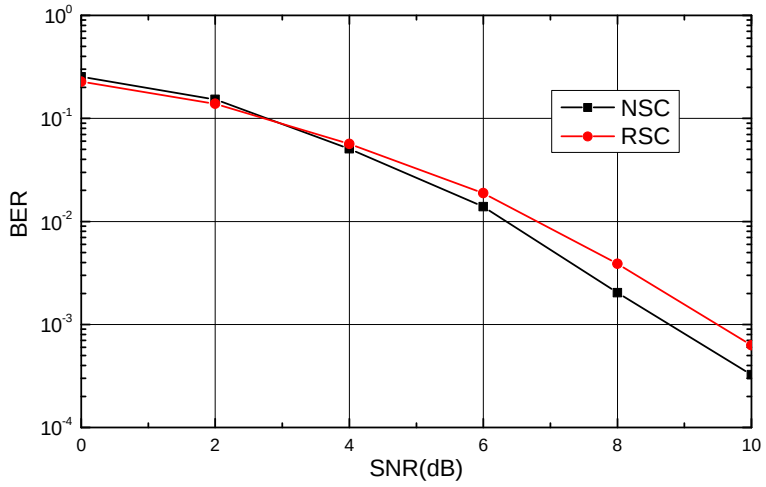


FIGURE 2.13: BER vs. SNR as parameterized by the encoder's type.

2.3.4 The influence of the channel's type

In Fig. 2.14, BER vs. SNR curves are shown parameterized by the channel's type (Rayleigh or AWGN). Gray mapping is considered. The simulation parameters, for this figure, are provided in Table 2.4. This figure shows that BICM has better performance over AWGN channel.

TABLE 2.3: Simulation parameters.

Figure #	Channel	Code generator	Code rate	Mapping	Block length
13	Rayleigh	$(7,5)_8$	$1/2$	NSC or RSC	8192

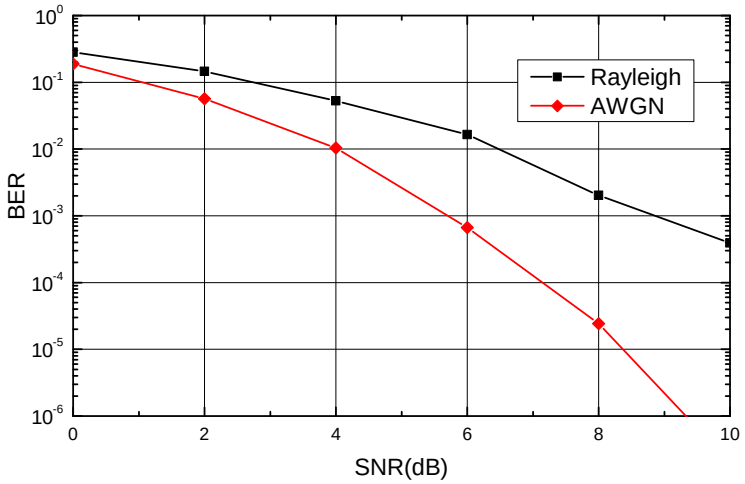


FIGURE 2.14: BER vs. SNR as parameterized by the channel's type.

TABLE 2.4: Simulation parameters.

Figure #	Channel	Code generator	Code rate	Encoder type	Block length
14	Rayleigh or AWGN	$(7,5)_8$	$1/2$	NSC or RSC	8192

2.4 Conclusion

BICM represents an important advancement of spectral efficient coded modulation. The results of simulations, carried out to characterize the BICM's BER performance, show that:

- The block length doesn't have a remarkable effect on the BICM's BER performance over both AWGN and Rayleigh fading channels.
- Gray mapping gives the best performance among the five studied mappings.
- For small SNR values, NSC code outperforms RSC code, but the opposite happens for high SNR values.
- BICM BER performance over AWGN channel is better than it over Rayleigh fading channel.

Bibliography

[1] G. Ungerboeck. "Channel Coding with multilevel phase signals", *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 55-67, Jan. 1982.

[2] D. Divsalar and M. K. Simon. "The design of trellis coded MPSK for fading channels: Performance criteria", *IEEE Trans. Commun*, vol. 36, pp. 1004–1011, Sept. 1988.

[3] E. Zehavi. "8-PSK trellis codes for a Rayleigh fading channel", *IEEE Trans. comm*, vol.40, pp. 873-883, May 1992.

[4] G. Caire, G. Taricco, and E. Biglieri. "Bit interleaved coded modulation", *IEEE Trans. Inform. Theory*, vol. 44, pp. 927-946, May 1998.

[5] S. Lin and D. Costello. "Error Control Coding: Fundamentals and Applications" (Engelwood Cliffs, NJ, USA: Prentice-Hall, Oct. 1982. ISBN: 013283796X).

[6] S. Haykin. "Communication Systems"(Fourth edition. John Willy & Sons, Inc. 2001. ISBN: 0-471-17869-1).

[7] X. Li and J. A. Ritcey. "Trellis-Coded Modulation with Bit Interleaving and Iterative Decoding", *IEEE Journal on selected areas in commun*, vol. 17, pp. 715-724, April. 1999.

[8] J. Tan, Stuber, G. L. "Analysis and design of Symbol mappers for Iteratively Decoded BICM", *IEEE Trans. Wireless. commun*. vol.2, pp. 662-672, March. 2005.

[9] X. Li, A. Chindapol and J. A. Ritcey. "Bit interleaved coded modulation with iterative decoding and 8PSK signaling", *IEEE Trans. commun*. vol. 50, pp. 1250-1257, Aug. 2002.

Chapter 3

Turbo Codes & Iterative Decoding

In this chapter, We'll go through the theory underlying iterative decoding of parallel concatenated convolutional codes (i.e. turbo codes). We'll talk, in details, about the BCJR algorithm used in their decoding. We'll carry out simulations to characterize their Bit Error Rate (BER) performance.

3.1 Fundamentals of Turbo Codes

Parallel concatenation of convolutional codes was first proposed by *Berrou et al* [1]. Because of the iterative decoding process that resembles the turbo engine's functionality, authors in [1], called this type of codes Turbo Codes. Essentially, a turbo encoder consists of the parallel concatenation of two or more Recursive Systematic (RSC) convolutional codes separated by an interleaver.

3.1.1 Recursive Systematic convolutional codes

A convolutional code is systematic if one of the output sequences is exactly the input sequence, otherwise it is non-systematic (NSC) [1]. In Fig.3.1(a), we show a systematic convolutional code of rate $1/2$, constraint length $k = 3$, 8-states and its generator polynomial is $G = [g_1, g_2] = [7, 5]_8$. If

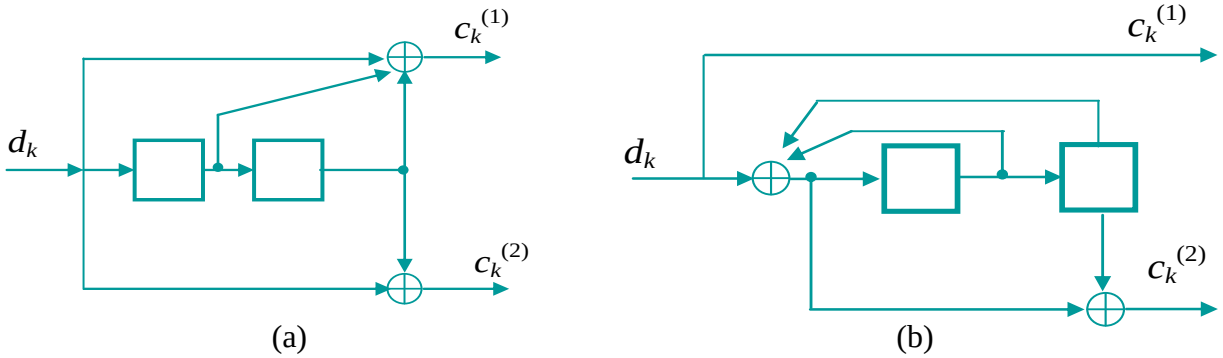


FIGURE 3.1: Rate-1/2 convolutional codes. (a) a NSC code. (b) a RSC code [1].

the state of the internal shift register depends on the past output, this code is said to be recursive, otherwise it is non-recursive. The code in Fig. 3.1(a) is a non-recursive, where the code in Fig. 3.1(b) is its equivalent RSC. A RSC encoder can be obtained from a NSC one by setting one of the outputs equal to the input (make it systematic) and using feedback in the encoding process, i.e. we need to transform the non-systematic feed-forward code generator into a systematic feed-back generator.

In Fig 3.1(a), the feed-forward code generator is $G_{NSC}(D) = [g_1(D), g_2(D)] = [1 + D + D^2, 1 + D^2]$ which must be transformed into $G_{RSC}(D) = [1, g_1(D)/g_2(D)] = [1, (1 + D^2)/(1 + D + D^2)]$ to construct the feed-back code generator. The trellis diagram for the RSC code depicted, in Fig. 3.1(b), is shown in Fig. 3.2. A NSC code and its equivalent RSC code have the same distance properties.

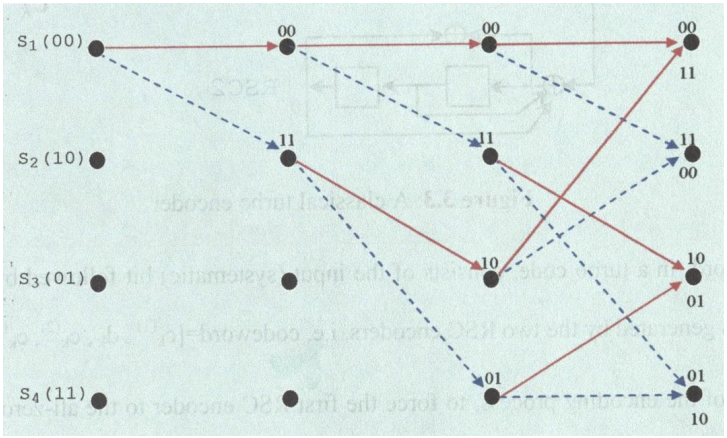


FIGURE 3.2: Trellis diagrams for the RSC cc depicted in figure 3.1 (b).

i.e. the same free distance [1]. The code generators are often expressed in an octal form. For the RSC code in 3.1(b), we write $G = [1, 5/7]_8$. RSC code is the basic building block in a turbo encoder.

3.1.2 Turbo encoder

In Fig. 3.3, a classical turbo encoder is shown. There are two RSC encoders concatenated in parallel. Before entering to the second RSC encoder, the information bits are to be interleaved using a bit-level interleaver. Interleaving plays an important role in the performance of turbo codes [1]. The

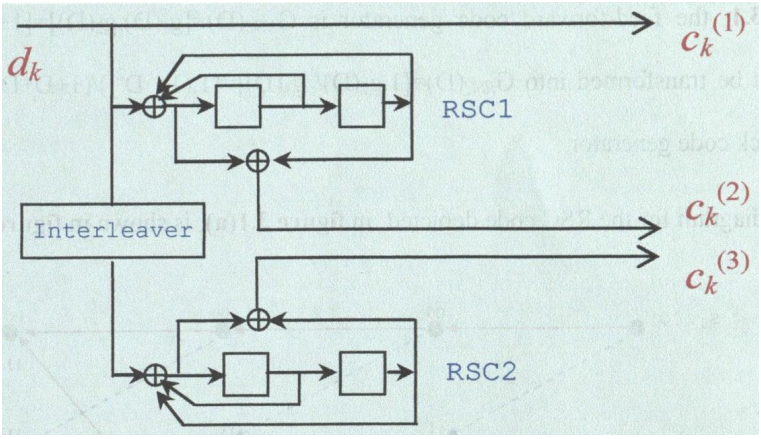


FIGURE 3.3: A classical turbo encoder [1].

code-word, in a turbo code, consists of the input (systematic) bit followed by the two other (parity) bits generated by the two RSC encoders. i.e. codeword $c_k == [c_k(1) = d_k, c_k(2), c_k(3)]$.
At the end of the encoding process, to force the first RSC encoder to the all-zero state, flushing bits are added. For the second RSC encoder, it may or may not be terminated to zero [3]. Each RSC

encoder in a turbo code is called a component or constituent encoder. The rate of the code in figure 3.3 is $R = 1/3$. To obtain a higher rate, this code may be punctured. For example, the puncturer in Fig. 3.4, Chooses odd bits from RSC1 and even bits from RSC2 to increase the code rate to $1/2$.

3.1.3 Iterative Decoding of turbo codes

Fig. 3.5 shows the idea of turbo decoding introduced in [1]. The first observation we can make, regarding the structure of this turbo decoder is that it consists of a serial concatenation of two decoders. The first decoder receives, from the channel, noisy versions of the systematic bits and of the parity bits generated by the first RSC encoder. At this level, we observe a de-multiplexer that allows sending, to each decoder the parity bits generated by its corresponding encoder. As shown

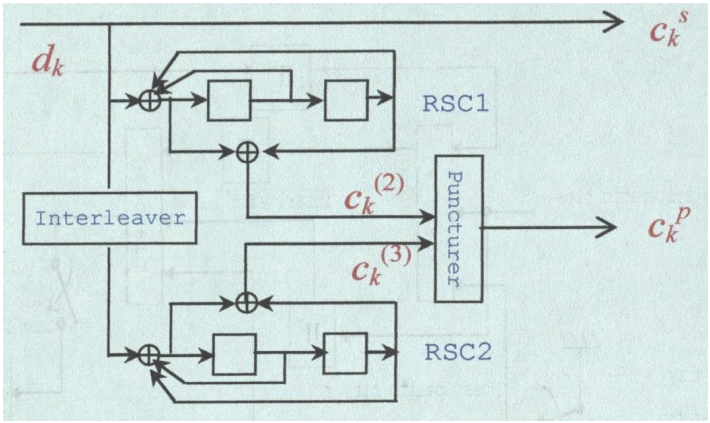


FIGURE 3.4: A rate-1/2 punctured turbo encoder [1].

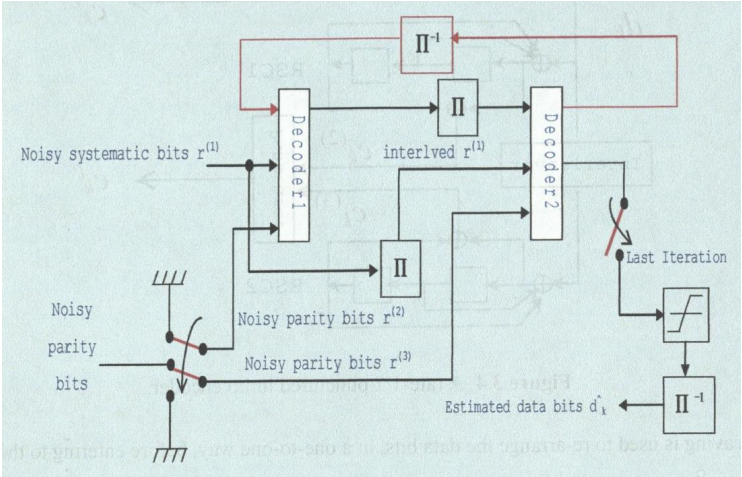


FIGURE 3.5: The structure of a turbo decoder [1].

in Fig. 3.5, the turbo decoder operates on noisy versions of systematic bits $r^{(1)}$ and the two sets of the parity bits $r^{(2)}$ and $r^{(3)}$, in two serial concatenated stages (decoders), to estimate the original information bits.
First, $r^{(1)}$ and $r^{(2)}$ are fed to decoder 1. This later performs decoding and produces extrinsic Log-Likelihood Ratio (LLR) information of systematic bits. Before application to decoder 2, this extrinsic

LLR information is interleaved, to compensate for the interleaving introduced in the turbo encoder. In addition, $r^{(2)}$ the noisy version of parity bits generated by encoder 2 and the interleaved version of $r^{(1)}$ are used as two other inputs to decoder 2. By using these inputs, decoder 2 produces a more refined estimate of the original information bits. This estimate is de-interleaved and transferred back to decoder 1 to begin a new iteration.
Each of the two decoders uses a Bahl, Cocke, Jelinek, and Raviv (BCJR) algorithm [2]. This later was first proposed to solve a MAP detection problem. To decode a convolutional code using a BCJR algorithm, two assumptions have to be considered. (1) The convolutional encoding operation is modeled as a Markov process, and (2) the channel is assumed memoryless.

3.1.4 The BCJR algorithm

Let us consider the rate$-1/2$ RSC encoder given in figure 3.6. A binary PSK modulator is used after this RSC encoder to map the binary bits 0 and 1 into BPSK symbols -1 and + 1. As shown in Fig.

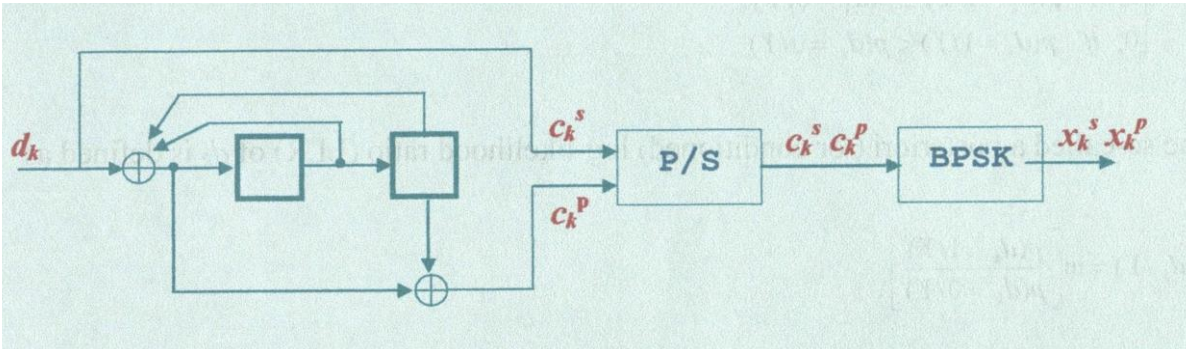


FIGURE 3.6: A transmitter in a turbo encoding process [3].

3.6, we will follow the same steps in references [3] and [4]:

- d_k is the information bit at time k. c_k^s and c_k^p are respectively the systematic bit and the parity bit corresponding to d_k. They all take on a value 0 or 1.
- $x_k = (c_k^s, c_k^p)$ is the binary PSK modulated pair corresponding to (c_k^s, c_k^p). They take on a value +1 or -1.
- $X^s = (x_1^s, x_2^s, ..., x_N^s)$ is the systematic sequence which corresponds to the information sequence to be coded $D = (d_1, d_2, ..., d_N)$.
- $X^p = (x_1^p, x_2^p, ..., x_N^p)$ is the non-systematic sequence which corresponds to the information sequence to be coded $D = (d_1, d_2, ..., d_N)$.
- The pair $y_k = (y_k^s, y_k^p)$ is the noisy version, of pair $x_k = (c_k^s, c_k^p)$ coming out of the channel.
- $Y = (y_1, y_2, ..., y_N) = (y_1^s, y_1^p, y_2^s, y_2^p, ..., y_N^s, y_N^p)$ is the noisy sequence observed by the decoder.
- m is the memory of the RSC encoder. The encoder is of 2^m states.
- S is the set of all 2^m RSC encoder's states, i.e. $S = \{s_1, s_2, ..., s_{2m}\}$ where s_k is the state of the encoder at time k.

- S^+ is the set of pairs s', s corresponding to all state transitions from $s_k = s'$ to $s_k = s$ caused by information bit $d_k = 1$. Similarly, S^- is defined for $d_k = 0$.

The maximum a posteriori (MAP) decoding is carried out as follows

$$d_k = \begin{cases} 1, \text{ if } p(d_k = 1/Y) \geq p(d_k = 0/Y) \\ 0, \text{ if } p(d_k = 1/Y) < p(d_k = 0/Y) \end{cases} \tag{3.1}$$

The so-called a posteriori (or conditioned) LLR of d_k is defined as

$$L(d_k/Y) = \frac{p(d_k = 1/Y)}{p(d_k = 0/Y)} \tag{3.2}$$

The numerator and denominator of equation (3.2) are a posteriori conditional probabilities. The sign of $L(d_k/Y)$ is an indication of which information bit d_k, 1 or 0, was coded at time k. The magnitude of $L(d_k/Y)$ is a measure of the likelihood (reliability or confidence) of the decision we have made: the more the $L(d_k/Y)$ magnitude is far away from the zero threshold decision the more we trust in the decision we have made [3].
Equation (3.1) can be re-written as follows

$$d_k = \begin{cases} 1, \text{ if } L(d_k/Y) \geq 0 \\ 0, \text{ if } L(d_k/Y) < 0 \end{cases} \tag{3.3}$$

By applying Bayes' rule to equation (3.2), we obtain

$$L(d_k/Y) = ln \frac{p(d_k = 1, Y)}{p(d_k = 0, Y)} \tag{3.4}$$

By observing the trellis structure of the RSC encoder, if we know the previous state $S_{k-1} = s'$ and the present state $S_k = s$, the information bit caused this transition is therefore known.
By taking into account that transitions between the previous state S_{k-1} and the present state S_k are mutually exclusive (just one can occur at a given time) the probability that any transition occurs is equal to the sum of the individual probabilities, that is,

$$p(d_k = 1, Y) = \sum_{s^+} p(S_{k-1} = s', S_k = s, Y) \tag{3.5}$$

$$p(d_k = 0, Y) = \sum_{s^-} p(S_{k-1} = s', S_k = s, Y) \tag{3.6}$$

Equation (3.4) will be reformulated as follows

$$L(d_k/Y) = ln \left[\frac{\sum_{s^+} p(S_{k-1} = s', S_k = s,)}{\sum_{s^-} p(S_{k-1} = s', S_k = s, Y)} \right] \tag{3.7}$$

We can partition the received $N-$symbols sequence Y into three sub-sequences: a past sequence $Y_{<k}$, a present sequence (i.e., at time k) Y_k, and a future sequence $Y_{>k}$:

$$Y = Y_{<k}, Y_k, Y_{>k} = y_1, y_2, ..., y_k, y_{k+1}, y_{k+2}, y_N \tag{3.8}$$

Then, we have

$$p(S_{k-1} = s', S_k = s, Y) = p(S_{k-1} = s', S_k = s, Y_{<k}, Y_k, Y_{>k}) \tag{3.9}$$

By using Bayes' rule, we have

$$p(S_{k-1} = s', S_k = s, Y) = p(Y_{>k}/S_{k-1} = s', Y_{<k}, Y_k).p(S_{k-1} = s', Y_{<k}, Y_k) \tag{3.10}$$

For a memoryless channel, the current signal $Y_{>k}$ depends only on the current states (it depends neither on the previous states' nor on the past sequence $Y_{<k}$ or current sequence Y_k). Therefore,

$$p(Y_{>k}/S_{k-1} = s', Y_{<k}, Y_k) = p(Y_{>k}/S_k = s) \tag{3.11}$$

$$\begin{aligned} p(S_{k-1} = s', Y_{<k}, Y_k) &= p(Y_k, S_k = s/S_{k-1} = s', Y_{<k}).p(S_{k-1} = s', Y_{<k}) \\ &= p(Y_k, S_k = s, Y_{<k-1} = s').p(S_{k-1} = s', Y< k) \end{aligned} \tag{3.12}$$

By substituting equations (3.11) and (3.12) in equation (3.10), we obtain

$$\begin{aligned} p(S_{k-1} = s', Y_{<k}, Y_k) = p(Y_{>k}/S_k = s).p(Y_k, S_k = s/Y_{<k-1} = s').p(S_{k-1} = s', Y< k) \\ = \beta_k(s).\gamma_k(s', s).\alpha_{k-1}(s') \end{aligned} \tag{3.13}$$

where,

- $\beta_k(s) = p(Y_{>k}/s)$ represents the conditional probability that given current state is s , the future sequence will be $Y_{>k}$.
- $\gamma_k(s', s) = p(Y_k, s/s')$ represents the probability that next state is sand the received symbol is Y_k given the previous state s'.
- $\alpha_{k-1}(s') = p(s', Y< k)$ represents the joint probability that at time $k-1$ the state is s' and the received sequence until then is $Y< k$.

Now, by substituting equation (3.13) in equation (3.7), we obtain

$$L(d_k/Y) = ln\left[\frac{\sum_{s^+} \beta_k(s).\gamma_k(s', s).\alpha_{k-1}(s')}{\sum_{s^-} \beta_k(s).\gamma_k(s', s).\alpha_{k-1}(s')}\right] \tag{3.14}$$

Now, we present how the BCJR algorithm computes the three probabilities $\alpha_{k-1}(s')$, $\beta_k(s)$ and $\gamma_k(s', s)$ for a rate$-1/2$ RSC encoder.
Firstly, $\gamma_k(s', s)$ is computed as follows

$$\gamma_k(s', s) = c_k.exp\left(\frac{d_k.L(d_k)}{2}\right).exp\left(\frac{L_c}{2}\sum_{i=1}^{2} x_k^i y_k^i\right) \tag{3.15}$$

where c_k is some constant that can be ignored because it will cancel out since it appears both in the numerator and denominator of equation (3.14).
The channel reliability value L_c is given by

$$L_c = 4gR(E_b/N0) \tag{3.16}$$

where $N0/2$ is the noise bilateral power spectral density,g is the fading amplitude (for non-fading channel $g = 1$, R is the RSC code's rate, and E_b is the bit's energy. $L(d_k)$ is the LLR for the information bit d_k, and it is given by

$$L(d_k) = ln\left[\frac{p(d_k = 1)}{p(d_k = 0)}\right] \tag{3.17}$$

Secondly, $\alpha_k(s)$ and $\beta_{k-1}(s')$ are computed recursively as follows

$$\alpha_k(s) = \sum_{s^+} \alpha_{k-1}(s')\gamma_k(s', s) \tag{3.18}$$

$$\beta_{k-1}(s') = \sum_{s^+} \beta_{k-1}(s)\gamma_k(s', s) \tag{3.19}$$

Since the BCJR algorithm assumes the trellis begins with and ends at the all-zero state $S_0 = 0$, the initialization conditions for $\alpha_k(s)$ and $\beta_{k-1}(s')$ are

$$\alpha_0(s_i) = \begin{cases} 1, \text{ for } s_i = s_0 \\ 0, \text{ for } s_i \neq s_0 \end{cases} \tag{3.20}$$

$$\beta_N(s_i) = \begin{cases} 1, \text{ for } s_i = s_0 \\ 0, \text{ for } s_i \neq s_0 \end{cases} \tag{3.21}$$

The BCJR algorithm uses equations (3.15), (3.18), and (3.19) to evaluate equation (3.14). α, β and γ are referred to as the forward, backward and transition metrics, respectively. Because of this the BCJR algorithm is also known as the forward-backward algorithm.
Due to the iterative nature of turbo-decoding, the forward and backward metrics may face a numerical overflow or underflow. To circumvent this problem, α and β should be normalized as follows

$$\alpha_k^{'}(s) = \frac{\alpha_k(s)}{\sum_s \alpha_k(s)} \tag{3.22}$$

$$\beta_{k-1}^{'}(s) = \frac{\alpha_k(s')}{\sum_(s')\beta_k(s)} \tag{3.23}$$

So that

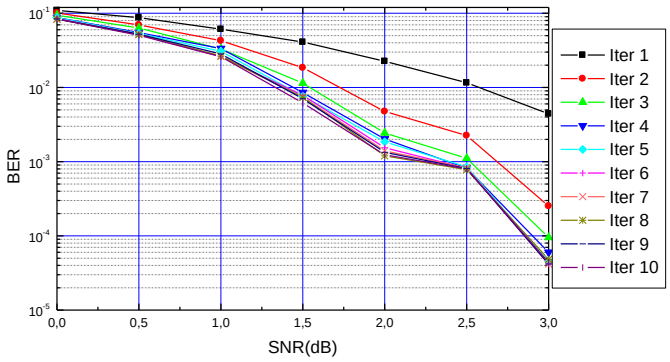
$$\sum_s \alpha_k^{'}(s) = \sum_{s'} \beta_{k-1}^{'}(s) = 1 \tag{3.24}$$

3.2 Performance evaluation of Turbo codes

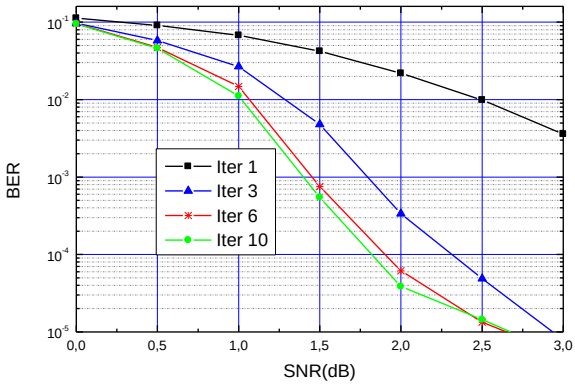
After studying the fundamental theory underlying turbo codes in the previous section we now undertake a comprehensive simulation to characterize the BER performance of turbo codes. BER vs. SNR curves are used to study the effects of various system's parameters on its performance.

3.2.1 The effect of the number of iterations

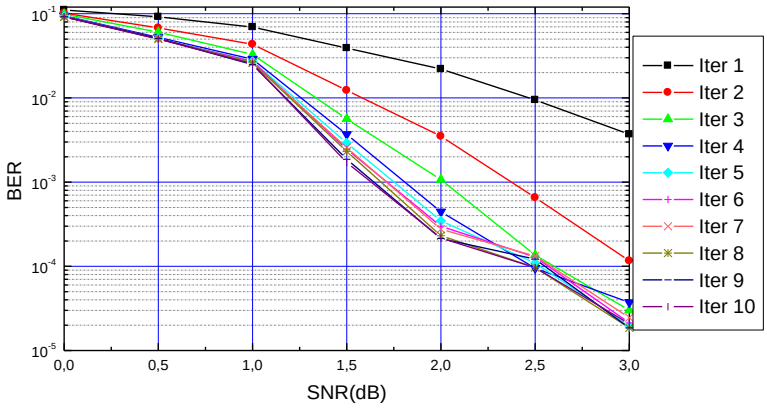
In Fig. 3.7 to 3.10, we show the influence of iteration number on the BER performance of turbo codes. Simulation parameters, for these figures, are provided in table 3.1.



(a) frame length= 200.

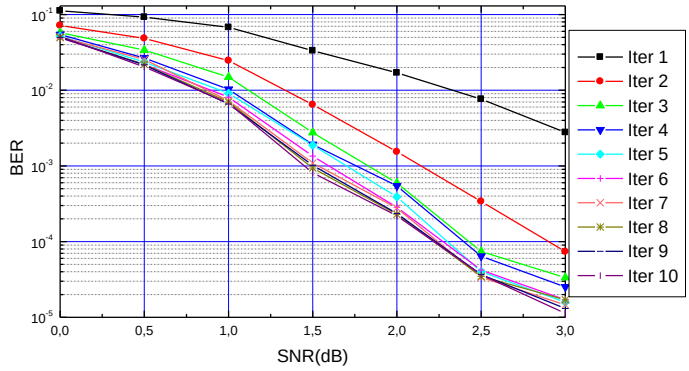


(b) frame length= 400.

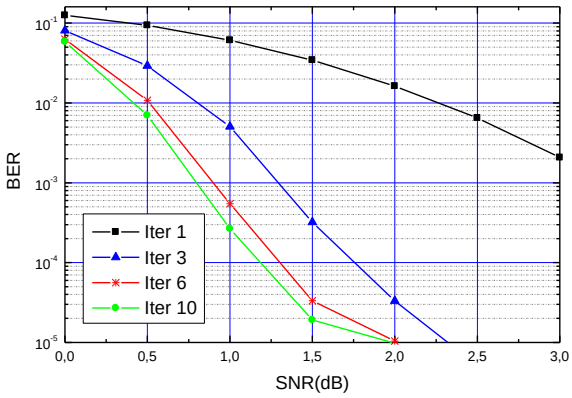


(c) frame length= 1000.

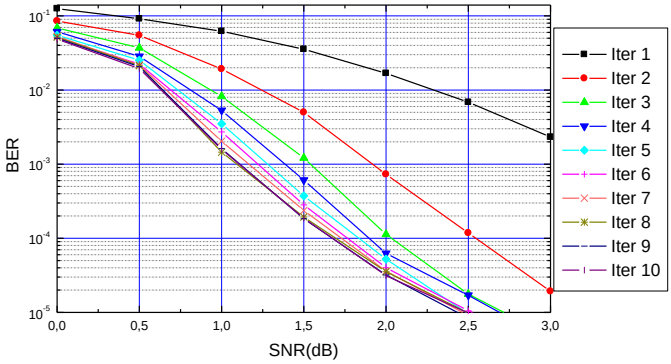
FIGURE 3.7: BER vs. SNR as parameterized by the number of iterations ($g = (1, 5/7)_8$, rate= $1/2$).



(a) frame length= 200.

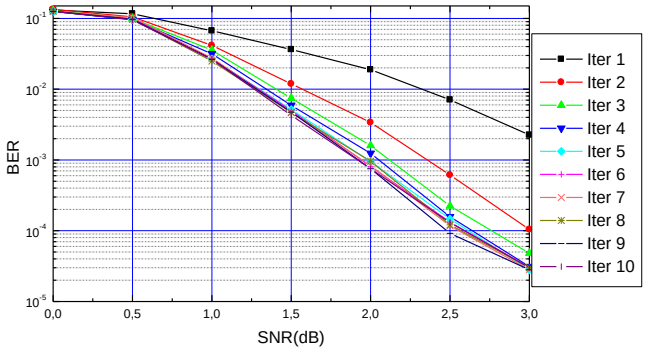


(b) frame length= 400.

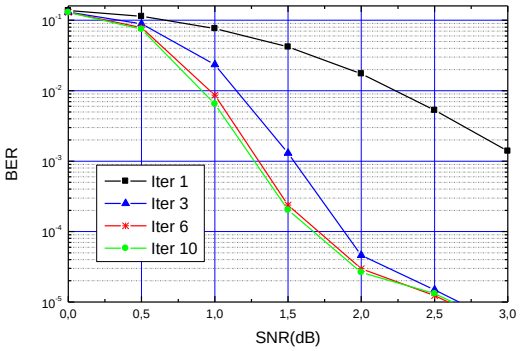


(c) frame length= 1000.

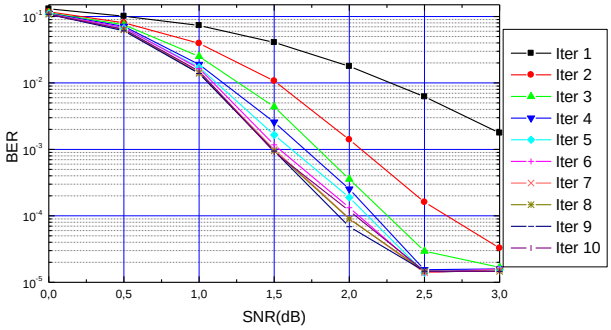
FIGURE 3.8: BER vs. SNR as parameterized by the number of iterations ($g = (1, 5/7)_8$, rate= $1/3$).



(a) frame length= 200.

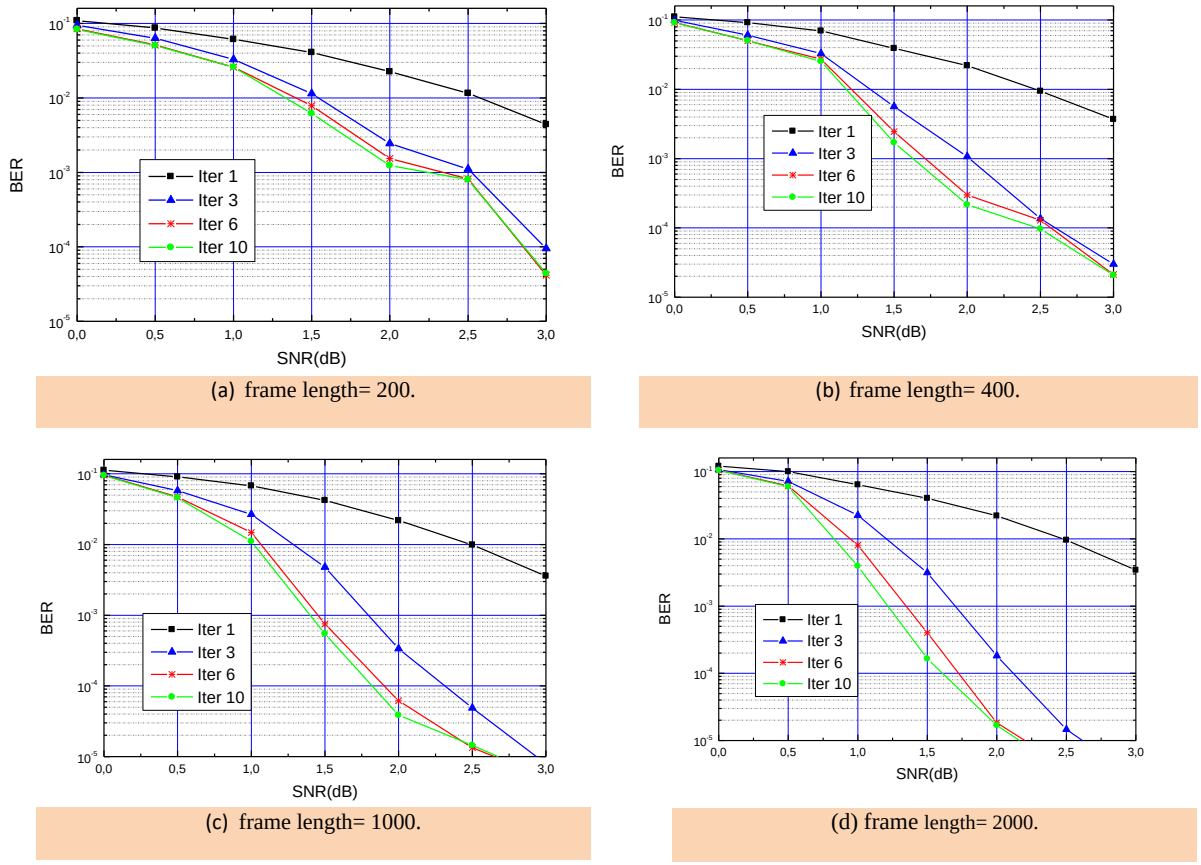


(b) frame length= 400.



(c) frame length= 1000.

FIGURE 3.9: BER vs. SNR as parameterized by the number of iterations ($g = (1, 21/37)_8$, rate= $1/2$).



(a) frame length= 200.

(b) frame length= 400.

(c) frame length= 1000.

(d) frame length= 2000.

FIGURE 3.10: BER vs. SNR as parameterized by the number of iterations ($g = (1, 21/37)_8$, rate= $1/3$).

TABLE 3.1: Simulation parameters.

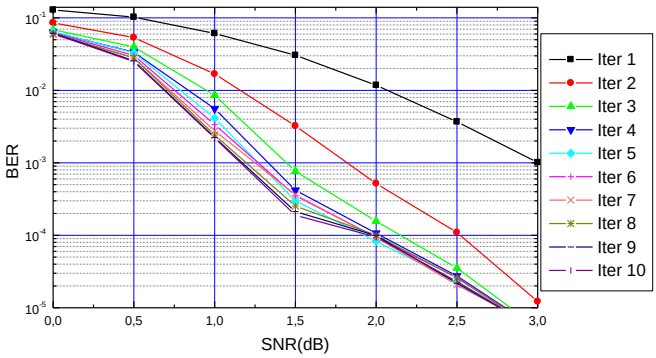
Figure #	**Code generator**	**Code rate**	**Frame sizes**	**Max. Number of iterations**
3.7	$(1, 5/7)_8$	$1/2$	200, 400, and 1000	10
3.8	$(1, 5/7)_8$	$1/3$	200, 400, and 1000	10
3.9	$(1, 21/37)_8$	$1/2$	200, 400, and 1000	10
3.10	$(1, 21/37)_8$	$1/3$	200, and 400	10

These figures show that as the number of iterations increases, the BER decreases and tends to

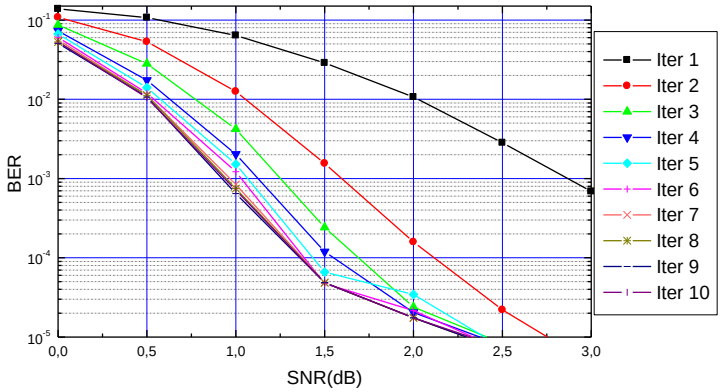
converge. When the frame size is small (200 and 400), to make the BER converge we need few number of iterations (3 to 5 iterations) and by increasing the number of iterations above 5, we obtain a small gain.
Also, we can observe that the non-punctured turbo code (Rate$= 1/3$) has a significant coding gain compared to the punctured one (Rate$= 1/2$). For example, for a SNR$= 1.5$ dB we can attain a BER$= 2.10^{-5}$ when we use a non-punctured turbo code (Fig 3.8 (c)) compared to a BER$= 5,5.10^{-4}$ when we use a punctured turbo code (Fig 3.7 (c)). Comparing Fig 3.7 to Fig 3.9 and Fig 3.8 to 3.10, we can observe that increasing the constraint length deceases the BER at the region where BER is smaller than 10^{-3}.

3.2.2 The effect of the frame size

In Fig. 3.11 to 3.12, we show the influence of the frame size on the BER performance of turbo codes. Simulation parameters, for these figures, are provided in table 3.2.

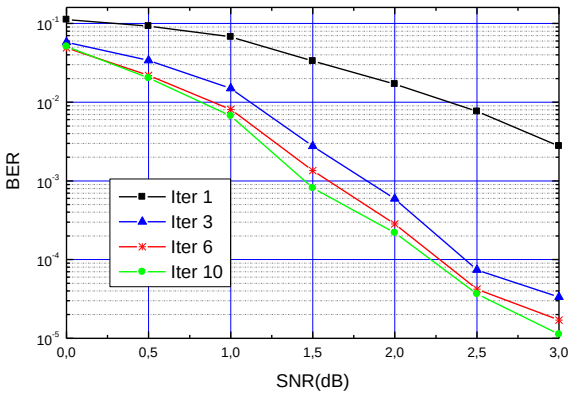


(a) frame length= 200.

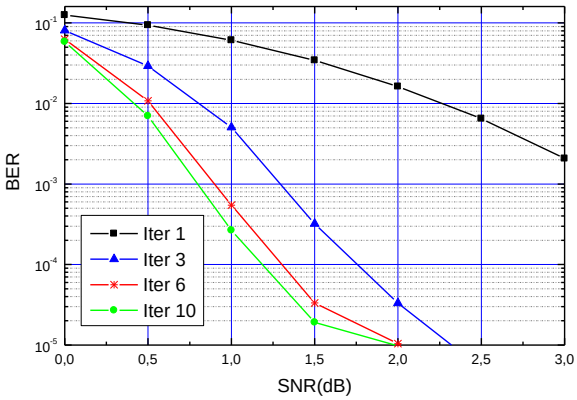


(b) frame length= 400.

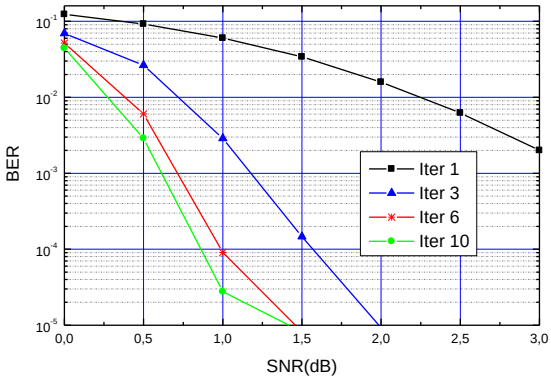
FIGURE 3.11: BER vs. SNR as parameterized by the frame size ($g = (1, 5/7)_8$, rate= $1/2$).



(a) frame length= 200.



(b) frame length= 400.



(c) frame length= 400.

FIGURE 3.12: BER vs. SNR as parameterized by the frame size ($g = (1, 5/7)_8$, rate= $1/3$).

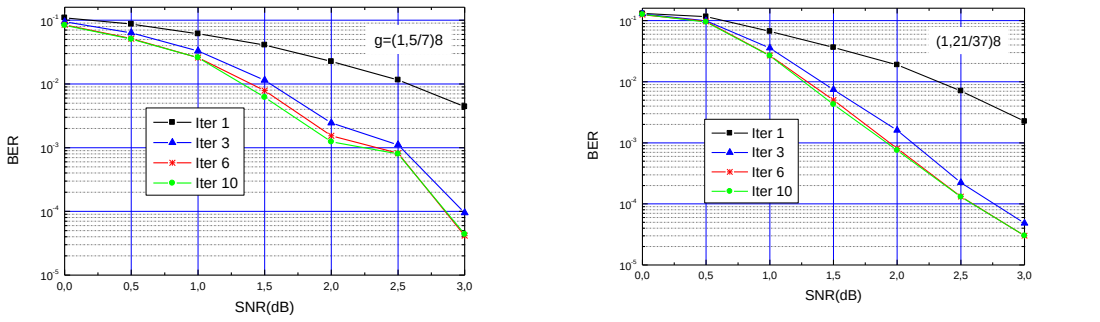
TABLE 3.2: Simulation parameters.

Figure #	**Code generator**	**Code rate**	**Frame sizes**	**Max. Number of iterations**
3.11	$(1, 5/7)_8$	$1/2$	200, 400, 100 and 1000	10
3.12	$(1, 5/7)_8$	$1/3$	200, 400, and 1000	10

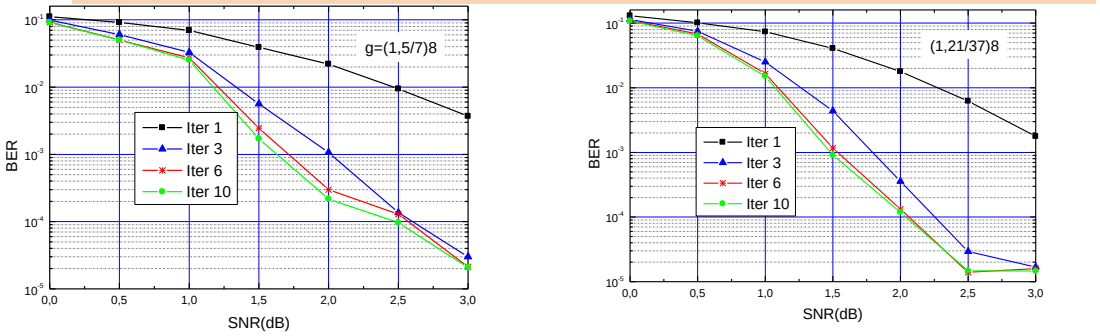
These figures show the importance of the frame size in the design of turbo codes. Generally, a large frame size corresponds to a lower BER.

3.2.3 The effect of the code rate

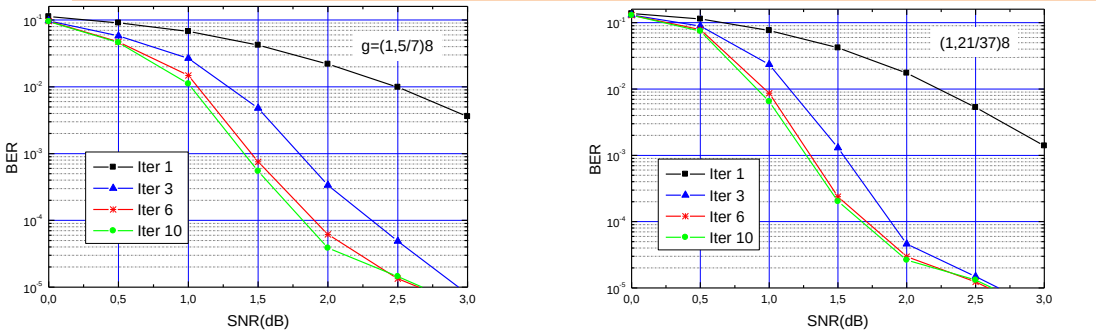
In Fig. 3.13 and 3.14, we show the influence of the code rate on the BER performance of turbo codes. Simulation parameters, for these figures, are provided in table 3.3.



(a) frame length= 200.

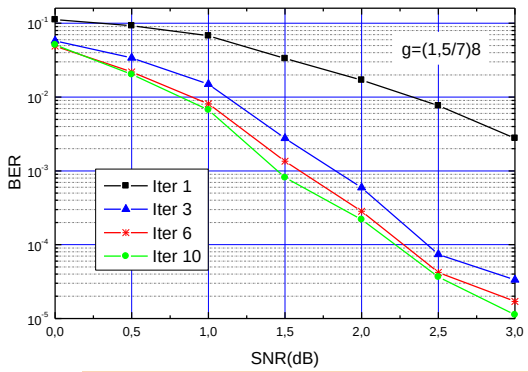


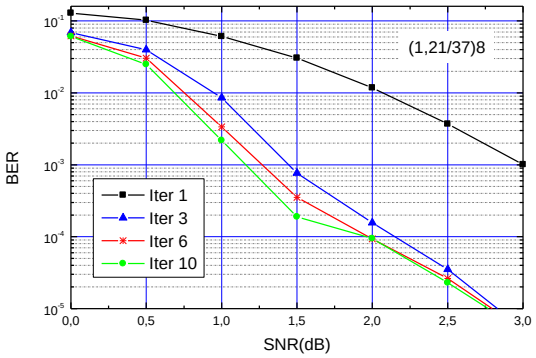
(b) frame length= 400.



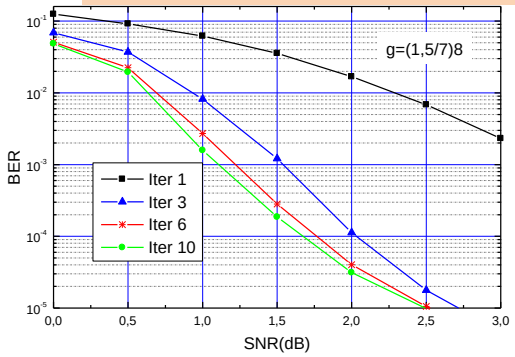
(c) frame length= 1000.

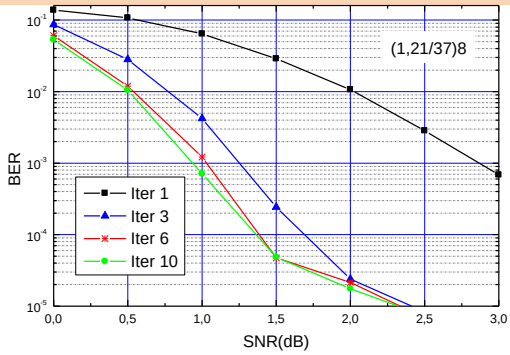
FIGURE 3.13: BER vs. SNR as parameterized by the code rate ($g = (1, 5/7)_8$, and $g = (1, 21/37)_8$ rate= $1/2$).



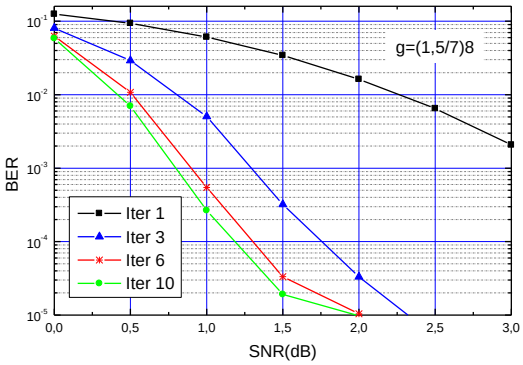


(a) frame length= 200.





(b) frame length= 400.



(c) frame length= 1000.

FIGURE 3.14: BER vs. SNR as parameterized by the code rate ($g = (1, 5/7)_8$, and $g = (1, 21/37)_8$ rate= $1/3$).

TABLE 3.3: Simulation parameters

Figure #	**Code generator**	**Code rate**	**Frame sizes**	**Max. Number of iterations**
3.13	$(1, 5/7)_8$ and $(1, 21/37)_8$	$1/2$	200, 400, and 1000	10
3.14	$(1, 5/7)_8$ and $(1, 21/37)_8$	$1/3$	200, 400, and 1000	10

From these two figures, we can observe that for a fixed frame size and a code rate but different constraint lengths, the BER curves diverges as the SNR increases. This divergence is more clear for large frame size.

3.3 Conclusion

This chapter was devoted to turbo codes to understand the theory underlying them and to characterize their BER performance.
The results of the simulations, carried out to characterize the turbo codes performance show that:

- The number of iterations has remarkable effect on the turbo codes BER performance where we've noticed that as we increase the number of iterations, the BER decreases and tends to converges.
- The frame size is an important parameter in the design of turbo codes. Generally, a large frame size corresponds to a lower BER.
- For a fixed code rate and a fixed frame size but with different constraint lengths, the BER curves diverges as the SNR increases. This divergence is more clear for large frame size.

Bibliography

[1] C. Berrou, A. Glavieux, "Near Optimum Error-Correcting Coding and Decoding", *IEEE Trans. Comm*, vol. 44, pp. 1261-1271, Oct 1996.

[2] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv " Optimal decoding of linear codes for minimizing symbol error rate", *IEEE Trans. on Inform. Theory*, vol. IT-20, pp. 284-287, March 1974.

[3] Silvio A. Abrantes, " From BCJR to turbo decoding: MAP algorithms made easier," Faculdade de Engenharia da Universidade do Porto (FEUP), Porto, Portugal. April 2004.

[4] Lin, S. and Costello, D.J. Jr. Error Control Coding, Prentice-Hall, Englewood Cliffs, NJ, 2004.

Chapter 4

Bit-Interleaved Coded Modulation with Iterative Decoding

In this chapter, We'll cover the basic theory underlying iteratively decoded Bit-Interleaved Coded Modulation (BICM). We'll present its two decision feedback methods. We'll talk about BICM-ID using punctured convolutional codes. Finally, we'll present some techniques used to reduces BICM-ID's receiver complexity.

4.1 Introduction

For a coded modulation system, an optimal maximum a posteriori (MAP) decision can be performed if the receiver can compute the overall a posteriori probability regarding the information bit d_k. i.e., $p(d_k = b/r_i, C, \Psi)$. Where the conditioning on C and Ψ denotes respectively the knowledge of the code structure and the knowledge on the modulation process. Such an optimal receiver which can perform joint demodulation and decoding is illustrated in fig 4.1.

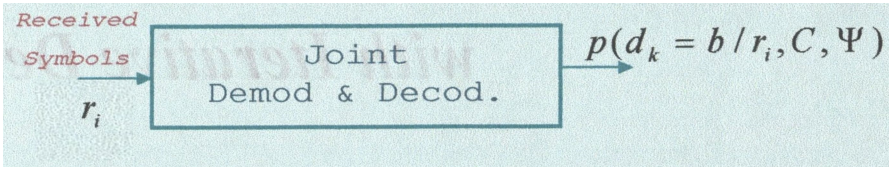


FIGURE 4.1: Optimal joint demodulation and decoding [1].

For BICM, the presence of the interleaver between the modulator and the channel encoder makes a joint demodulation and decoding tremendously complex [1]. This problem is similar to that of decoding turbo codes [2]. This suggest that an iterative structure is also an efficient sub-optimal technique to solve this problem and that good performance can still be achieved [3]. For conventional BICM, the information travels only in a single direction from the demodulator to the decoder. When we use iterative decoding, the additional information available at the output of the channel decoder and that travels back to the demodulator can potentially improve the detection accuracy. The iterative structure will allow the demodulator to take into consideration the likelihood of the transmitted sequence computed by using the code structure, and thus to take into account the structure imposed by the channel code C.
Reference [3] is the first paper that applied iterative decoding for BICM where a BICM's iterative decoder with hard-decision feedback was proposed. In [3] and [4], it was shown that BICM with

hard-decision feedback significantly outperforms Trellis Coded Modulation (TCM) over Rayleigh fading channels and compares favorably with it over AWGN channels. Also, through the use of punctured codes, BICM-ID provides a simple mechanism for variable rate transmission. BICM-ID with hard-decision feedback suffers a performance degradation when the feedback contains errors. The key solution to this problem was proposed in [5] where soft-decision feedback was introduced. In [5], it was shown that BICM with soft-decision feedback significantly outperforms TCM and performs closely to turbo-TCM over both AWGN and Rayleigh fading channels.

4.2 BICM with hard decision feedback

4.2.1 The principle of hard decision feedback

BICM-ID system has the same transmitter as conventional BICM. For the receiver, the binary hard-decision from the output of the Viterbi algorithm is transferred back to the demodulator after being interleaved (Fig 4.2). The exact procedure is as follows [6]:

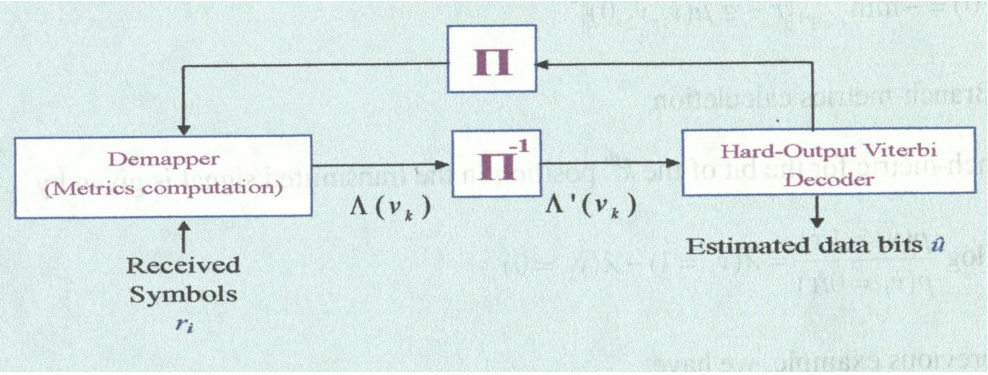


FIGURE 4.2: Hard decision feedback for BICM-ID [6].

1. The first round of decoding is identical to that of BICM.
2. For any next iteration, the bit metric $\lambda(v_k = b)$ is re-calculated as follows

This example illustrates the calculation of $\lambda(v_k = 1)$ with an 8-ary signal constellation.The a priori information for any signal s_i with label $s_i = \mu(v_1 = 1, v_2, v_3) \in \Psi_1^1$ is given by

$$p(s_i) = \begin{cases} 1, \text{ if } v_1 = 1, v_2 = \hat{v}_2, v_3 = \hat{v}_3 \\ 0, \text{ otherwise} \end{cases} \tag{4.1}$$

Where $\hat{v}_2$ and $\hat{v}_3$ are the results reflected from the previous iteration. The bit-metric for the first bit of the transmitted signal taking on value 1 is

$$\lambda(v_1 = 1) = -min_{s_i \in \Psi_1^1} \|r - g.\mu(1, \hat{v}_2, \hat{v}_3)\|^2 \tag{4.2}$$

The same thing for $s_i = \mu(v_1 = 0, v_2, v_3) \in \Psi_0^1$:

$$p(s_i) = \begin{cases} 1, \text{ if } v_1 = 0, v_2 = \hat{v}_2, v_3 = \hat{v}_3 \\ 0, \text{ otherwise} \end{cases} \tag{4.3}$$

The bit-metric for the first bit of the transmitted signal taking on value 0 is

$$\lambda(v_1 = 0) = -min_{s_i \in \Psi_0^1} \|r - g.\mu(0, \hat{v}_2, \hat{v}_3)\|^2 \tag{4.4}$$

The same thing for $s_i = \mu(v_1 = 0, v_2, v_3) \in \Psi_0^1$: The bit-metrics for the second and third bits of the transmitted signal are to be computed in the same manner, and their values are

$$\begin{cases} \lambda(v_2 = 1) = -min_{s_i \in \Psi_1^2} \|r - g.\mu(\hat{v}_1, 1, \hat{v}_3)\|^2 \\ \lambda(v_2 = 0) = -min_{s_i \in \Psi_0^2} \|r - g.\mu(\hat{v}_1, 0, \hat{v}_3)\|^2 \end{cases} \tag{4.5}$$

$$\begin{cases} \lambda(v_2 = 1) = -min_{s_i \in \Psi_1^2} \|r - g.\mu(\hat{v}_1, \hat{v}_2), 1\|^2 \\ \lambda(v_2 = 0) = -min_{s_i \in \Psi_0^2} \|r - g.\mu(\hat{v}_1, \hat{v}_2), 0\|^2 \end{cases} \tag{4.6}$$

3. Branch-metrics calculation
The branch-metric for the bit of the k^{th} position in the transmitted signal is given by

$$\Lambda(v_k) = log \left[\frac{p(v_k = 1/r)}{p(v_k = 0/r)} \right] = \lambda(v_k = 1) - \lambda(v_k = 0) \tag{4.7}$$

For the previous example, we have

$$\begin{cases} \Lambda(v_1) = \lambda(v_1 = 1) - \lambda(v_1 = 0) \\ \Lambda(v_2) = \lambda(v_2 = 1) - \lambda(v_2 = 0) \\ \Lambda(v_3) = \lambda(v_3 = 1) - \lambda(v_3 = 0) \end{cases} \tag{4.8}$$

4. After being de-interleaved, the newly generated branch-metrics are used as inputs for a Viterbi decoder.
As mentioned in [3], the minimum Euclidian distance between coded sequences can be made large for BICM-ID by choosing an appropriate signal labeling (i.e. the operation $s_i = \mu(v_1, v_2, v_3)$ which is the key that BICM-ID outperforms conventional BICM and that make it suitable for both Rayleigh fading and AWGN channel.
Fig. 4.3 compares the operations of bit-metric computation, for the first bit $v_1 = 0$, between BICM and BICM-ID. For BICM, Fig. 4.3(a), it is given by

$$\lambda(v_1 = 0) = min(d^2_{010}, d^2_{001}, d^2_{000}) \tag{4.9}$$

For BICM-ID, Fig 4.3(b), it is directly given by

$$\lambda(v_1 = 0) = d^2_{0\hat{v}_2\hat{v}_3} \tag{4.10}$$

Where $\hat{v}_2$ and $\hat{v}_3$ are the previous iteration's decoding decisions for the second and third bits. Bit

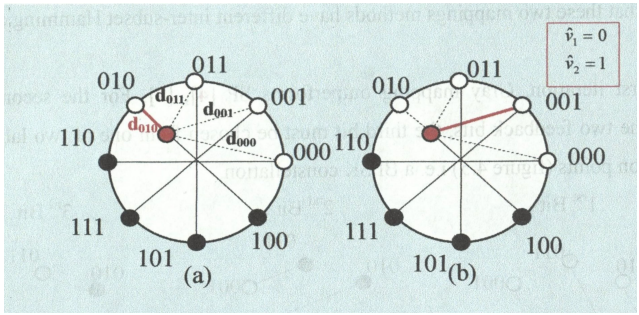


FIGURE 4.3: Comparison of bit-metrics computation in BICM and BICM-ID. (a) BICM. (b) BICM-ID [6].

interleaving, in the upper branch of the iterative receiver, insure the feedback bits' independence of the bit for which we calculate the bit metric.

4.2.2 Signal mappings for BICM-ID

Figure 4.4 illustrates the subset partitioning for each of the three bit positions for two mapping methods namely: Gray and Set Partitioning (SP), for a conventional constellation. It is clear that

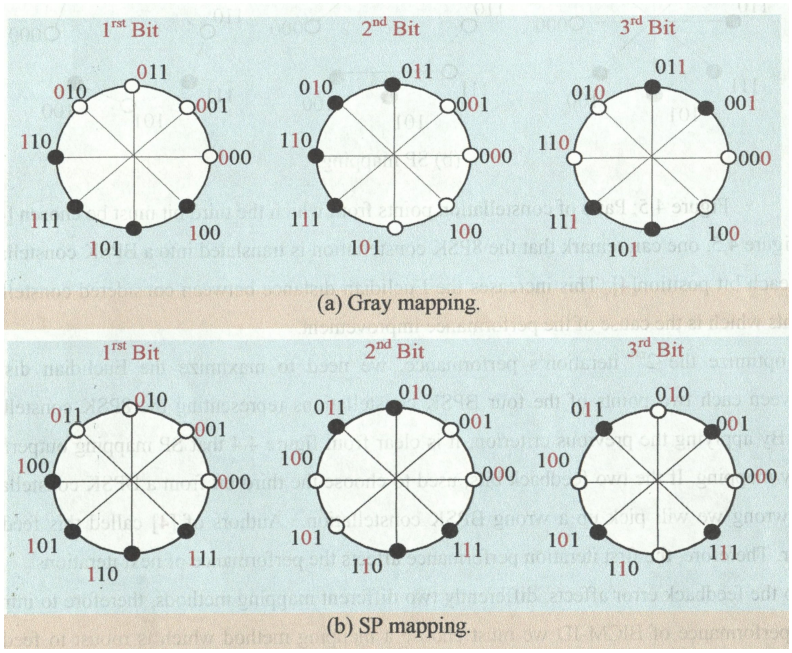


FIGURE 4.4: Gray and SP mappings for conventional $8-$PSK constellation [4].

these two mappings methods have different inter-subset Hamming distance properties. For the first iteration, Gray mapping outperforms SP [4], [7]. For the second iteration, by knowing the two feedback bits, the third bit must be chosen from one of two labels of a pair of constellation points (Fig 4.5) i.e. a BPSK constellation. In Fig 4.5, one can observe that the $8-$PSK constellation is translated into a BPSK constellation, for each bit position [4]. This increases the Euclidian distance

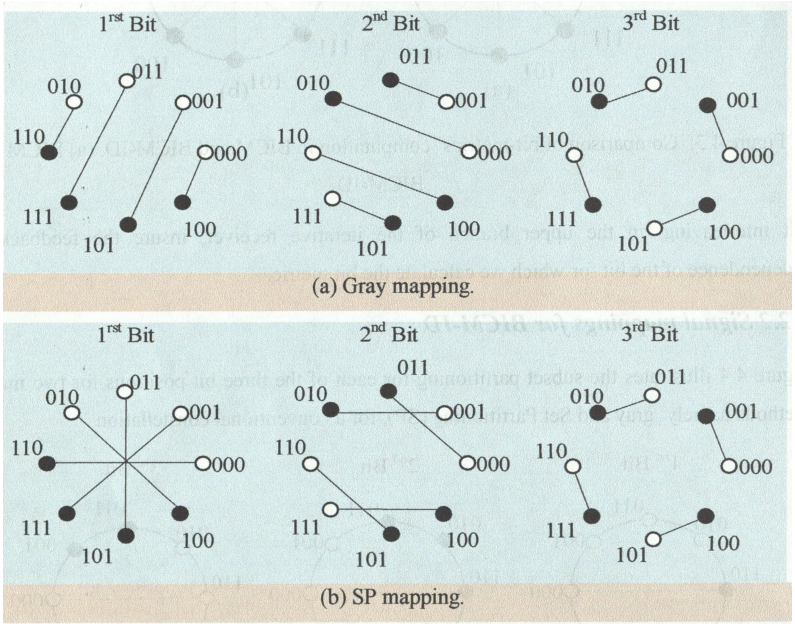


FIGURE 4.5: Pairs of constellation points from which the third bit must be chosen [4].

between considered constellation points which is the cause of the performance improvement. To optimize the 2^{nd} iteration's performance, we need to maximize the Euclidian distance between each two points of the four BPSK constellations representing the $8-$PSK constellation [4]. Therefore, it is clear from figure 4.4 that SP mapping outperforms Gray mapping. If the two feedback bits, used to choose the third bit from a BPSK constellation, are wrong we will pick up a wrong BPSK constellation. Authors of [4] called this feedback error. Therefore, the first iteration performance affects the performance of next iterations. Also the feedback error affects, differently two different mapping methods, therefore to improve the performance of BICM-ID we must choose a mapping method which is robust to feedback errors. Paper [4] proposes Mixed mapping as a compromise between optimizing the first iteration performance and maximizing the improvement provided by iterative decoding over both AWGN and Rayleigh fading channels.

4.3 BICM with soft decision feedback

As said in the previous section, the feedback data sections that are less affected by noise can enhance the decoding of data sections that are more affected by noise. Also, we know that errors in feedback lead us to choose the wrong constellation point. Therefore, it is decisive to control the error propagation and to reduce the effect of feedback errors. BICM-ID with soft decision feedback [5] was proposed as a solution to this problem. A simplified soft-decision feedback iterative decoder for the BICM-ID is depicted in Fig. 4.6. The SISO de-mapper and the SISO decoder are the MAP decoders for the mapper and the convolutional code, respectively. One iteration includes SISO de-mapping, de-interleaving and SISO decoding, in this order. We consider a fully interleaved frequency non-selective channel. The discrete-time complex baseband received signal can be given

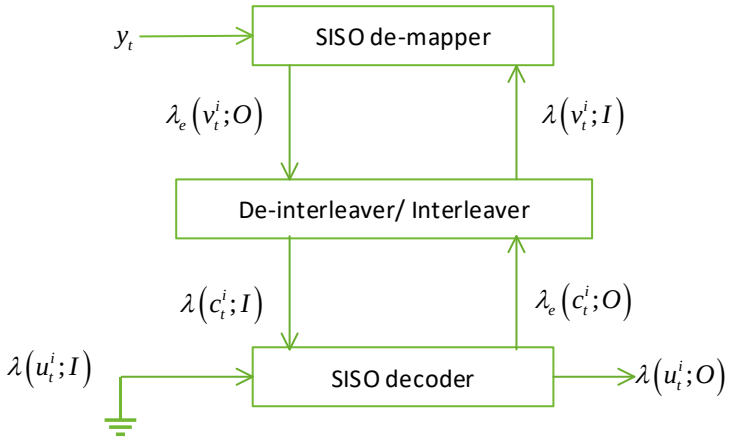


FIGURE 4.6: A simplified diagram for a BICM's soft-decision feedback iterative decoder.

as

$$y_t = \alpha_t e^{j\theta_t} x_t + n_t. \tag{4.11}$$

where $\alpha_t e^{j\theta_t}$ is the complex fading coefficient, and n_t is the complex white Gaussian noise sample with variance $N_0 = E\{|n_t|^2\}$. The extrinsic LLR $\lambda_e(v_t^i;O)$ for the interleaved bit v_t^i can be computed as [8]

$$\lambda_e(v_t^i;O) = \frac{\sum\limits_{x_t \in X_1^i} exp\{log(p(y_t/x_t)) + \sum\limits_{\substack{j=1 \\ j\neq i}} \tilde{v}_t^j(x_t).\lambda(v_t^i;I)\}}{\sum\limits_{x_t \in X_0^i} exp\{(logp(y_t/x_t)) + \sum\limits_{\substack{j=1 \\ j\neq i}} \tilde{v}_t^j(x_t).\lambda(v_t^i;I)\}}. \tag{4.12}$$

where $\tilde{v}_t^j(x_t)$ is the value of the j-th bit of the label corresponding to x_t. In the first iteration, we assume that the interleaved bits v_t^i are equally likely so we initialize the a priori LLR $\lambda_e(v_t^i;I) = 0$. The SISO decoder uses the a priori LLR $\lambda(c_t^i;I)$ of the coded bits, the a priori LLR $\lambda(u_t^i;I)$ of the information bits and the structure of the convolutional code to generate the extrinsic LLR $\lambda_e(v_t^i;O)$ of each coded bit and the extrinsic LLR $\lambda(u_t^i;0)$ of each information bit. After the last iteration, we take the hard decision on the LLRs $\lambda(u_t^i;0)$ of the information bits to get the final decoded bits.

4.4 Punctured convolutional codes for BICM-ID

For a digital communication system, the transmission rate can be adjusted by changing the code rate or the modulation's constellation or both. We call this technique variable rate transmission. Convolutional codes puncturing proposed in [9] to reduce the decoding complexity of high rate codes, allows us to change the code rate. Punctured codes are high rate codes that are generated by selectively removing bits from the convolutional code's output symbol to obtain a higher rate. All punctured codes from the same mother code, can share the same encoder/decoder. Punctured convolutional codes have been used for TCM in [10], [11] and [12], which have simplified the implementation of this latter but leads to a performance degradation compared to Ungerboeck's TCM.

To overcome this problem references [12] and [13] proposed solutions. However, these solutions are optimized for only a certain channel type or a specific code rate. Fig 4.7 shows an example on how to puncture a rate$-1/2$ convolutional code. If N is the coded sequence's length and N' is the number of non-transmitted bits, then the resultant punctured code has a rate $R = N/2(N - N')$.

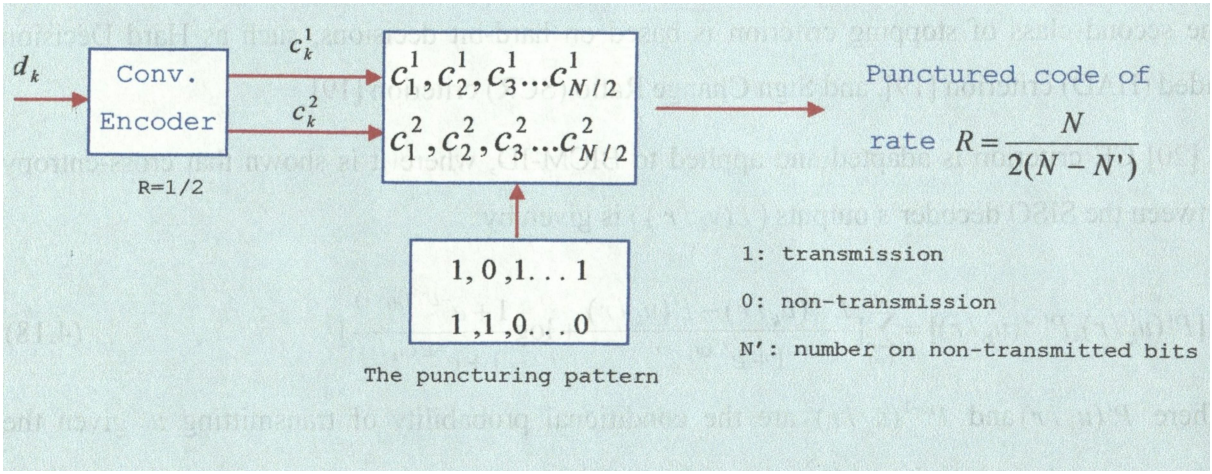


FIGURE 4.7: Puncturing of a rate-1/2 convolutional code [9].

4.5 BICM-ID with reduced complexity

4.5.1 M-BCJR and T-BCJR

Due to iterative decoding, BICM-ID suffers a considerable amount of decoding latency which represents a drawback. To reduce the decoding latency, one can search a method to decrease the complexity of the decoding algorithm i.e. the MAP algorithm. Two well known methods to reduce the complexity of the MAP algorithm are the T-BCJR and the M-BCJR variants [14]. M-BCJR algorithm uses only M most likely states to reduce the BCJR's complexity, where T-BCJR uses a threshold to determine the states to be eliminated (a state with probability smaller than the threshold value will be discarded). At different trellis stages, the T-BCJR uses variable number of states, but the M-BCJR uses a fixed subset of M states. By this T-BCJR has a dynamic behavior where M-BCJR has a static behavior. Because of this the former shows better performance compared to the latter [14].

4.5.2 Stopping criteria for BICM-ID

Decoding complexity is also related to the number of iterations. Stopping criteria, to reduce the number of iterations, used when negligible performance improvement is observed are another approach to reduce the decoding latency. Stopping criteria for turbo codes can be modified and adapted to be suitable for BICM-ID. Turbo decoding stopping criteria can be classified into two classes. The first is based on soft-bit decisions, such as Mutual Information (MJ) criterion [15], Cross-Entropy (CE) criterion [16], Measurement of Reliability (MOR) criterion [17], and Probability of Error (PE) criterion [15]. The second class of stopping criterion is based on hard-bit decisions, such as Hard Decision Aided (HAD) criterion [18], and Sign Change Ratio (SCR) criterion [19].

In [20] CE criterion is adapted and applied to BICM-ID, where it is shown that the cross-entropy between the SISO decoder's outputs $\lambda(u_t^i;O)$ is given by

$$H[P^j(u_t^i/y_t), P^{j-1}(u_t^i/y_t)] = \sum_{t=1}^{N}\left[\frac{\lambda^{j-1}(u_t^i;O) - \lambda^j(u_t^i;O)}{1+e^{\lambda^j(u_t^i;O)}} + log\frac{1+e^{-\lambda^{j-1}(u_t^i;O)}}{1+e^{-\lambda^j(u_t^i;O)}}\right] \tag{4.13}$$

where $P^j(u_t^i/y_t)$ and $P^{j-1}(u_t^i/y_t)$ are the conditional probability of transmitting u_t^i given the received signal y_t at iteration $(j-1)$ and j respectively. $\lambda^{j-1}(u_t^i;O)$ and $\lambda^j(u_t^i;O)$ are the a posteriori LLR of data bit u_t^i at iteration $(j-1)$ and respectively. After calculating the CE, the following stopping condition is applied

$$T(j) < M(j) \tag{4.14}$$

where $T(j)$ is given by

$$T(j) = \sum_{t=1}^{N}\left[\frac{\lambda^{j-1}(u_t^i;O) - \lambda^j(u_t^i;O)}{1+e^{\lambda^j(u_t^i;O)}} + log\frac{1+e^{-\lambda^{j-1}(u_t^i;O)}}{1+e^{-\lambda^j(u_t^i;O)}}\right] \tag{4.15}$$

and $M(j)$ is chosen as

$$M(j) = T(1).10^{-4} \tag{4.16}$$

It was found in [20], that BICM-ID with CE stopping criterion can achieve almost the same performance with less iterations compared with fixed iterative decoding number. The SCR criterion is a simplified version of the CE criterion. It is shown that $T(j)$ in equation (4.20) is directly related to the number of sign changes $c(j)$ in $\lambda^j(u_t^i;O)$ from iteration $(j-1)$ to iteration j [18]. Iterations in the SISO decoder will be stopped when

$$c(j) \leq (0.5 \text{ to } 0.03).N \tag{4.17}$$

is satisfied. Where N is the length of the information bits' sequence. This method requires less complexity compared to CE. An improved SCR approach was proposed in [21]. In HAD criterion, the hard decisions on $\lambda^{j-1}(u_t^i;O)$ at iteration $(j-1)$ are stored to be compared with their counterparts $\lambda^j(u_t^i;O)$ at iteration j. If they agree (i.e. they have the same values) for the entire block, the iterations in the SISO decoder will be stopped at iteration j.

4.6 Conclusion

In this chapter, the theory underlying BICM-ID was covered. Two decision feedback methods were presented. Some techniques used to reduce the BICM-ID's receiver complexity were presented.

Bibliography

[1] Christian. D, "Joint Iterative Demodulation and Decoding for Turbo-Coded PSK Signals on Flat Fading Channels", a Master thesis submitted to the department of Systems and Computer Engineering, University of Carleton. Canada, September, 2000.

[2] C. Berrou, A Glavieux, "Near Optimum Error-Correcting Coding and Decoding", *IEEE Trans. Comm, vol*. 44, pp. 1261-1271, Oct 1996.

[3] X. Li and J. A. Ritcey, "Bit interleaved coded modulation with iterative decoding", *IEEE Commun. Letter*, vol. 1, pp. 169-171, Nov. 1997.

[4] X. Li and J. A Ritcey, ”Trellis-Coded Modulation with Bit Interleaving and Iterative Decoding", *IEEE Journal on Selected Areas in Commun*, vol. 17, pp. 715-724, April. 1999.

[5] X. Li, A. Chindapol and J. A. Ritcey, "Bit interleaved coded modulation with iterative decoding and 8PSK signaling", *IEEE Trans. Commun*. vol. 50, pp. 1250-1257, Aug. 2002.

[6] N. Tran, "Signal mapping design for bit interleaved coded modulation with iterative decoding (BICM-ID)", a Master thesis submitted to the department of Electrical Engineering, University of Saskatchewan. Canada, December, 2004.

[7] E. Zehavi, "8-PSK trellis codes for a Rayleigh fading channel", *IEEE Trans. Comm*, vol.40, pp. 873-883, May 1992.

[8] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "A soft-input soft-output APP module for iterative decoding of concatenated codes," *IEEE Commun. Letters*, vol. 1, pp. 22-24, Jan.1997.

[9] N. S. Muhammad and J. Speidel, "Joint Optimization of Signal Constellation and Bit Labeling for Bit Interleaved Coded Modulation with Iterative Decoding", *IEEE Commun. Letter*. vol. 9, pp. 775-777, Sept. 2005.

[10] J. B. Cain, G. C. Clark, and J. L. Geist, "Punctured convolutional codes of rate (n-A)/n and simplified maximum likelihood decoding", *IEEE Trans. on Inform. Theory*, vol. 25(1), pp. 97-100, Jan. 1974.

[11] J. K. Wolf and E. Zehavi, ”p2 codes: pragmatic trellis cods utilizing punctured convolutional codes", *IEEE Commun. Magazine*. vol. 33, n0. 2, pp. 94-99, Feb. 1995.

[12] E. J. Rossin, D. J. Rowe, and C. Heegard, "Rotationally invariant punctured trellis coding", *Proc. of 995 IEEE Symp. on Inform. Theory*, New York, NY, USA, Sept 1995, IEEE Inf. Theory Soc., pp. 126.

[13] J. Kim and G. J. Pottie, "On punctured trellis-coded modulation", *IEEE Trans. on Inform. Theory*, vol.IT-42, n0. 2, pp. 627-636, March. 1996.

[14] F. Chan and D. Haccoun, "Performance of punctured trellis coded modulation over fading channels", *Proc. of VTC'97*, pp. 339-343, May.1997.

[15] V. Franz and J. B. Anderson, "Concatenated decoding with reduced search BCJR algorithm", *IEEE Journal. on Selected Areas in Commun*, vol. 16, n0. 2, pp. 186-195, Feb.1998.

[16] W. Jiang and D. Li, "Two efficient stopping criteria for iterative decoding", *Proc. of* 1^{rst} *Internal. Conf on Communs and Networking in China*, pp. 1-4, Oct 25-27. 2006.

[17] J. Hagenauer, E. Offer and L. Papke, "Iterative decoding of binary block and convolutional codes", *IEEE Trans. on Inform. Theory*, vol. 42, n0.2, pp. 429-445, Mars. 1996.

[18] L. Fan-Min, W. An-Yeu, "On the stopping criteria of iterative turbo decoding by using decoding threshold", *IEEE Trans. on Signal. Processing*, vol. 55, pp. 5505-5516, Nov. 2007.

[19] R. Y. Shao, S. Lin, and M. P. C. Fossorier "Two simple stopping criteria for turbo decoding", *IEEE Trans. Commun*, vol. 47, pp. 1117-1120, Aug. 1999.

[20] F. Weiwei, L. Jianping, and C. Chaoshi "A variable iterative decoding scheme for various mappings based on BICM-ID", *IEEE Internal. Conf on Software Eng. and Service Sciences (ICSESS)* in Beijing, China, pp. 182 - 186, 16-18 July.2010.

[21] Y. Wu, D. Woerner, and J. Ebel "A simple stopping criteria for turbo decoding", *IEEE Commun. Letter*, vol. 4, pp. 258-260, Aug. 2000.

Chapter 5

Performance analysis of BICM-ID

In this chapter, We'll present the main contribution of this thesis. Firstly, we'll analyze the effect of various parameters on BICM-ID's BER performance. Then, we'll investigate the joint optimization of signal constellation points and their mappings to introduce a new improved symbol mapper/8-ary constellation. Finally, we'll examine different MAP algorithm's variants and we propose a new approximation for the Jacobian logarithm and we'll demonstrate its suitability for MAP decoding in BICM-ID.

5.1 The effects of various parameters on BICM-ID's BER performance

In this section, the BER vs. SNR curves were plotted to show the influence of various parameters, including the iterations' number, the block length and the mapping operation. Also an extensive comparison between various mappings has been carried out. The textbooks' simple convolutional encoder, shown if Fig. 5.1, is used in all the following simulations. It has a rate$= 1/2$, a constraint

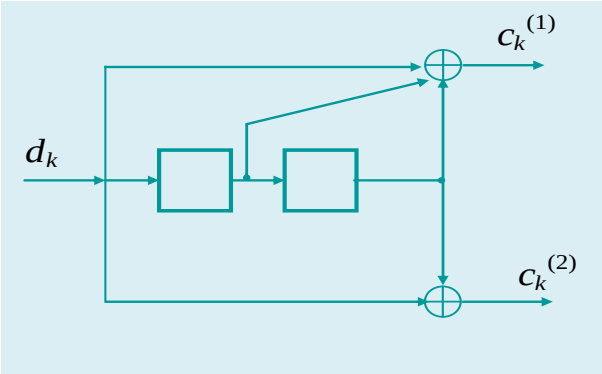


FIGURE 5.1: The NSC convolutional encoder $(7,5)_8$ of rate-1/2 and memory m=2 [1].

length$= 3$, four states and a generator polynomial $G = [g_1, g_2] = [7,5]_8$. It is non-systematic. In convolutional coding, non-systematic codes are preferred over systematic codes [1].

5.1.1 The influence of the iterations' number

In Fig. 5.2 and 5.3, BER vs. SNR curves are shown parameterized by the number of iterations. The simulation parameters, for these two figures, are provided in Table 5.1. The NSC convolutional code shown in Fig. 5.1 is considered.

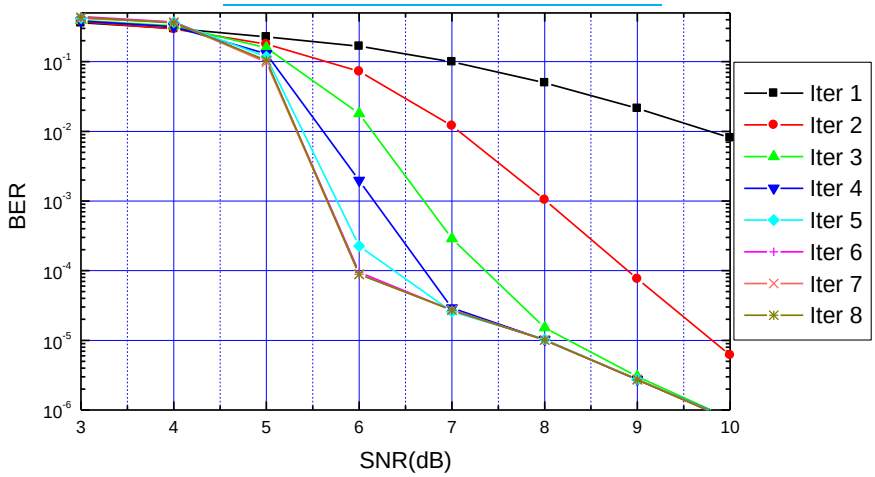
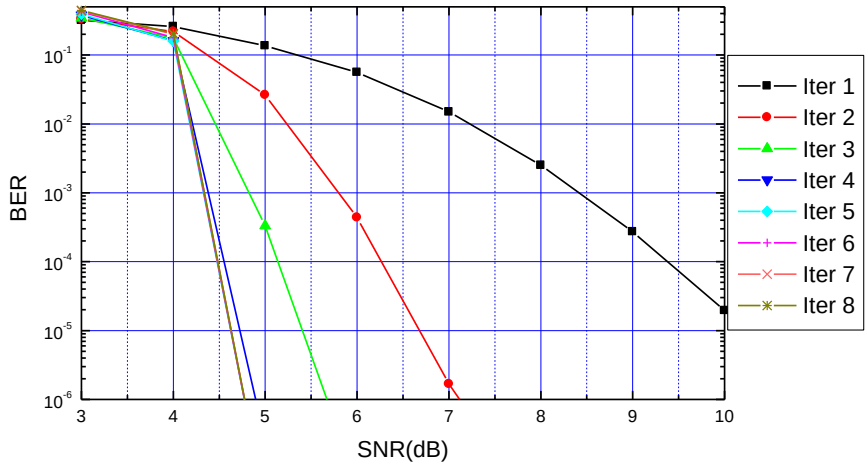
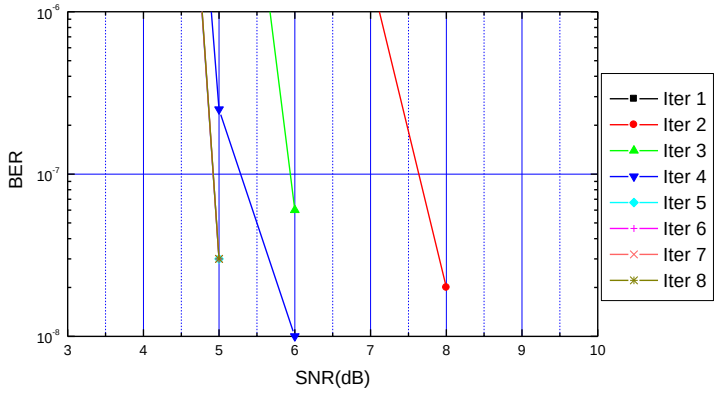


FIGURE 5.2: BER vs. SNR (over Rayleigh fading channel) as parameterized by the number of iterations.



(a)



(b)

FIGURE 5.3: BER vs. SNR (over AWGN channel) as parameterized by the number of iterations. (a) BER=$[10^{-6}, 5.10^{-1}]$. (b) BER=$[10^{-8}, 10^{-6}]$.

TABLE 5.1: Simulation parameters

Figure #	Channel	Code generator	Code rate	Mapping	Block length	Iteration number
5.2	Rayleigh	$(7,5)_8$	$1/2$	SSP	4000	1 to 8
5.3	AWGN	$(7,5)_8$	$1/2$	SSP	4000	1 to 8

Fig. 5.2 and 5.3 show that, for both Rayleigh and AWGN channels, the BER decreases as the number of iterations increases and tends to converge. However, when the number of iterations increases, the time required for the decoding process increases. Also, we can observe that when we pass from iteration 1 to iteration 2, the BER significantly deceases. This is due to the fact that, by the information transferred back from the SISO decoder to the SISO demapper, this later will have further knowledge about coded bits which directly influence the decoding process. Through iterations, the BICM-ID's BER decreases, however the decreasing rate decreases. This can be explained as: after some iterations, feedback will not bring a significant information about coded bits compared to it when we pass from iteration 1 to iteration 2. The system's convergence behavior over AWGN channel is very different from it over Rayleigh fading channel. For AWGN channel, we have a remarkable divergence between iterations 1, 2, 3 and 4. Iterations 5, 6, 7 and 8 have almost the same BER performance. To further study the behavior of iterations 5, 6, 7 and 8 we run simulations for the BER's interval $[10^{-8}, 10^{-6}]$. The result is given in Fig. 5.3(b). For Rayleigh fading channel, iterations 6, 7 and 8 have almost the same BER performance and they converge with iterations 4 and 5 when SNR is greater than 7.

5.1.2 The influence of the block length

In Fig. 5.4 and 5.5, BER vs. SNR curves are shown parameterized by the block length. The simulation parameters, for these two figures, are provided in Table 5.2. These figures show that, for the first iteration the block length has almost no influence on the BER. However, starting at the 2^{nd} iteration, a large block length is desirable to obtain a good performance. We can observe also that, over both Rayleigh fading and AWGN channels more than 0.8 dB gain can be provided at a BER equal to 10^{-4}, when the block length is increased from 500 to 2000 bits. However, a small loss is observed when the block length is reduced from 4000 to 2000 bits.

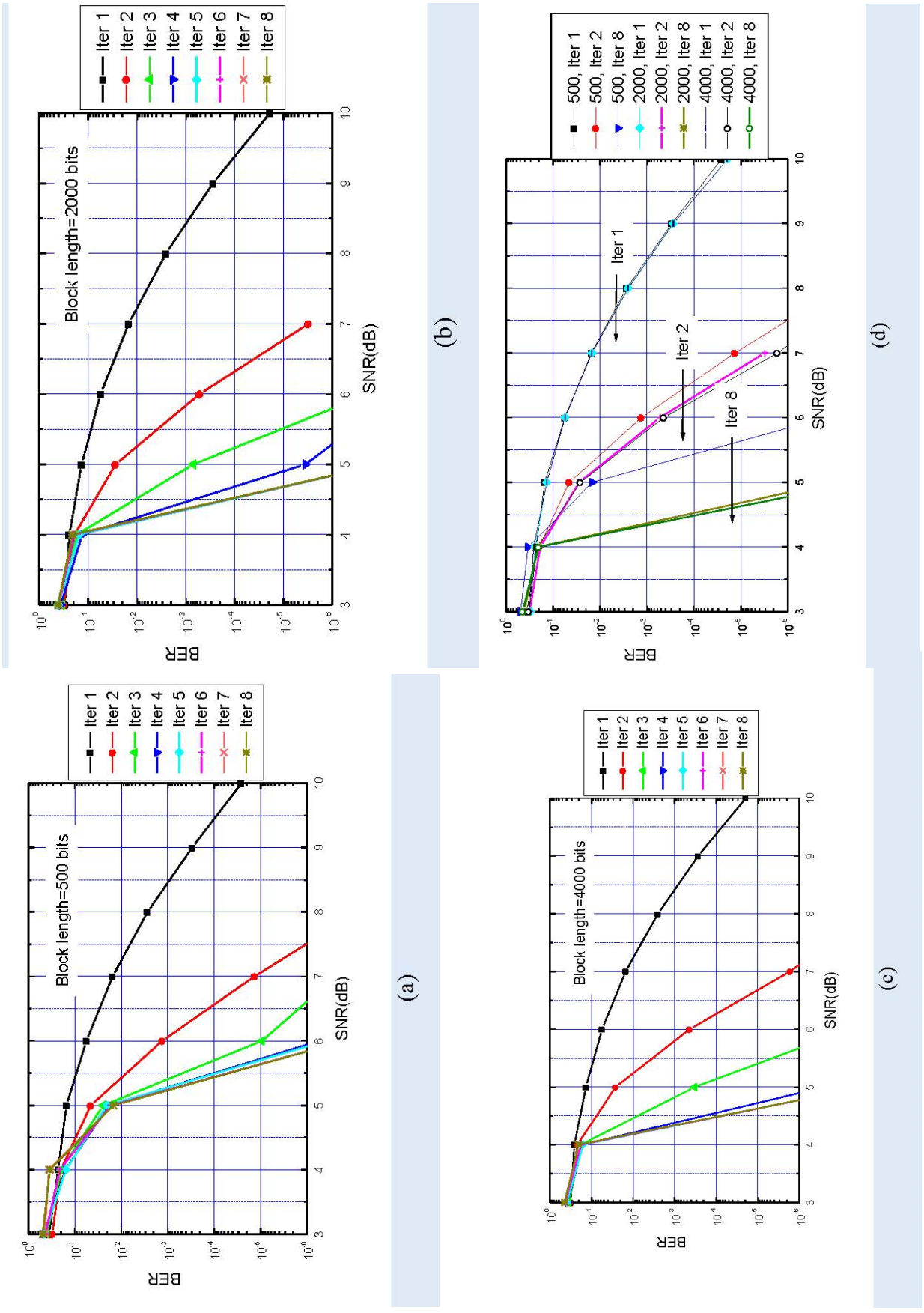


FIGURE 5.4: BER vs. SNR (over Rayleigh fading channel) as parameterized by the block length. (a) Block length=500 bits. (b) Block length=2000 bits. (c) Block length=4000 bits. (d) Comparison.

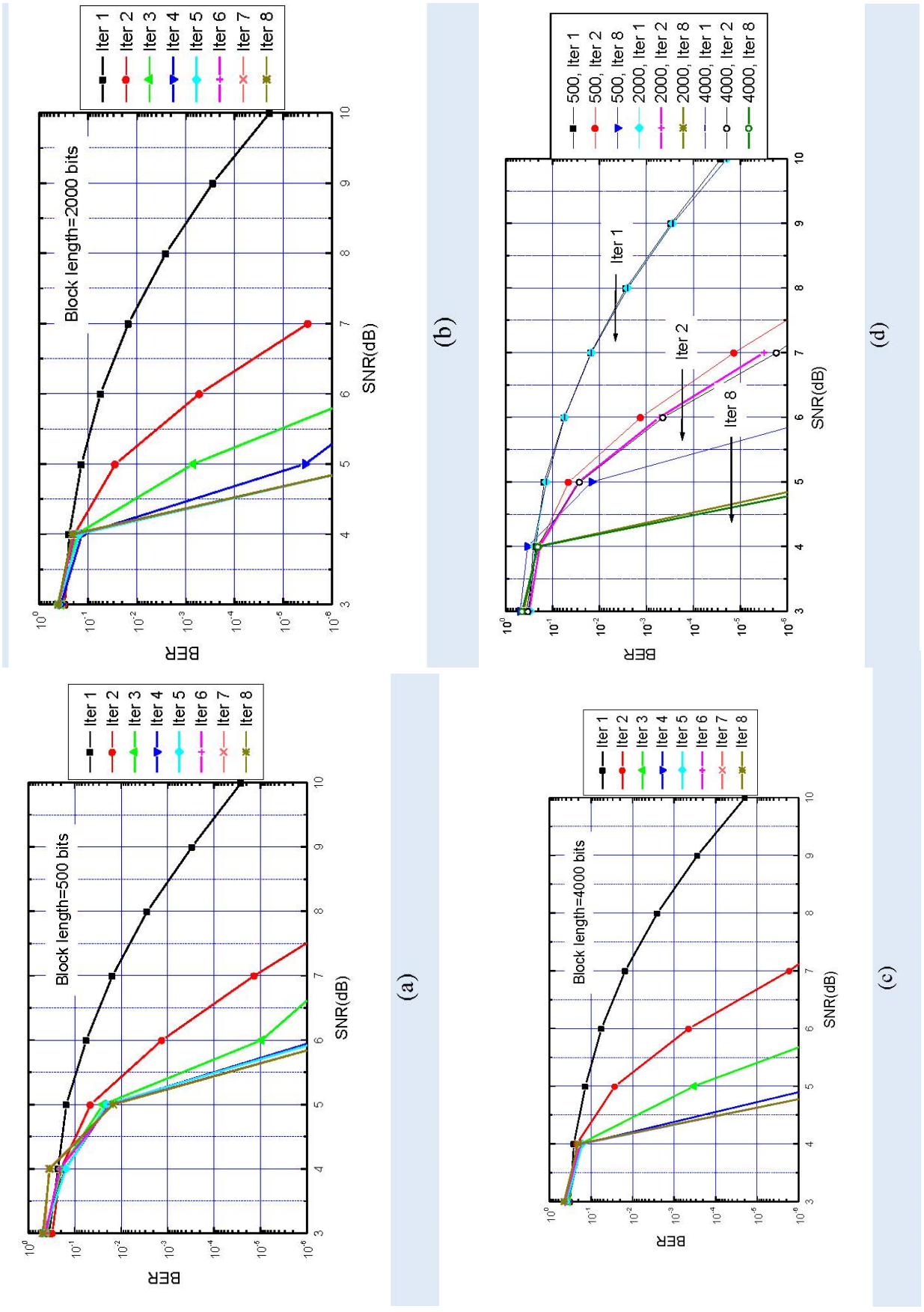


FIGURE 5.5: BER vs. SNR (over AWGN channel) as parameterized by the block length. (a) Block length=500 bits. (b) Block length=2000 bits. (c) Block length=4000 bits. (d) Comparison.

TABLE 5.2: Simulation parameters.

Figure#	Channel	Code generator	Code rate	Mapping	Block length	Iteration number
5.4	Rayleigh	$(7,5)_8$	$1/2$	SSP	500, 2000, and 4000	1,2,4 and 8
5.5	AWGN	$(7,5)_8$	$1/2$	SSP	500, 2000, and 4000	1,2,4 and 8

5.1.3 The influence of signal mapping

In Fig. 5.6 to 5.15, BER vs. SNR curves are shown parameterized by the signal mapping. Five different signal mappings namely Gray, SP, SSP, MIXED [2] and MSEW [3] are considered. Fig. 5.16 and Fig. 5.17 compare the performance of these mappings over both Rayleigh and AWGN channels. The simulation parameters, for all these figures, are provided in Table 5.3.These figures show that:

- Gray mapping, for the 1^{rst} iteration, gives the best BER performance. However, it doesn't bring any performance gain through iterations. This is because its Harmonic Mean of the Minimum Squared Euclidian Distance (HMMSED) is almost unchanged after iterations (Table 5.4).
- Contrary, SSP mapping brings a very significant gain through iterations due to the large improvement in its HMMSED (Table 5.4).
- SP mapping has a performance between those of Gray and SSP.
- MIXED mapping has better performance than SP for the first iteration but its performance is poorer than it of SP through iterations.
- MSEW mapping shows almost the same performance as SSP over AWGN channel (Fig. 5.17) and it is slightly better than SSP over Rayleigh fading channel (Fig. 5.16).

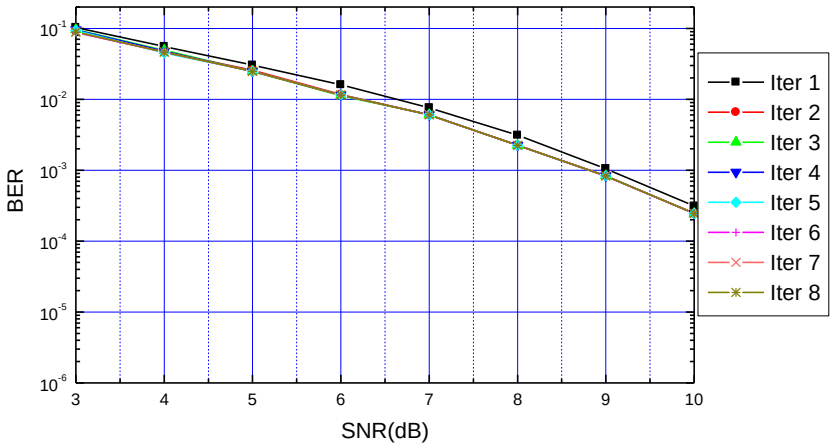


FIGURE 5.6: BER vs. SNR (over Rayleigh fading channel) using Gray mapping.

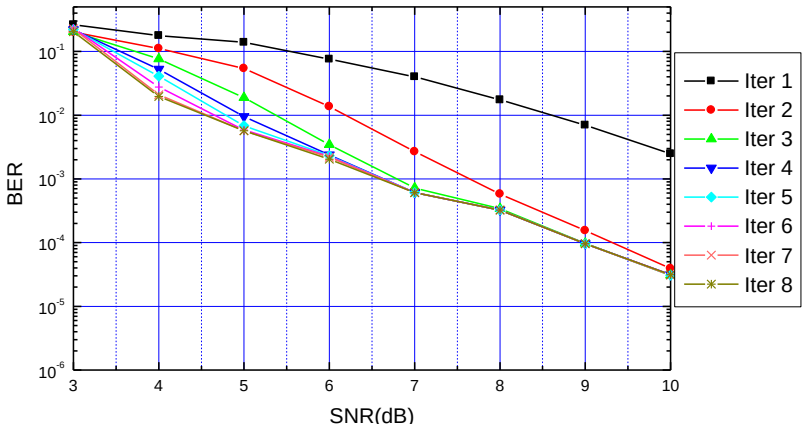


FIGURE 5.7: BER vs. SNR (over Rayleigh fading channel) using SP mapping.

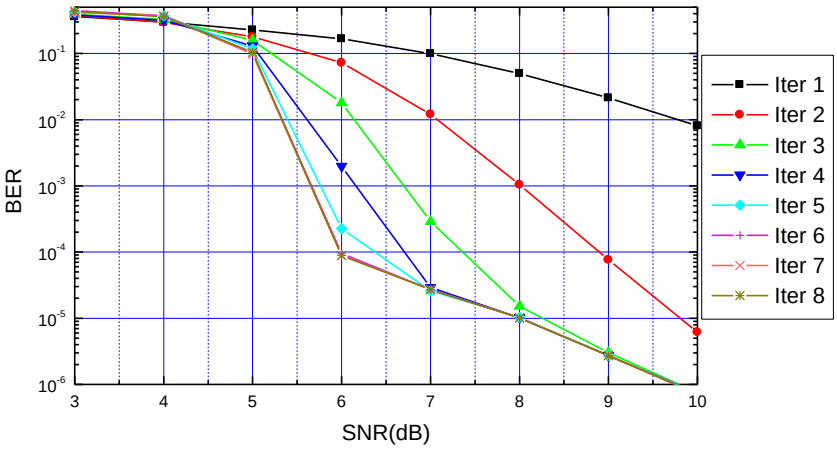


FIGURE 5.8: BER vs. SNR (over Rayleigh fading channel) using SSP mapping.

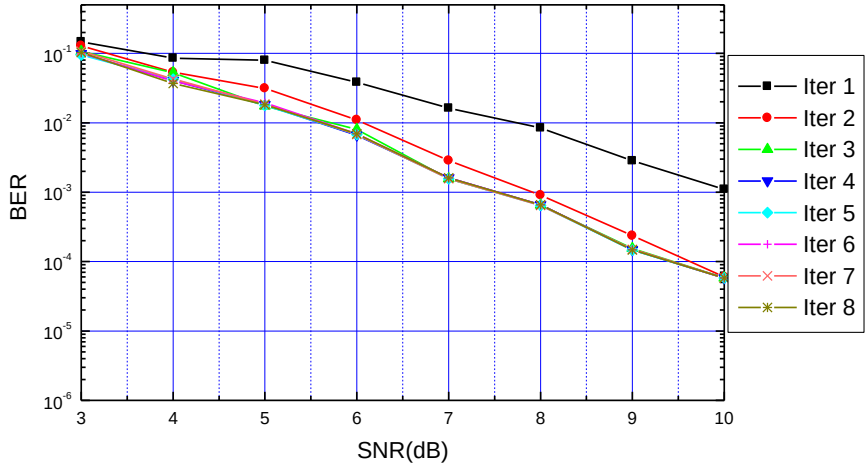


FIGURE 5.9: BER vs. SNR (over Rayleigh fading channel) using MIXED mapping.

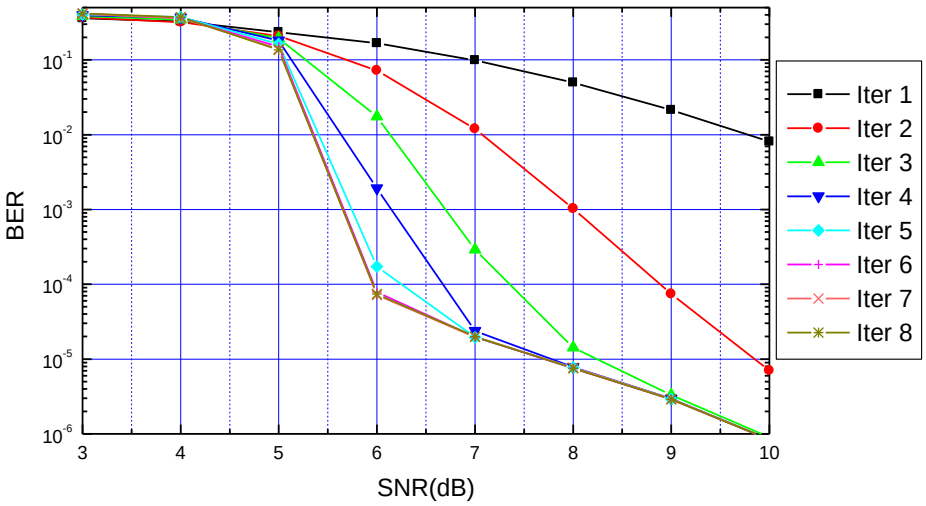


FIGURE 5.10: BER vs. SNR (over Rayleigh fading channel) using MSEW mapping.

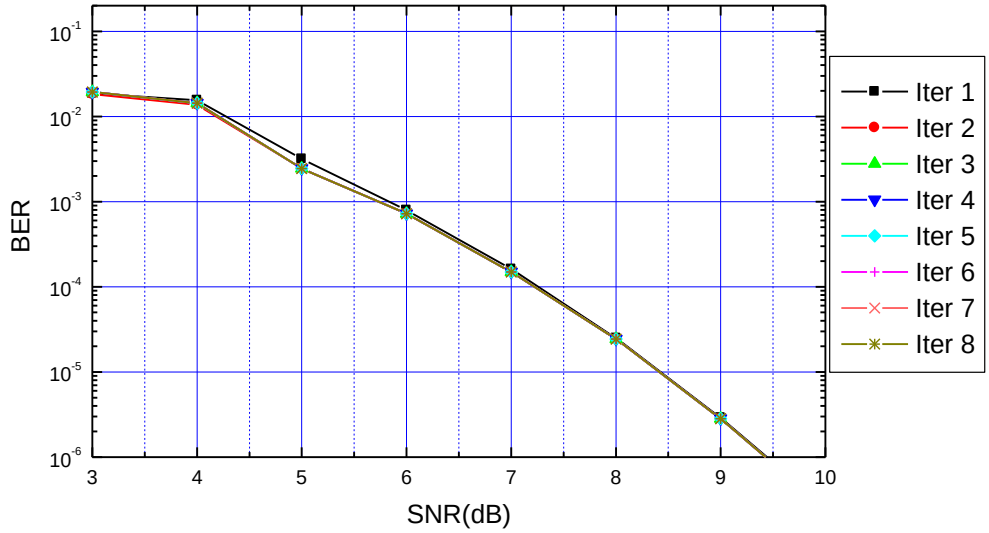


FIGURE 5.11: BER vs. SNR (over AWGN channel) using Gray mapping.

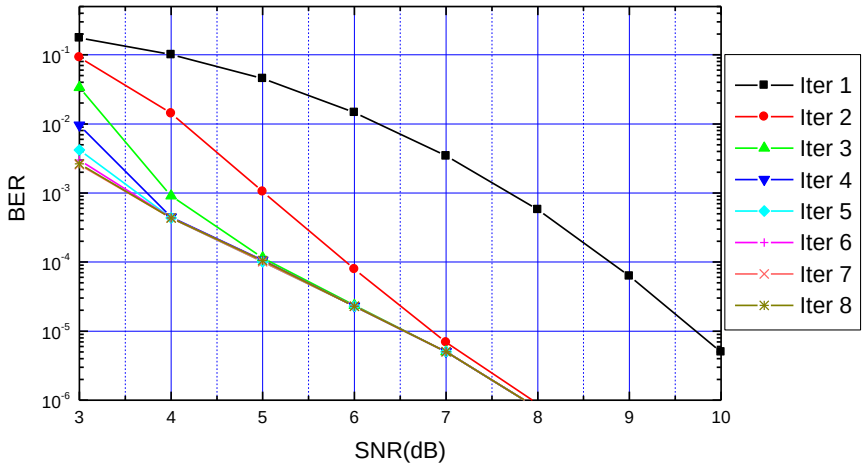


FIGURE 5.12: BER vs. SNR (over AWGN channel) using SP mapping.

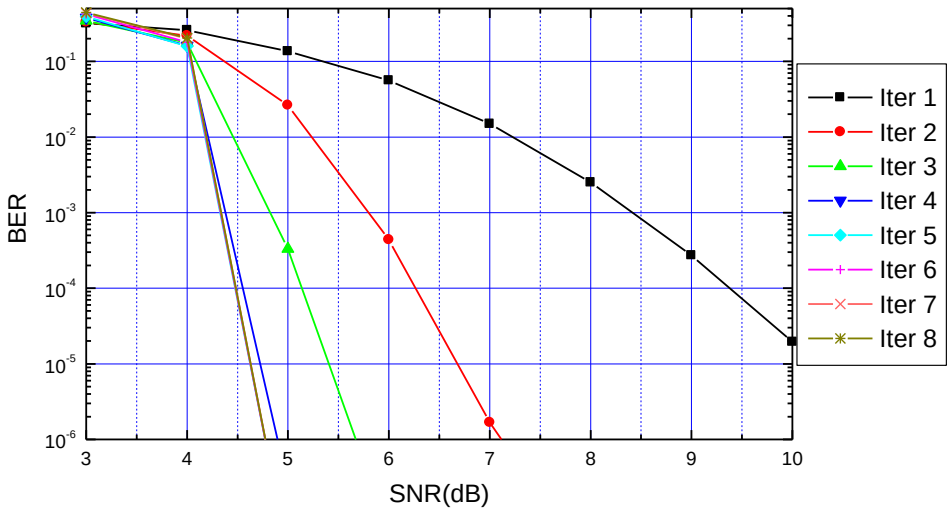


FIGURE 5.13: BER vs. SNR (over AWGN channel) using SSP mapping.

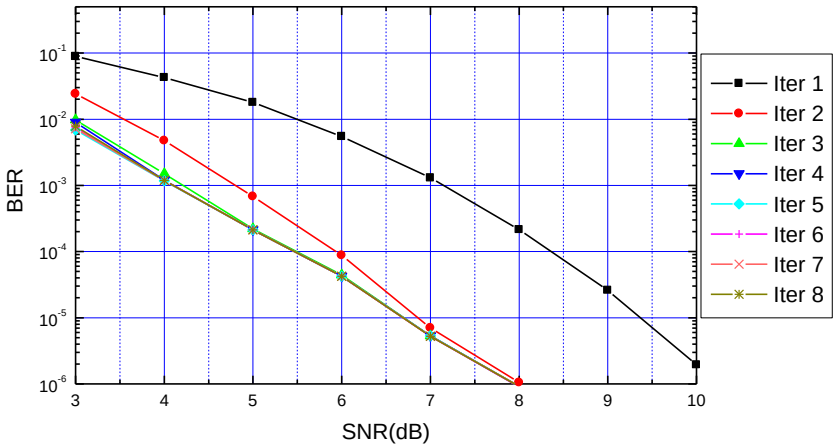


FIGURE 5.14: BER vs. SNR (over AWGN channel) using MIXED mapping.

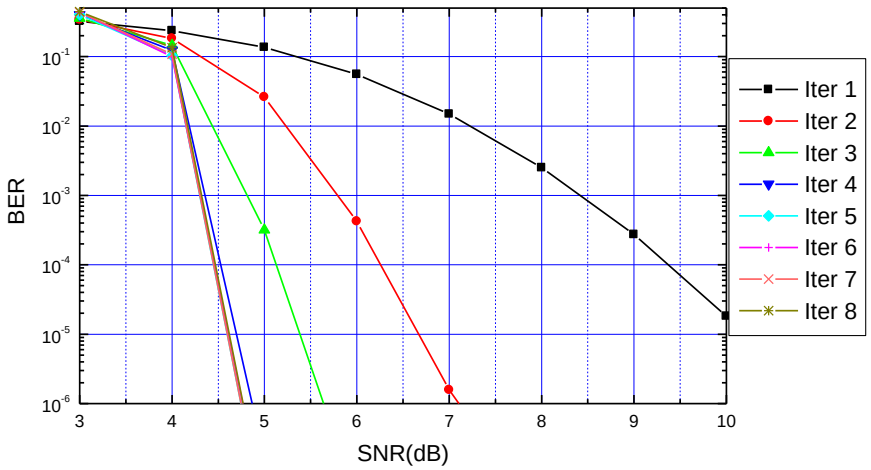


FIGURE 5.15: BER vs. SNR (over AWGN channel) using MSEW mapping.

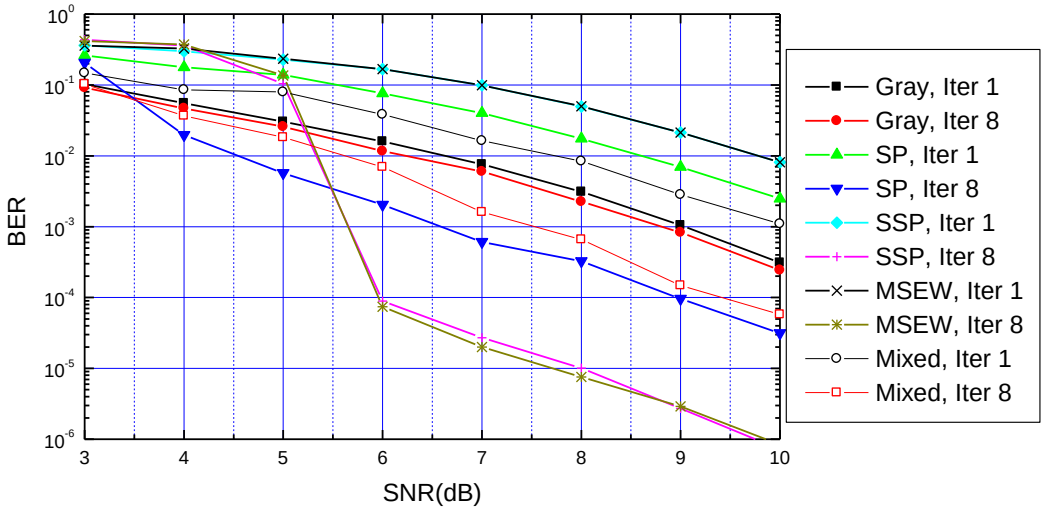


FIGURE 5.16: BER vs. SNR (over Rayleigh fading channel) for all mappings.

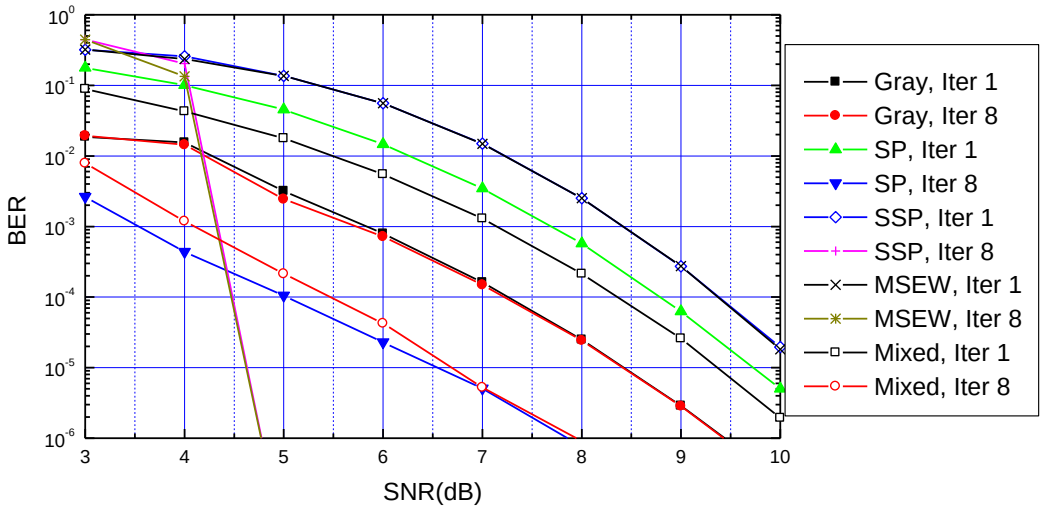


FIGURE 5.17: BER vs. SNR (over AWGN channel) for all mappings.

TABLE 5.3: Simulation parameters.

Figure#	Channel	Code generator	Code rate	Mapping	Block length	Iteration number
5.6	Rayleigh	$(7,5)_8$	$1/2$	Gray	4000	1 to 8
5.7	Rayleigh	$(7,5)_8$	$1/2$	SP	4000	1 to 8
5.8	Rayleigh	$(7,5)_8$	$1/2$	SSP	4000	1 to 8
5.9	Rayleigh	$(7,5)_8$	$1/2$	MIXED	4000	1 to 8
5.10	Rayleigh	$(7,5)_8$	$1/2$	MSEW	4000	1 to 8
5.11	AWGN	$(7,5)_8$	$1/2$	Gray	4000	1 to 8
5.12	AWGN	$(7,5)_8$	$1/2$	SP	4000	1 to 8
5.13	AWGN	$(7,5)_8$	$1/2$	SSP	4000	1 to 8
5.14	AWGN	$(7,5)_8$	$1/2$	MIXED	4000	1 to 8
5.15	AWGN	$(7,5)_8$	$1/2$	MSEW	4000	1 to 8
5.16	Rayleigh	$(7,5)_8$	$1/2$	Gray, SP, SSP, MIXED and MSEW	4000	1, 2 and 8
5.17	AWGN	$(7,5)_8$	$1/2$	Gray, SP, SSP, MIXED and MSEW	4000	1, 2 and 8

TABLE 5.4: HMMSED before and after feedback (see equation (17) and table II in reference [4]).

Mapping	HMMSED before	HMMSED after
Gray	0.7664	0.8093
SP	0.6640	1.2209
SSP	0.5858	2.8766

5.2 Joint Optimization of signal constellation points and their mappings

5.2.1 Seven different 8-ary constellations for BICM-ID

Constellations with different symbols' positions have different Euclidian distances. Therefore, they have different performances. In this section, seven different $8-$ary constellations are used. These constellations are: Conventional, Rectangular, Cross $8-$ary, (1,7)(see reference [22] of the current chapter), Optimum (it is optimum in the sense that it performs optimally, at high SNR, for AWGN channel [5]), (4,4), and Triangular. Fig. 5.18 shows these constellations.

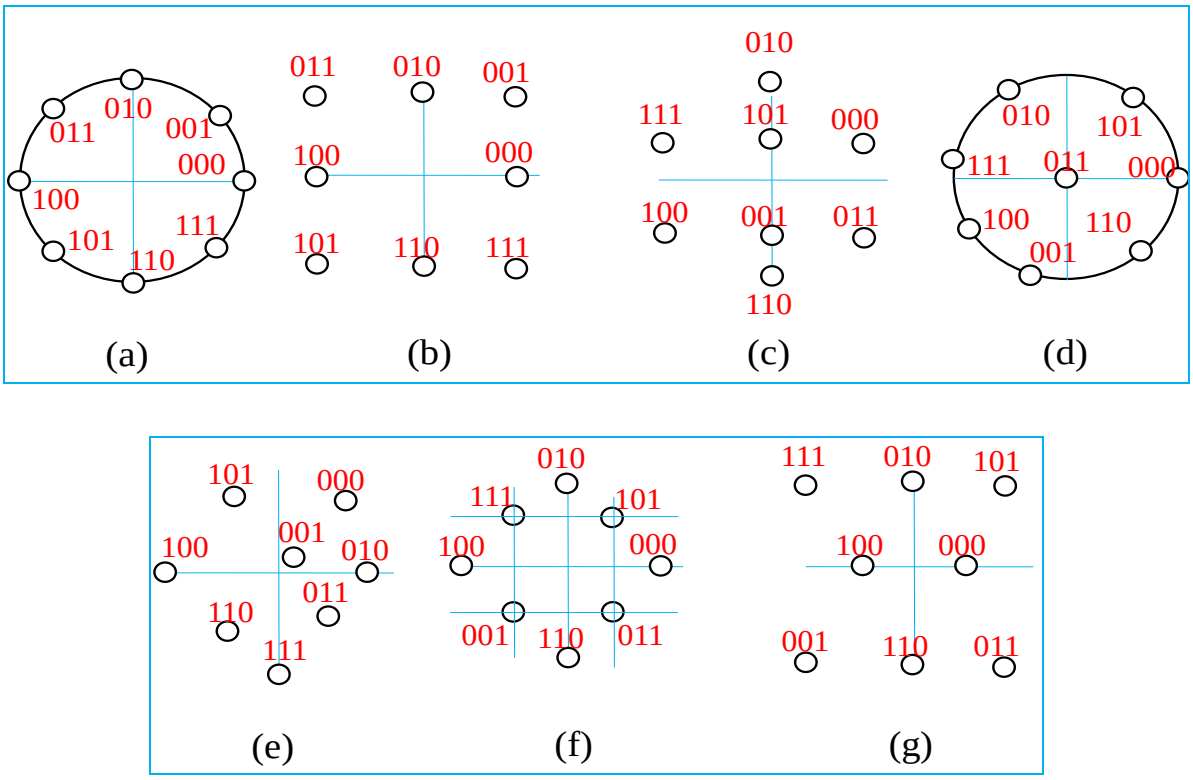


FIGURE 5.18: Different constellations with their mappings: (a). Conventional 8-PSK, (b). Rectangular, (c). Cross 8-ary, (d). (1,7), (e). Optimum,(f). (4,4) and (g). Triangular.

For Conventional and Rectangular constellations, SSP mapping is used, because in [6], [7] we

found that it is very suitable for these constellations. For other constellations the mappings mentioned in Fig. 5.18 are used.

As this investigation concentrates on signal constellations and their mappings, a $4-$states, rate$-1/2$ NSC convolutional encoder whose generator polynomials are $G = [g_1, g_2] = [7, 5]_8$ is considered (Fig. 5.1). A random bit-interleaver is used. For each BER point, we simulate up to 10^{+8} information bits, and we count at least 100 erroneous bit. The Max-log-Map version of the BCJR algorithm is used to decode the NSC code. A Rayleigh fading channel model is considered.
Fig. 5.19 to 5.25 give the BER curves for each constellation for iterations 1, 4 and 8. These results show that the system's performance can be improved through iterations except for the optimum constellation when we pass from iteration 4 to iteration 8.

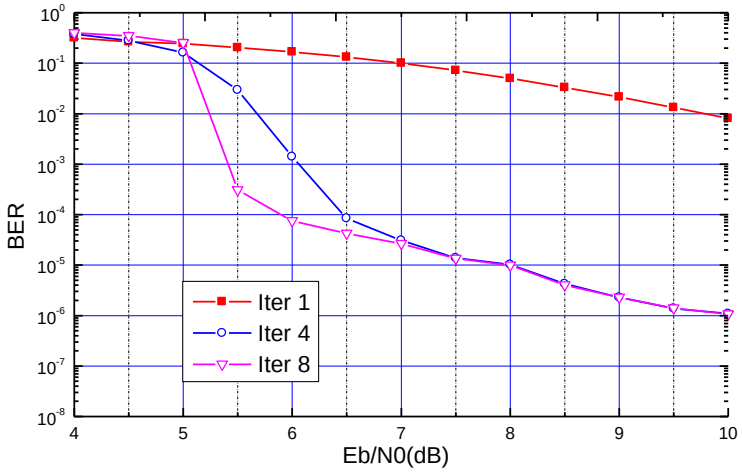


FIGURE 5.19: BER performance of BICM-ID with Conventional constellation for iterations 1, 4 and 8.

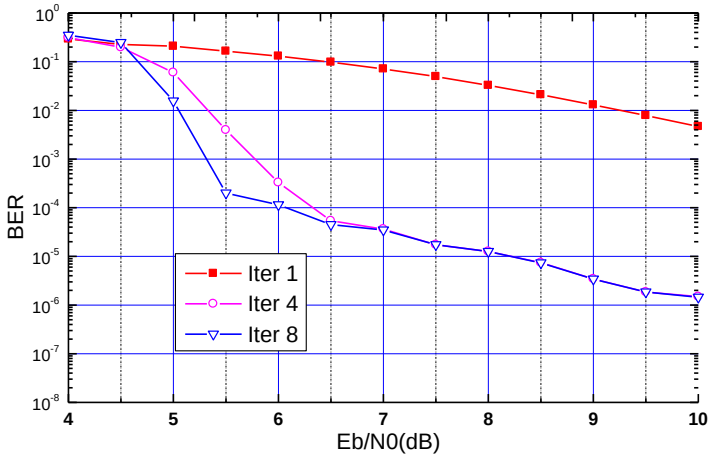


FIGURE 5.20: BER performance of BICM-ID with Rectangular constellation for iterations 1, 4 and 8.

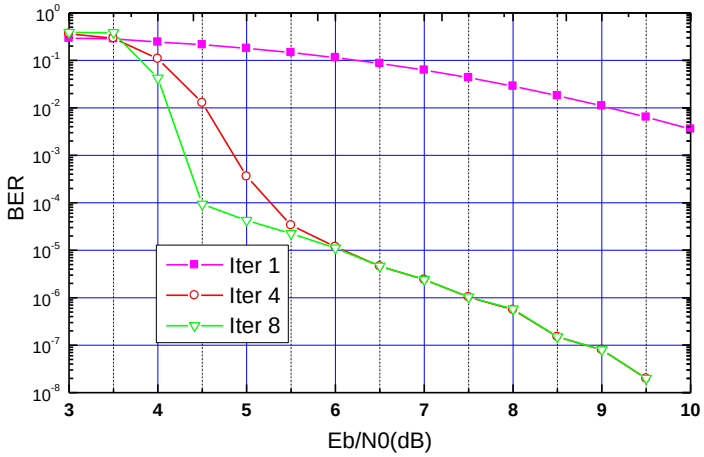


FIGURE 5.21: BER performance of BICM-ID with Cross 8-ary constellation for iterations 1, 4 and 8.

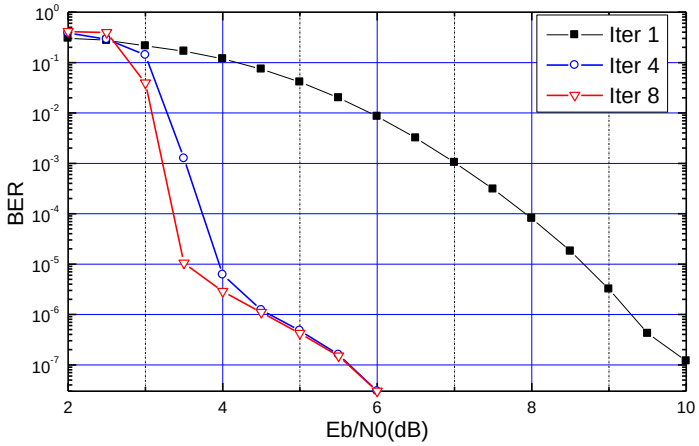


FIGURE 5.22: BER performance of BICM-ID with (1,7) constellation for iterations 1, 4 and 8.

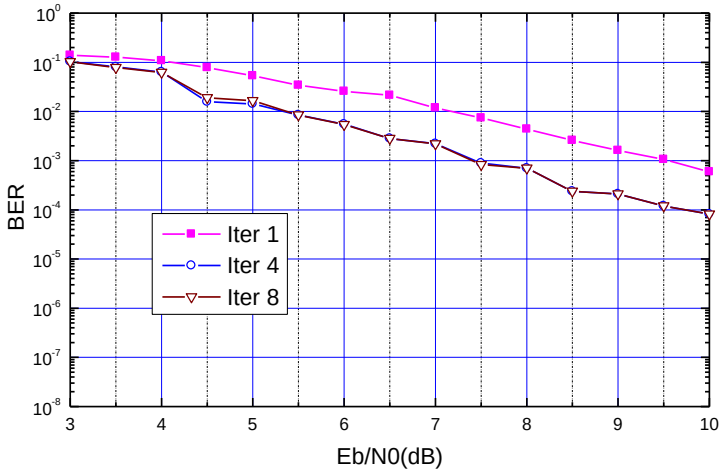


FIGURE 5.23: BER performance of BICM-ID with Optimum constellation for iterations 1, 4 and 8.

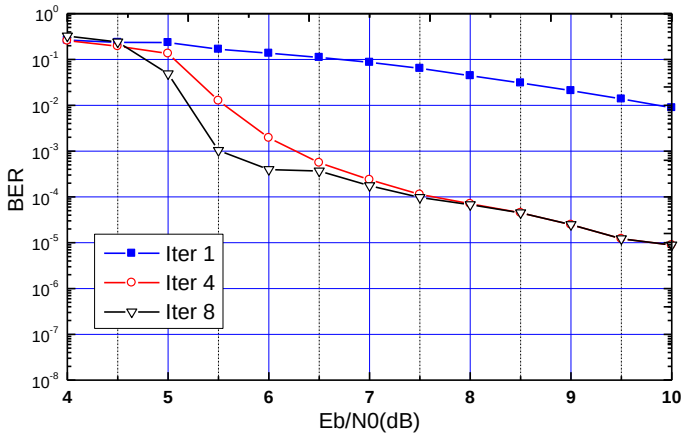


FIGURE 5.24: BER performance of BICM-ID with (4,4) constellation for iterations 1, 4 and 8.

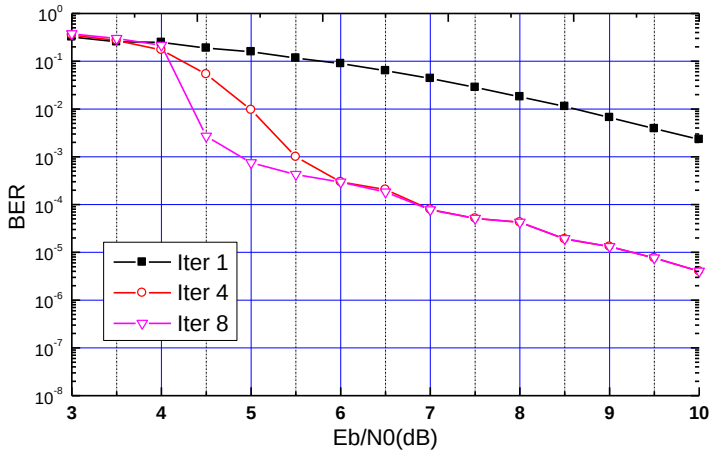


FIGURE 5.25: BER performance of BICM-ID with Triangular constellation for iterations 1, 4 and 8.

Fig. 5.26, explicitly compares the BER performances of all constellations after 8 iterations. This figure shows that after 8 iterations for SNRs greater than 2.7 dB, (1,7) constellation outperforms with a significant gain all other constellations. For example, at BER level of 10^{-6}, the SNR gain provided by (1,7) constellation over Cross 8-ary constellation is about 3 dB, and it is about 5.5 dB over Conventional constellation. We note also that with a SNR=3.5 dB (respectively 6 dB) we can reach a BER level equal to 10^{-5} (respectively 3.10^{-7}).
We conclude that (1,7) constellation associated with its mapping gives better performance than all other studied constellations.

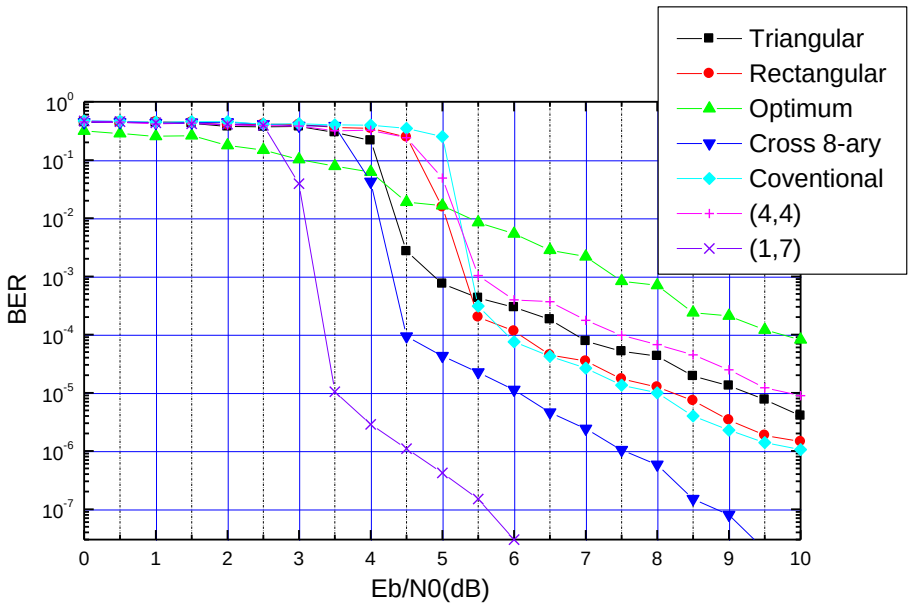


FIGURE 5.26: BER performance of BICM-ID for all constellations after 8 iterations.

5.2.2 A new improved symbol mapper/8-ary constellation for BICM-ID

In this section we propose a new combination symbol mapper/8-ary constellation, which is a joint optimization of an 8-ary signal constellation and its symbol mapping operation, to improve the performance of BICM-ID.
The basic idea was to use the so called (1,7) constellation instead of the conventional 8-PSK constellation and to choose the most suitable mapping for it.
A comparative study between the combinations most suitable mapping /(1,7) constellation and SSP mapping/conventional 8-PSK constellation has been carried out. Simulation results showed that the 1^{rst} combination significantly outperforms the 2^{nd} combination and with only 4 iterations, it gives better performance than the 2^{nd} combination with 8 iterations. A gain of 4 dB is given by iteration 4 of the 1^{rst} combination compared to iteration 8 of the 2^{nd} combination at a BER level equal to 10^{-5}, and it (iteration 4 of the 1^{rst} combination) can attain a BER equal to 10^{-7} for, only, a SNR$= 5.6$ dB.
(1,7) Constellation and the proposed mappings
For a given number of bits (the value of m) that should be transmitted per symbol, the signal constellation and the mapping method must be specified. The (1,7) constellation's symbols set is given by

$$s_L = exp(j2\pi(L-1)/7)) \text{ for } L = 1, 2, ..., 7 \text{ and } s_L = 0 \text{ for } L = 8 \tag{5.1}$$

These symbols are shown in Fig. 5.27(a) accompanied with the symbols of conventional 8-PSK constellation in Fig. 5.27(b). In table 5.5, we show the seven different mapping methods (called MAP1, MAP2, ..., MAP7) for (1,7) constellation and the SSP mapping used for conventional 8-PSK constellation.

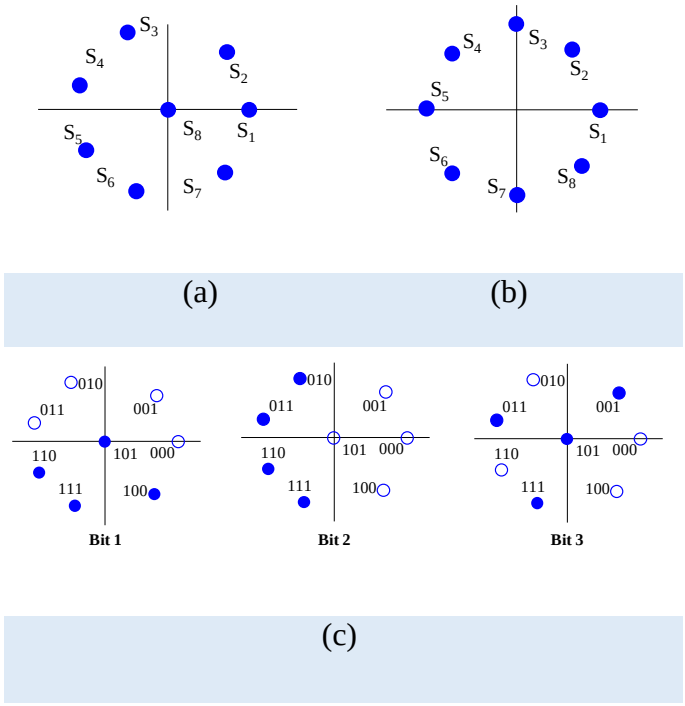


FIGURE 5.27: The two used constellations: (a) and (b) their symbols' positions. (c). Sub-sets partition for (1,7) constellation (MAP3 is considered).

TABLE 5.5: (1,7) constellation and the seven used mappings (MAP1, MAP2,..., MAP7).

(1,7) constellation	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8
MAP1	000	001	011	010	110	111	101	100
MAP2	000	001	010	011	100	101	110	111
MAP3	000	001	010	011	110	111	100	101
MAP4	000	101	010	011	100	001	110	011
MAP5	001	010	111	100	011	010	110	000
MAP6	001	111	100	101	011	110	010	000
MAP7	111	101	100	001	011	010	110	000
8-PSK constellation	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8
SSP	000	101	010	111	100	001	110	011

In practice, to compute the a posteriori LLRs of the interleaved bits (de-mapping operation), we need to divide the constellation set Ψ into two sub-sets Ψ_k^b for each position k where $b \in \{0, 1\}$ (the bit's binary value). For a MAP3 mapped (1,7) constellation (Fig. 5.27(c)), we have the following constellation's sub-sets

$$\Psi_1^0 = \{s_1, s_2, s_3, s_4\}, \Psi_1^1 = \{s_5, s_6, s_7, s_8\} \tag{5.2}$$

$$\Psi_2^0 = \{s_1, s_2, s_8, s_7\}, \Psi_2^1 = \{s_3, s_4, s_5, s_6\} \tag{5.3}$$

$$\Psi_3^0 = \{s_1, s_3, s_5, s_7\}, \Psi_3^1 = \{s_2, s_4, s_6, s_8\} \tag{5.4}$$

In Fig. 5.27(c), a filled (respectively an unfilled) circle represents a symbol for which the binary label has the value $b = 1$(respectively $b = 0$) in the k^{th} bit's position.For simulation, the same parameters used in section 5.2.1 are considered in this section. Fig. 5.28 explicitly compares the BER performances of these seven mappings after 8 iterations.
These results show that MAP3 is the most suitable (among all used mappings) mapping for (1,7) constellation. For example, with MAP3 we can achieve a BER equal to 5.10^{-8} for 6 dB (Fig. 5.28). From Fig. 5.28, we can also observe that to achieve better performance we can use MAP4 for SNRs smaller than 3.2 dB and MAP3 for SNRs greater than this value (we call this technique *adaptive symbol mapping*).

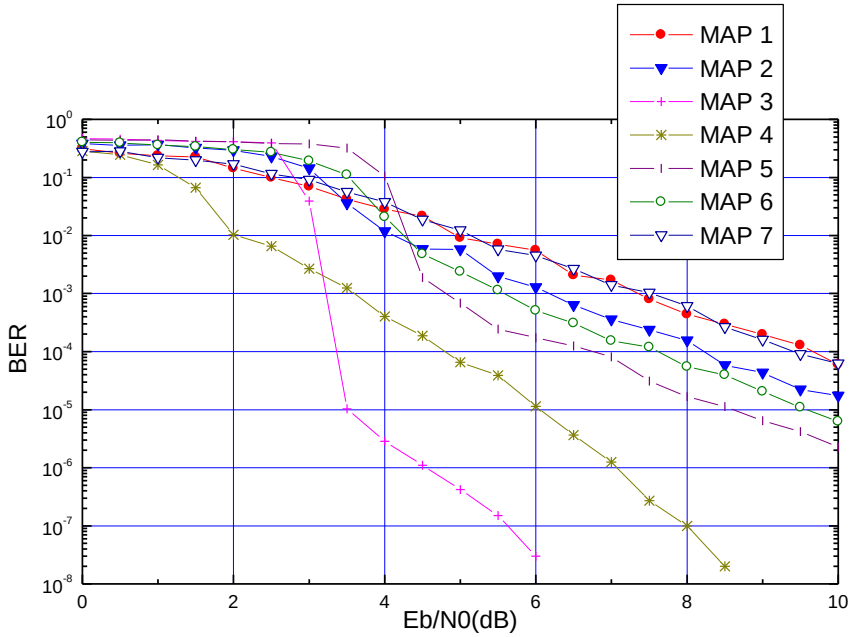


FIGURE 5.28: The BICM-ID's BER performance for all used mappings (for (1,7) constellation) after 8 iterations.

A comparison between the combinations MAP3 mapping/(1,7) constellation and SSP mapping/ Conventional 8-PSK constellation were carried out and its result is given in Fig. 5.29. This figure shows that the first combination gives a very significant gain compared to the second combination and this is from the 1^{rst} to the 8^{th} iteration.

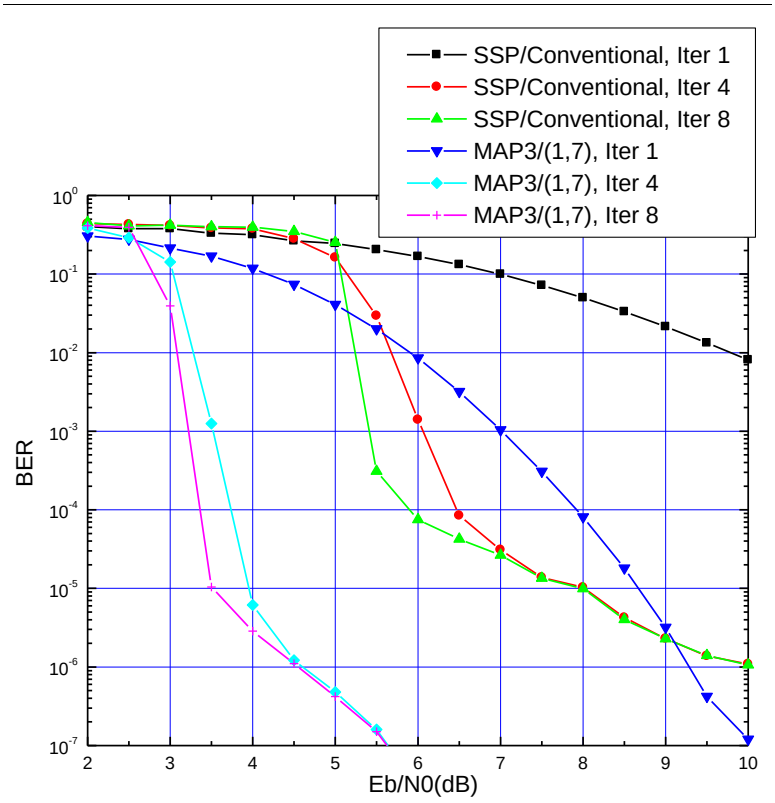


FIGURE 5.29: A comparison between the performances of the system using the two combinations.

It is necessary to note that the 1^{rst} combination, with only 4 iterations, gives better performance than the 2^{nd} combination with 8 iterations. For example, a gain of 4 dB (4.5 dB respectively) is given by iteration 4 (iteration 8 respectively) of the 1^{rst} combination compared to iterations 4 and 8 of the 2^{nd} combination at a BER level equal to 10^{-5}, and they (iterations 4 and 8 of the first combination) can attain a BER equal to 10^{-7} for a SNR$= 5.6$dB.
We conclude that the new combination *MAP3/(1,7) constellation* significantly outperforms the well-known combination *SSP mapping/ conventional 8-PSK constellation*. Fig. 5.30 to 5.36 give the BER vs. SNR curves for each mapping for (1,7) constellation for iterations 1 until 8.

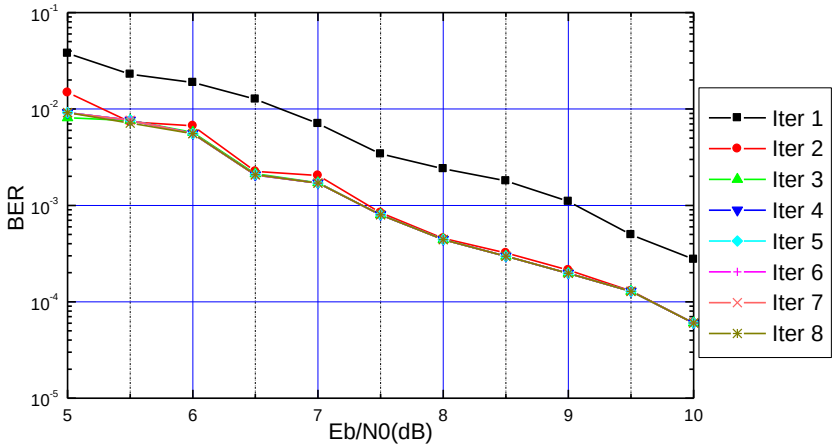


FIGURE 5.30: The BICM-ID's BER performance with MAP1/ (1,7) constellation for iterations 1-8.

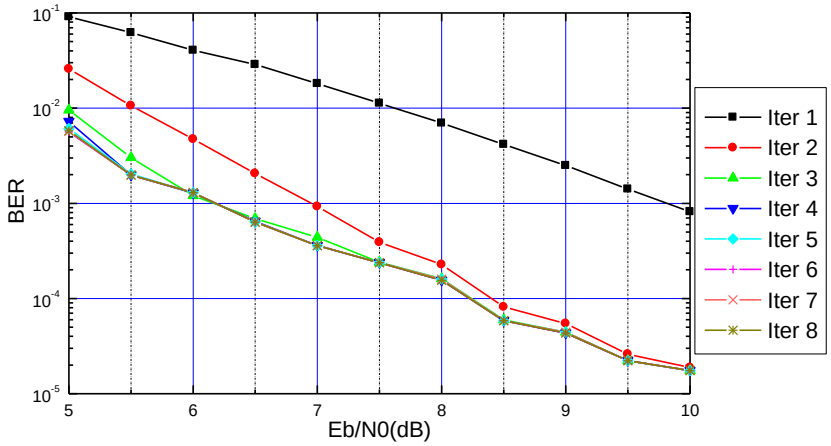


FIGURE 5.31: The BICM-ID's BER performance with MAP2/ (1,7) constellation for iterations 1-8.

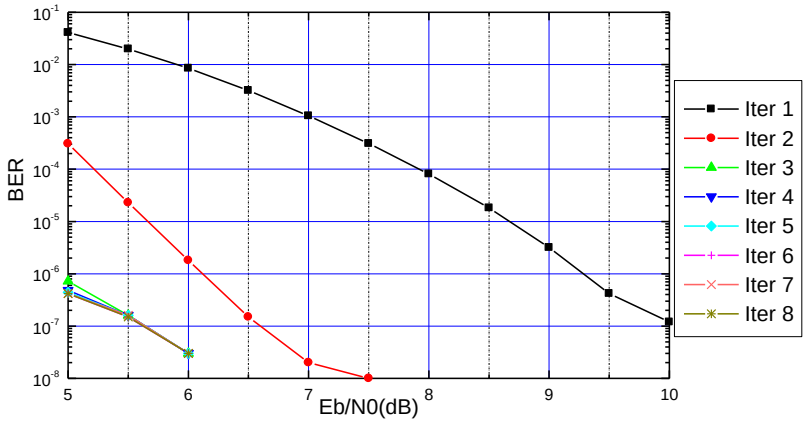


FIGURE 5.32: The BICM-ID's BER performance with MAP3/ (1,7) constellation for iterations 1- 8.

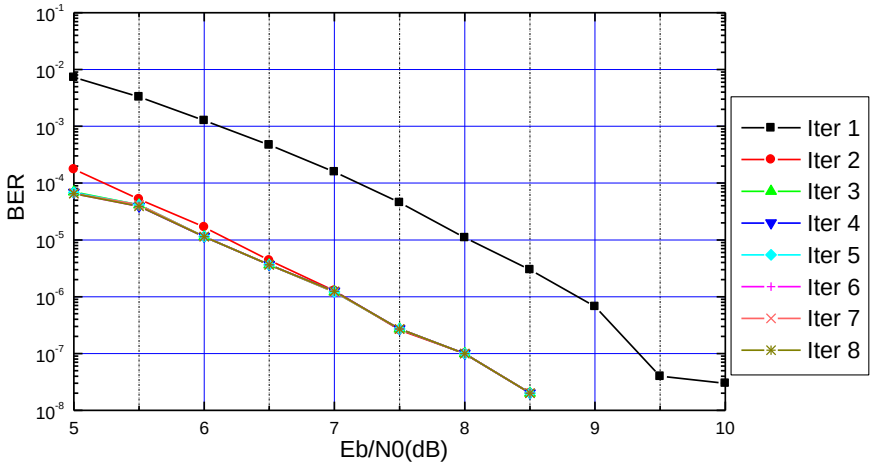


FIGURE 5.33: The BICM-ID's BER performance with MAP4/ (1,7) constellation for iterations 1-8.

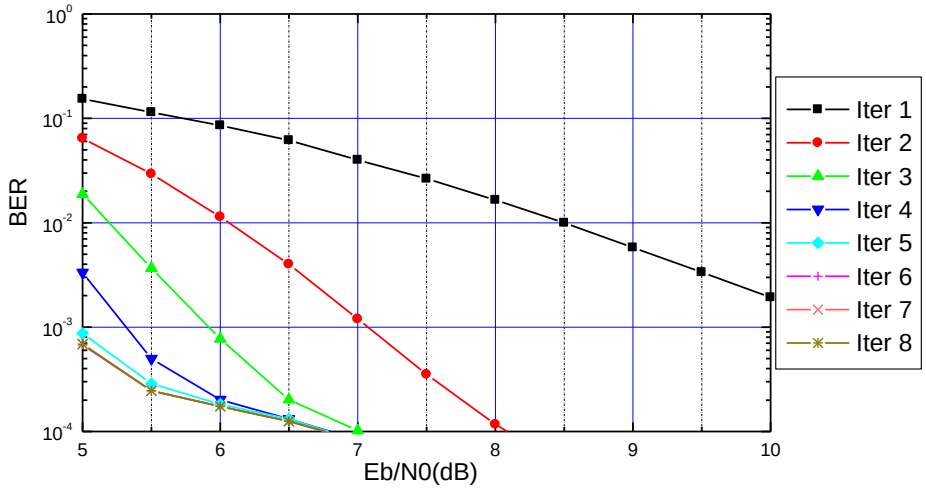


FIGURE 5.34: The BICM-ID's BER performance with MAP5/ (1,7) constellation for iterations 1- 8.

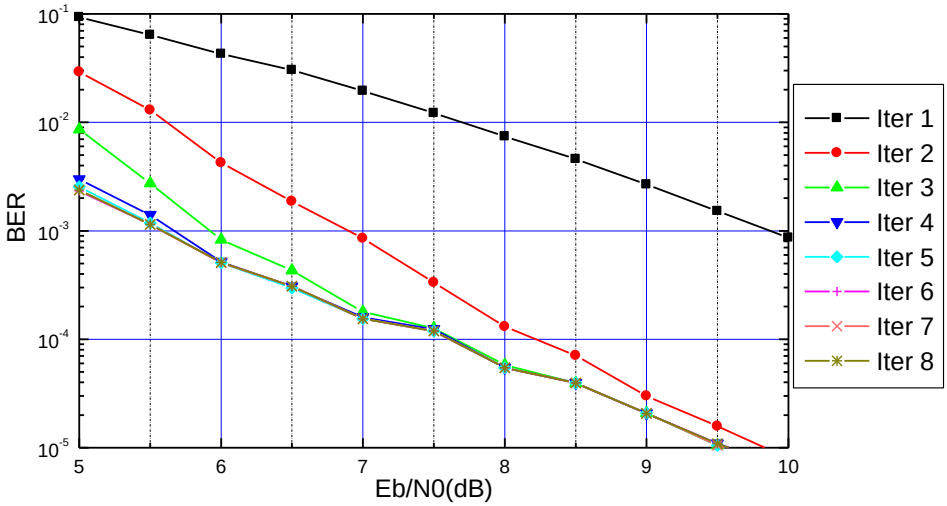


FIGURE 5.35: The BICM-ID's BER performance with MAP6/ (1,7) constellation for iterations 1-8.

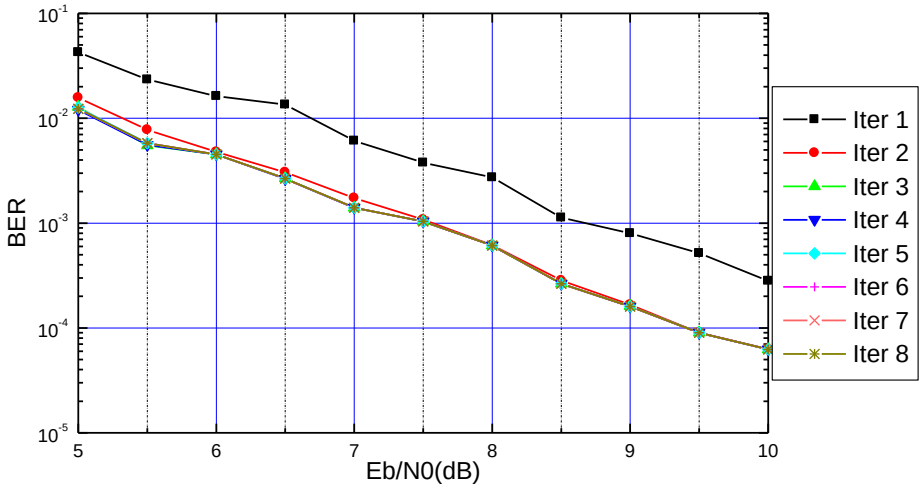


FIGURE 5.36: The BICM-ID's BER performance with MAP7/ (1,7) constellation for iterations 1-8.

The work in this subsection was published in *Paper A* (see Appendix B).

5.3 Simplified SISO algorithms

5.3.1 Simplified versions of the MAP algorithm

5.3.1.1 Historical background

The original MAP, also called BCJR [8], algorithm suffers from complexity due to its need to perform many multiplications and exponentiation.In order to reduce the MAP algorithm's complexity several variants have been proposed. A logarithmic version is proposed by Robertson *et al* [9]. In a logarithmic version, exponentials disappear and multiplications become additions.

Authors of [10] used the Jacobian logarithm to simplify the MAP algorithm. The Jacobian logarithm is given by [11]

$$ln(e^x + e^y) = max(x, y) + f_c(|x - y|) \tag{5.5}$$

where $f_c(|x - y|) = ln(1 + e^{-z})$, with $z = |x - y|$, and $f_c(.)$ is referred to as the correction or compensation term.

To reduce the computational complexity of the MAP algorithm involved when evaluating the Jacobian logarithm, authors in [12] and [13] proposed to neglect the compensation term which resulted in the so called Max-log-MAP algorithm.

Although, max-log-MAP algorithm is the least complex proposed algorithm and that it isn't sensitive to imperfect noise estimation [14] its sub-optimality yields to an unacceptable performance loss in some applications [15].

Robertson *et al* [16] proposed to approximate $f_c(.)$ by a pre-computed table and they showed that only few values need to be stored. In this approximation an 8-values correction table results in a negligible performance loss.

A slightly more complex variant than the max-log-MAP called constant-log-MAP was proposed in [17]. This variant uses a constant correction term which has two values, i.e, $f_c(z) = a$ for $z < T$, otherwise zero is added, where the two parameters a and T are optimized by computer search.

Classon *et al* [18] proposed to take $a = 0.5$, and $T = 1.5$ as the best values for UMTS' turbo code. The performance of the constant-log-MAP is between that of the max-log-MAP and the (exact)-log-MAP algorithms [19].

The constant-log-MAP algorithm is equivalent to a Robertson's approximation with a two-value correction table [19], and it can be seen as a simplified version of [16].

Although its BER performance is only slightly worse than log-MAP, the disadvantage of constant-log-MAP is that it is susceptible to noise variance estimation errors than is log-MAP [19].

A linear approximation was proposed in [15] for which $f_c(z) = a.(z - T)$ for $z < T$ and zero otherwise, where the two parameters are also optimized via computer search. The two values $a = -0.24904$, and $T = 2.5068$ were found to minimize the total squared error between the exact correction term and its linear approximated version [20].

The linear-log-MAP algorithm is between the exact-log-MAP and the constant-log-MAP algorithms in term of performance and complexity, and it converges faster than the constant-log-MAP algorithm.

5.3.1.2 BICM-ID's, with various MAP variants, performance

In the following simulations four variants of the MAP algorithm namely Exact-log-MAP, Max-log-MAP, Constant-log-MAP and Linear-log-MAP are considered. In Fig. 5.37 to 5.40, BER vs. SNR curves for BICM-ID using the previous four MAP's variants are shown. Fig. 5.41 compares the performance of these MAP's versions. A Rayleigh fading channel model is considered for all previous simulations. Table 5.6 contains simulation parameters for Fig. 5.37 to 5.41.

These figures show that, Linear-log-MAP has the nearest performance to Exact-log-MAP, followed by Constant-log-MAP and finally by Max-log-MAP that gives the poorer performance especially for the SNR interval [4,7]. Also, it is clear that Linear-log-MAP converges faster than

Constant-log-MAP.

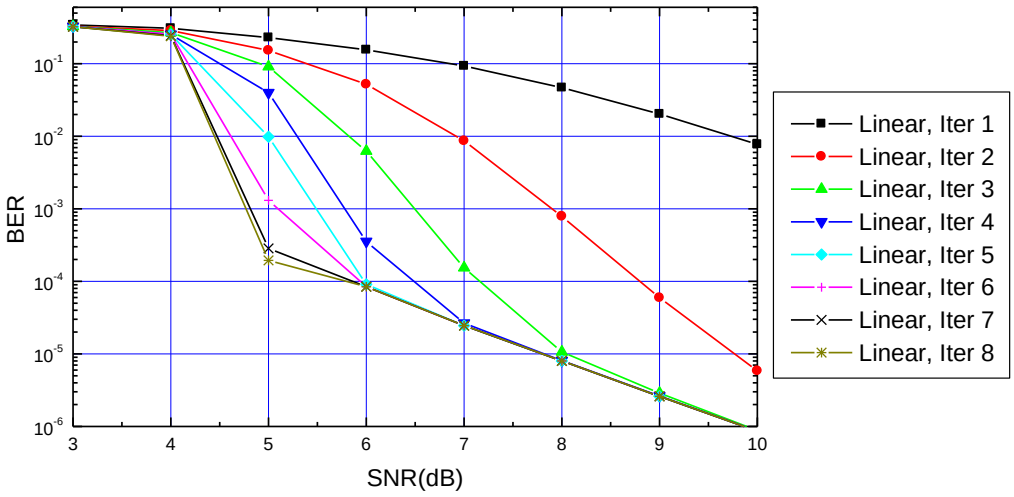


FIGURE 5.37: BER vs. SNR of BICM-ID using Linear-log-MAP.

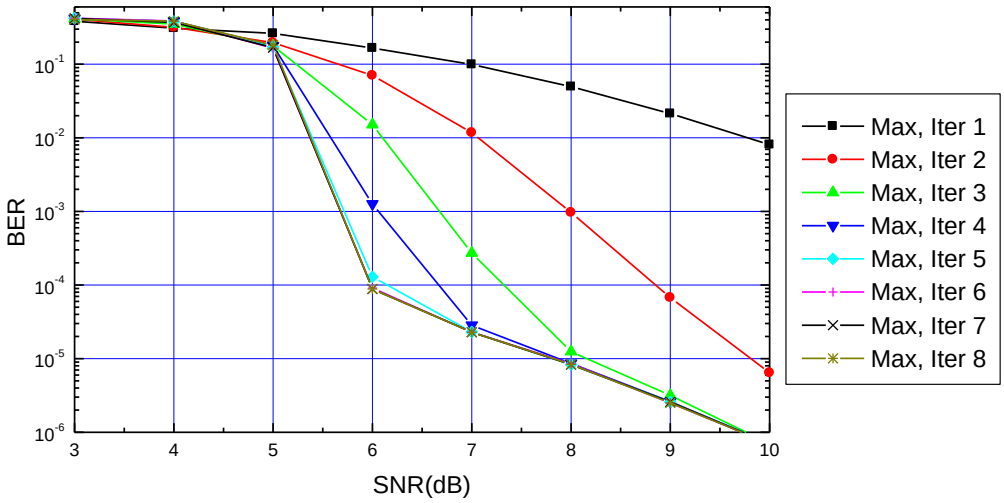


FIGURE 5.38: BER vs. SNR of BICM-ID using Max-log-MAP.

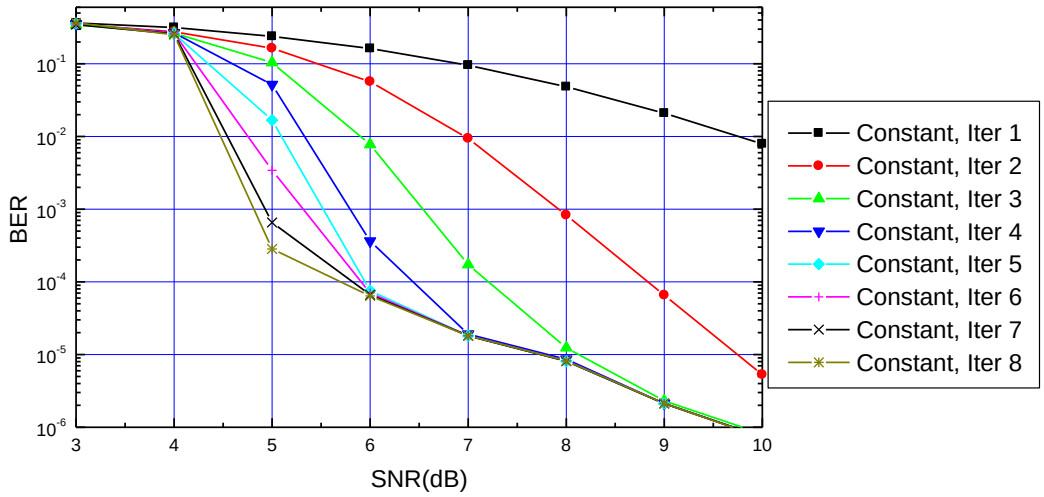


FIGURE 5.39: BER vs. SNR of BICM-ID using Constant-log-MAP.

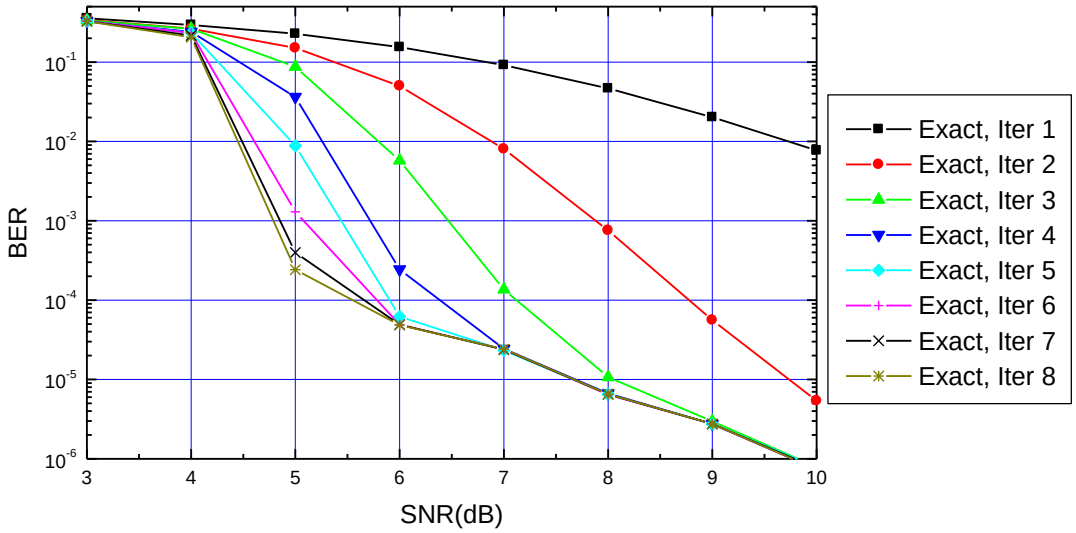
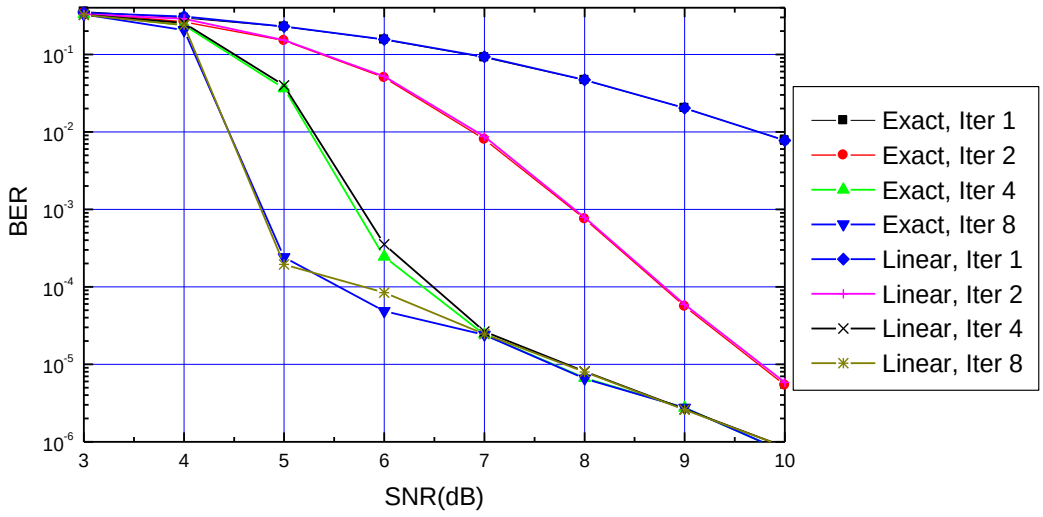


FIGURE 5.40: BER vs. SNR of BICM-ID using Exact-log-MAP.



(a)

FIGURE 5.41: Comparison of various MAP's versions. (a) linear-log-MAP vs. Exact-log-MAP. (b) Max-log-MAP vs. Exact-log-MAP. (c) Constant-log-MAP vs. Exact-log-MAP. (d) All MAP's versions.

TABLE 5.6: Simulation parameters.

Figure #	**Code generator**	**Code rate**	**Modulation /Mapping**	**MAP variant**	**Block length**	**iteration number**
5.37	$(7,5)_8$	$1/2$	Conventional 8PSK/SSP	Linear-log-MAP	6138	1 to 8
5.38	$(7,5)_8$	$1/2$	Conventional 8PSK/SSP	Max-log-MAP	6138	1 to 8
5.39	$(7,5)_8$	$1/2$	Conventional 8PSK/SSP	Constant-log-MAP	6138	1 to 8
5.40	$(7,5)_8$	$1/2$	Conventional 8PSK/SSP	Exact-log-MAP	6138	1 to 8
5.41	$(7,5)_8$	$1/2$	Conventional 8PSK/SSP	All	6138	1, 2 and 8

5.4 A new approximation to the Jacobian logarithm suitable for MAP decoding in BICM-ID

In this section, we propose a new method to approximate the compensation term in the Jacobian logarithm used by the MAP decoder. Using the proposed approximation, the complex functions $ln(.)$ and $exp(.)$ in the exact-log-MAP algorithm can be estimated with high accuracy and lower computational complexity. The efficacy of the proposed approximation is investigated and demonstrated by applying it to iteratively decoded BICM.
Due to Iterative Decoding in BICM-ID we need a repeated evaluation of the Jacobian logarithm used by the MAP decoder and this results in an inevitable high computational complexity which lead to an increase in the latency of decoding and the energy's consumption.
The main goal of the proposed approximation is to estimate the compensation term in the Jacobian logarithm by another function that is very close to it and that has less computation operations.
Fig. 5.42 illustrates a plot of the computation term's values versus the absolute values of $z = |x-y|$.

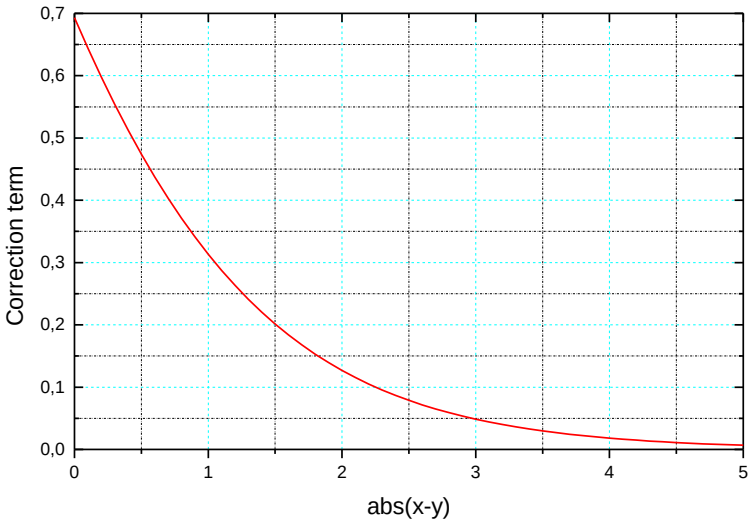


FIGURE 5.42: The exact values of $f_c(|x-y|) = ln(1+exp(-|x-y|))$ versus the absolute values |x-y|.

As can be seen from Fig. 5.42, $f_c(.)$ is a descending non-linear function and it has a negative differentiation value and a zero asymptote in infinity.
when $z \in [0, 1]$, $f_c(.)$ can be modeled by [21]

$$f_c(z) = a/(b + z) \tag{5.6}$$

The values of a and b are calculated in such a way that the error is minimized. When the maximum difference between the exact correction function and its approximated version is chosen as an error criterion Shahab [21] found that $a = 1.343$ and $b = 1.405$ are optimized values. When $z \in [0, 1]$, if we divide this range into sub-ranges we can model the correction function by linear functions, for

each sub-range, of the form

$$f_c(z) = a.z + b \tag{5.7}$$

where a and b are the slope and the intersection of each linear function. The approach of dividing the whole range into sub-ranges was first introduced in [21] for turbo codes. In our case we take 4 sub-ranges (]1,1.5],]1.5,2],]2,3], and]3,4]). From the previous figure, we can remark that when the absolute value of the difference is greater than 4, the compensation term takes almost a constant value (0.01 is used in our simulations). Fig. 5.43, shows the proposed correction function along with that used by the exact-log-MAP. As can be seen from Fig. 5.43, the proposed approximation is very precise and there isn't much difference between the exact and the proposed versions.

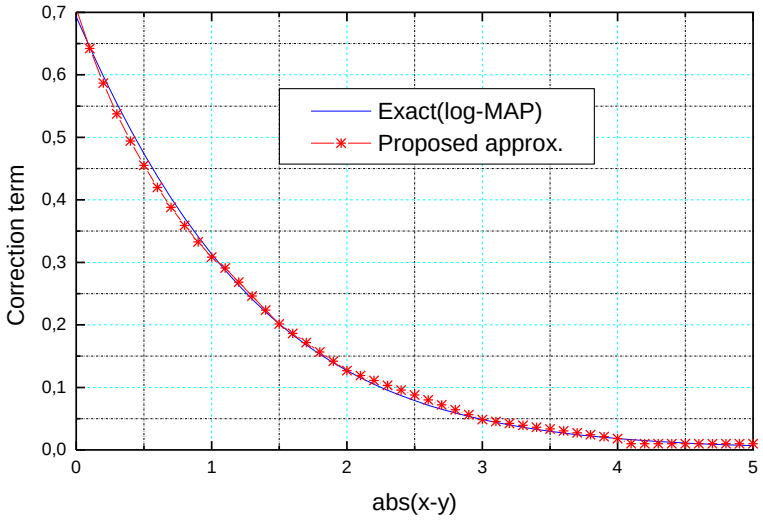


FIGURE 5.43: The exact values of the compensation term along with the proposed approximated version.

The following computer simulation results are presented for a BICM-ID using a 4-states, rate-1/2 NSC convolutional encoder with the generator polynomials . A random bit-interleaver of 12276 coded bits in length is used. A fully-interleaved Rayleigh fading channel model is considered and a conventional 8PSK constellation with SSP mapping is used.
Figure 5.44 compares the BER vs. SNR for BICM-ID of the proposed approximation with it of the exact-log-MAP for 8 iterations. Figure 5.45 compares the BER vs. SNR for BICM-ID of the proposed approximation with it of the exact-log-MAP for iteration number 8. From these comparisons, we can note that for all iterations the performance loss, except for the SNR interval [5,7] dB, is very small.

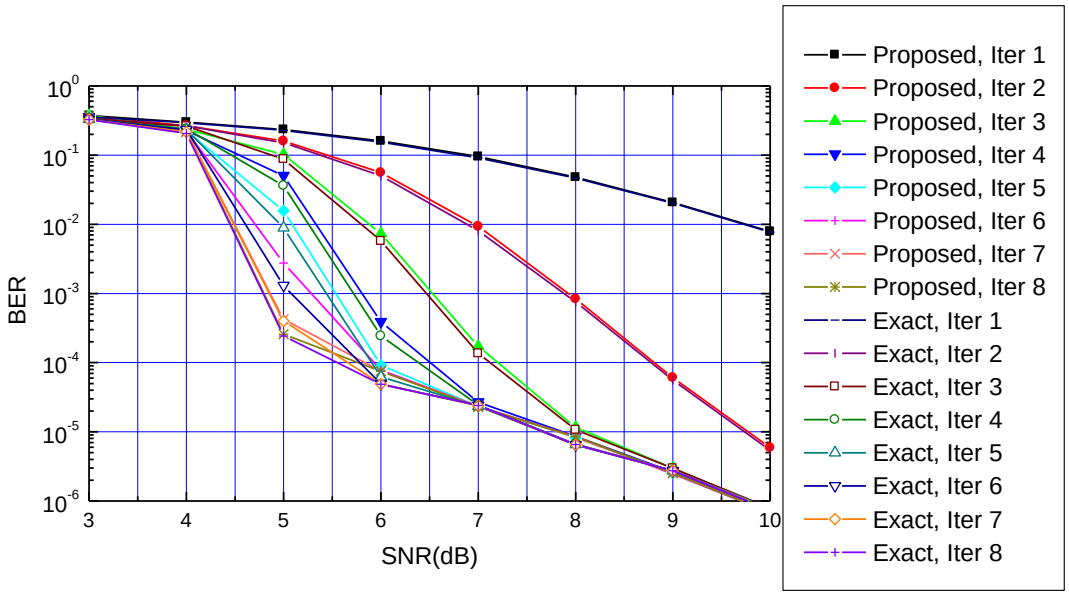


FIGURE 5.44: The BICM-ID's BER performance for Exact-log-MAP and the proposed algorithms for 8 iterations.

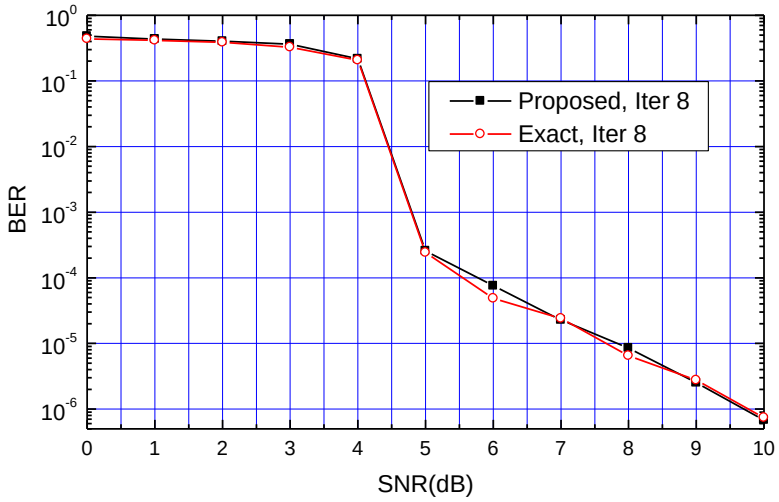


FIGURE 5.45: The BICM-ID's BER performance for Exact-log-MAP and the proposed algorithms for iteration 8.

From this table, we can see that after 8 iterations and for a SNR=5 a BER= 2.4×10^{-4} is attained by the exact-log-MAP algorithm vs. a BER= 2.5×10^{-4} attained by the proposed algorithm where there is a very small difference. The maximum performance loss is at a BER level of $BER = 7.5 \times 10^{-5}$ and it is equal to 0.5 dB.

It can be said that the difference between these two algorithms, except for the SNR interval [5,7] dB, is insignificant and they have roughly the same BER performance. We conclude that the proposed algorithm achieves nearly identical performance compared to the exact-log-MAP algorithm when they are applied to BICM-ID. We conclude that the proposed algorithm is suitable for MAP

decoding in BICM-ID.
The work in this subsection was published in *Paper B* (see Appendix B).

5.5 Conclusion

The main contribution of this thesis was presented in this chapter. A new improved symbol/8-ary constellation was introduced and a new approximation for the Jacobian logarithm used in BICM-ID's MAP decoding was proposed. Simulation results show that this latter is fairly suitable for BICM-ID's MAP decoding.
The proposed new combination MAP3/(1,7) constellation outperforms the well-known combination SSP mapping/conventional 8-PSK constellation.
The proposed approximation for the Jacobian logarithm helps to reduce the system's complexity and gives a comparable performance compared to the non-approximated one.

Bibliography

[1] S. Haykin, Communication Systems. 4^{th} edition. John Willy & Sons, Inc. 2001. ISBN: 0-471-17869-1.

[2] X. Li and J. A. Ritcey, "Trellis-Coded Modulation with Bit Interleaving and Iterative Decoding", *IEEE Journal on selected areas in commun*, vol. 17, pp. 715-724, April. 1999.

[3] J. Tan, Stuber, G. L, "Analysis and design of Symbol mappers for Iteratively Decoded BICM", *IEEE Trans. Wireless. commun*. vol.2, pp. 662-672, March. 2005.

[4] X. Li, A. Chindapol and J. A. Ritcey, "Bit interleaved coded modulation with iterative decoding and 8PSK signaling", *IEEE Trans. commun*. vol. 50, pp. 1250-1257, Aug. 2002.

[5] N. Tran, "Signal mapping design for bit interleaved coded modulation with iterative decoding (BICM-ID)", a Master thesis submitted to the department of Electrical Engineering, University of Saskatchewan. Canada, December, 2004.

[6] S. Benmahmoud, A. Djebbari and A. Taleb-ahmed, "The effects of Signal Mappings on the performances of a BICM-ID System based on 8PSK Constellation operating over a Rayleigh Fading Channel", *Proc. of the 2nd international conference on advanced computer theory and engineering*. Cairo. Egypt 2009.

[7] S. Benmahmoud, A. Djebbari, "Signal Mappings of Rectangular 8QAM Constellation for BICM-ID systems over an AWGN channel", *Proc. of the international conference on Image, Signal Processing and their Applications, ISPA'09*. Mostaganem. Alegria 2009.

[8] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv "Optimal decoding of linear codes for minimizing symbol error rate", *IEEE Trans. on Inform. Theory*, vol. .., pp. 284-287, March 1974.

[9] P. Robertson, P. Hoeher, and E. Villebrun, "Optimal and sub-optimal maximum a posteriori algorithms suitable for turbo decoding," *European Trans. on Telecommun*, Vol. 8, pp. 119–125, Mar./Apr. 1997.

[10] J. Erfanian, S. Pasupathy, and G. Gulak, "Reduced complexity symbol detectors with parallel structure for ISI channels," *IEEE Trans. Comm.*, vol. 42, pp. 1661-1671, Feb 1994.

[11] P. Robertson, E. Villebrun, and P. Hoeher, "A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain," *in Proc. IEEE Int. Conf. Comm*. (ICC), vol. 2, pp. 1009-1013, Jun 1995.

[12] W. Koch and A. Baier, "Optimum and sub-optimum detection of coded data distributed by time-varying inter-symbol interference," *in Proc. GLOBECOM*, pp. 1679-1684, 1990.

[13] A. Worm, P. Hoeher, and N. Phamdo, "Turbo decoding without SNR estimation," *IEEE Commun. Lett.*, vol. 4, no. 6, pp. 193-195, June 2000.

[14] J. F. Cheng and T. Ottosson, "Linearly approximated log-MAP algorithms for turbo decoding," *in Proc. IEEE Vehicular Technol. Conf. (VTC)*, vol. 3, pp. 2252-2256, 2000.

[15] P. Robertson, P. Hoeher, and E. Villebrun, "Optimal and sub-optimal maximum a posteriori algorithms suitable for turbo decoding," *European Trans. on Telecommun.*, Vol. 8, pp. 119–125, Mar./Apr. 1997.

[16] W. Gross and P. Gulak, "Simplfied MAP algorithm suitable for implementation of turbo decoders," *Electron. Letters*, vol. 34, no. 16, pp. 1577-1578, Aug 1998.

[17] B. Classon, K. Blankenship, and V. Desai, "Turbo decoding with the constant-log-MAP algorithm," in Proc., *Second Int. Symp. Turbo Codes and Related Appl.*, (Brest, France), pp. 467-470, Sept. 2000.

[18] M. C. Valenti, "An Efficient Software Radio Implementation of the UMTS Turbo Codec", *in Proc., 12th IEEE Int. Symp. Personal, Indoor and Mobile Radio Communications.*, (San Diego, CA), vol. 2, pp. G-108-G-113, Sep/Oct 2001.

[19] M. C. Valenti and J. Sun, "The UMTS turbo code and an efficient decoder implementation suitable for software-defined radios," *Int. J. Wireless Inf. Networks.*, vol. 8, no. 4, pp. 203-215, Oct. 2001.

[20] S. Ashoodeh, "A new constructive approximation in log-MAP Turbo Decoder", *Proc. Int. Conf. Electronics, Hardware, Wireless and optical communications*, Cambridge, UK, Feb 2008.

[21] K. Thangarajah, B. Shahrrava and M.A.S. Khalid, "A novel simplified Log-MAP algorithm suitable for hardware implementation of turbo decoding", *Proc. 24th Canadian Conference on Electrical and Computer Engineering (CCECE)*, Niagara Falls, on 8-11 May 2011.

[22] N. H. Tran and H. H. Nguyen, "Signal Mappings of 8-ary Constellations for BICM-ID Systems over a Rayleigh Fading Channel", IEICE Trans. on Commun., vol. E88-B, pp. 4083-4086, Oct. 2005.

Appendix A

Conclusion and suggestions for future research

Conclusion

This thesis considered the problem of BICM-ID's performance improvement over both Rayleigh fading and AWGN channels. First of all, we presented some basic concepts needed in our study. Then we made a detailed study on BICM, and its functioning. We developed some Matlab scripts to help simulate BICM. We also carried out some simulations to evaluate the BICM's BER performance. After this, studying turbo codes helped us to enhance our understanding of iterative decoding. We studied in details the BCJR algorithm used in the decoding of turbo codes. The basic theory underlying BICM-ID was then studied. To improve the BICM-ID's performance, we firstly studied the effects of various parameters on its BER performance, and then we proposed a new improved symbol mapper/8-ary constellation which helps to improve its performance. Finally, we studied in details MAP algorithm used in BICM-ID's decoding. This study leaded us to propose a new approximation for the Jacobian logarithm which helps to reduce the system's complexity.

Suggestions for future research

Although making some contributions, the research work in this thesis has only focused on two problems: the joint optimization of the signal constellation points with their mappings, and reducing the complexity of the MAP algorithm used in the decoding of the convolutional code used in BICM-ID. Many open problems remain to be answered. For example, analytical methods to characterize and optimize new classes of symbol mappers are to be investigated. Reducing the decoder's complexity is another suggestion for future research. The M-BCJR and the T-BCJR algorithms are to be used in BICM-ID context. Iterations' stopping criteria used in turbo codes are to be adapted for BICM-ID.

Appendix B

Included Papers

Paper A

A new improved Symbol Mapper/8-ary Constellation for BICM-ID.
Slimane. Benmahmoud, and Ali. Djebbari.
The Wireless Engineering and Technology (WET) journal. April 2013, Vol.4 No.2, PP. 65-70.
doi: 10.4236/wet.2013.42010.

Paper B

A new approximation to the Jacobian logarithm suitable for MAP decoding in BICM-ID.
Slimane. Benmahmoud, and Ali. Djebbari.
The Computer Technology and Application (CTA.) journal. 4 (2013) 241-244.

Paper C

Seven different 8-ary Constellations for BICM-ID.
Slimane. Benmahmoud, and Ali. Djebbari.
The *9th International Conference on Electronics Computer and Computation* "ICECC0'2012". 1-3 Nov. at Turgut Ozal University in Ankara, Turkey.