



Departamento de Física

# Communications Systems Based on Chaos

Francisco Javier Escribano Aparicio

Universidad Rey Juan Carlos

May, 2007



**Miguel Ángel Fernández Sanjuán**, Catedrático de Universidad perteneciente al Área de Física Aplicada y Director del Departamento de Física de la Universidad Rey Juan Carlos, y **Luis López Fernández**, Profesor Titular de Universidad de la Universidad Rey Juan Carlos

**CERTIFICAN:**

Que la presente memoria de tesis doctoral, titulada "*Communications Systems Based on Chaos*", ha sido realizada bajo nuestra dirección por **Francisco Javier Escribano Aparicio** para optar al grado de Doctor por la Universidad Rey Juan Carlos.

Y para que conste que la citada tesis reúne todos los requisitos necesarios para su defensa y aprobación, firmamos el presente certificado en Móstoles a dieciocho de mayo de dos mil siete.

Móstoles, 18 de mayo de 2007

Fdo. **Luis López Fernández**  
Profesor Titular de Universidad  
Universidad Rey Juan Carlos

Fdo. **Miguel Ángel Fernández Sanjuán**  
Catedrático de Universidad  
Director del Departamento de Física  
Universidad Rey Juan Carlos



# Acknowledgements

First of all, I would like to thank my co-supervisors, Prof. Miguel Ángel Fernández Sanjuán and Prof. Luis López Fernández, who have shown me, with their support and ideas, the successful path to the finishing of this PhD thesis.

I would also like to give special thanks to Prof. Martin Hasler and Dr. Slobodan Kozic, whose help and encouragement during my short visit at the École Polytechnique Fédéral de Lausanne, in Switzerland, have contributed decisively to the final form of this PhD thesis.

Finally, I would like to thank Prof. José Antonio Martín Pereda, Prof. Inés Pérez Mariño, Prof. Gonzalo Álvarez Marañón, Prof. José Luis Rodríguez Marrero and Prof. Francisco Javier Ramos López for having accepted taking part in the PhD thesis committee.

The research work collected in the present PhD thesis has been supported by the Spanish Ministry of Science and Technology under project BFM2003-03081, by the Spanish Ministry of Education and Science under projects FIS2006-08525 and TSI2006-07799, by the research program from the Universidad Rey Juan Carlos project number PPR-2004-03, by the Government of Madrid under projects S-0505/TIC/0285 and URJC-CM-2006-CET-0643, and by the Swiss National Science Foundation under project 200020-109474. I acknowledge as well financial support from the Universidad Rey Juan Carlos to carry out a short visit to the École Polytechnique Fédéral de Lausanne, in Switzerland, in order to work at the Laboratory of Nonlinear Systems with Prof. Martin Hasler and Dr. Slobodan Kozic, from September to December 2005.

Móstoles, 18th May 2007

Francisco Javier Escribano Aparicio



# Preface

The present PhD thesis collects the work done during the past four years on digital communications based on Chaos Theory. Our main objective was to build efficient chaos-based communications systems while trying to contribute to the study and understanding of said systems, so that our work could be a step forward in the solid foundation of a comprehensive joint digital communications and chaos-based communications theory. This had been an elusive task almost right after the initial attempts in the 90s, but our hopes were supported by the recent results on chaos coded modulations and chaotic maps controlled by small perturbations. According to this, we studied several possibilities and managed to arrive at the results we present in the following chapters. Here we proceed guided by the story of our own research, where one can almost follow the series of partial failures and partial successes which are the matter of this work.

1. **Introduction:** In this chapter, we describe the specific problem we try to address and we provide a general context for the kind of chaos-based communications we worked with. We also explain our main objectives.
2. **Coding and Decoding with Discrete Chaotic Maps:** This chapter is devoted to the examination of the class of chaotic encoding with initial conditions, both for piecewise linear and piecewise nonlinear maps. We describe the encoding method and a way to design maps, together with a number of decoding algorithms suitable for such encoding. The comparison between the systems proposed as a function of the decoding algorithms, though deluding from a practical point of view, allows us to gain an invaluable insight in this sort of chaos-based communications and opens the road for our next developments.
3. **Chaos Coded Modulations over Dispersive Channels:** Here we attempt to give an answer to the question whether there exists a communications environment where chaos-based communications can work reasonably well. We show by means of a chaotic modulation performing poorly in non-dispersive channels that it can provide some degree of robustness in two specific kinds of dispersive ones. Together with the simulation results, we provide some useful bounds which could help in the analysis and design tasks of such systems in the dispersive environments considered.

4. **Serially Concatenated Chaos Coded Modulations:** The analogy recently developed between chaos coded modulations and other standard coded and coded modulated systems suggested the work of this and the following chapters. Here we address specifically the possibility of serially concatenating channel codes and chaos coded modulations, just as an extension of the well known serial concatenation of channel encoders and trellis coded modulations. The concatenated encoder and the iterative decoder are described with the needed detail. We show that it is possible to look into the performance of the resulting communications systems using tools borrowed from digital communications theory. We validate the promising results with simulations in non-dispersive and dispersive channels, and show that the attainable performance can be comparable with standard systems.
5. **Parallel Concatenated Chaos Coded Modulations:** Following the path suggested in the last chapter, we study here the possibility of parallel concatenating two chaos coded modulations, in analogy with the turbo trellis coded modulated systems. The encoder and decoder are described and we show that, again, it is possible to draw bounds and make predictions over the final performance by using the suitable tools taken from the digital communications field. The encouraging results obtained in non-dispersive and dispersive channels, together with the validation of the predictions, make us foresee good possibilities for this kind of concatenation.
6. **Conclusions:** To conclude, we list the conclusions we had achieved thanks to the experience gained during the hazardous travel made from Chapter 1 to Chapter 5.

# Contents

<b>Preface</b>	<b>v</b>
<b>1. Introduction</b>	<b>1</b>
<b>2. Coding and Decoding with Discrete Chaotic Maps</b>	<b>5</b>
2.1. Introduction . . . . .	5
2.2. Encoding binary data with discrete chaotic maps . . . . .	6
2.2.1. Map synthesis . . . . .	10
2.3. Channel model . . . . .	14
2.4. Decoding in AWGN . . . . .	15
2.4.1. Previous remarks . . . . .	15
2.4.2. Decoding algorithms . . . . .	18
2.5. Performance analysis . . . . .	24
2.6. Simulation results . . . . .	27
2.7. Concluding remarks . . . . .	35
<b>3. Chaos Coded Modulations over Dispersive Channels</b>	<b>39</b>
3.1. Introduction . . . . .	39
3.2. System model . . . . .	40
3.2.1. Encoder . . . . .	41
3.2.2. Decoder . . . . .	43
3.3. Chaos Coded Modulations over ISI channels . . . . .	46
3.3.1. Channel model . . . . .	46
3.3.2. Performance analysis and bounds . . . . .	48
3.3.3. Simulation results . . . . .	52
3.4. Chaos Coded Modulations over frequency-non selective fading channels . . . . .	54
3.4.1. Channel model . . . . .	54
3.4.2. Performance analysis and bounds . . . . .	56
3.4.3. Simulation results . . . . .	62
3.5. Concluding remarks . . . . .	65
<b>4. Serially Concatenated Chaos Coded Modulations</b>	<b>67</b>
4.1. Introduction . . . . .	67

4.2.	System model . . . . .	68
4.2.1.	Serially concatenated encoder . . . . .	69
4.2.2.	Channel . . . . .	72
4.2.3.	Iterative decoder . . . . .	73
4.3.	Convergence analysis . . . . .	74
4.4.	Error probability analysis . . . . .	82
4.5.	Simulation results . . . . .	87
4.6.	Concluding remarks . . . . .	95
<b>5.</b>	<b>Parallel Concatenated Chaos Coded Modulations</b>	<b>97</b>
5.1.	Introduction . . . . .	97
5.2.	System model . . . . .	98
5.2.1.	Parallel concatenated encoder . . . . .	98
5.2.2.	Channel . . . . .	105
5.2.3.	Iterative decoder . . . . .	106
5.3.	Convergence analysis . . . . .	107
5.4.	Minimum squared Euclidean distance analysis . . . . .	113
5.5.	Simulation results . . . . .	127
5.6.	Concluding remarks . . . . .	137
<b>6.</b>	<b>Conclusions</b>	<b>139</b>
	BIBLIOGRAPHY	143
	Curriculum Vitae	153
	Resumen	157
	Introducción . . . . .	157
	Antecedentes . . . . .	158
	Objetivos . . . . .	159
	Conclusiones . . . . .	160

# Chapter 1

## Introduction

Since the foundation of *Chaos Theory* and its subsequent development throughout the past century [Sprott, 2003], the possibilities of this new science have attracted the attention of many researchers from many different fields. As a matter of fact, Nonlinear Dynamics and Chaos Theory constitutes a very important topic in Physics and its development has contributed to give the needed support for many disciplines in Science and Engineering [Fröhlich et al., 2000; Sprott, 2003]. From the beginning, Nonlinear Dynamics and Chaos Theory found direct application in Physics or Chemistry, where they were used to explain a number of phenomena, from the dynamics of complex systems to diffusion processes or chemical reactions. But the theory was also quickly spawned to other disciplines, like Engineering, Economy, or Social Sciences, where it could also provide valid models for a diversity of systems or for their own interactions.

Nonetheless, apart from the modeling abilities of Nonlinear Dynamics and Chaos Theory, the signals produced by chaotic systems themselves were thought soon as specially suitable for some applications thanks to their special characteristics. For example, the chaotic sequences are attractive candidates in signal analysis, signal synthesis, practical engineering, and analog and digital communications [Schimming et al., 1999; Rovatti et al., 2002]. Among the properties of chaotic signals that make them so interesting, we can name their easy generation, a highly complex and broadband nature, and a low probability of detection. Therefore, it is not surprising that they have been considered for a long time in applications for communications systems [Boltt et al., 1997; Kennedy et al., 2000b; Lau and Tse, 2003; Larson et al., 2006] or cryptography [Blackledge, 2000].

In this context, the main interest within the communications engineering field relied in incorporating the chaotic systems as natural alternatives for wideband communications. The leading work that opened the road [Hayes et al., 1993] showed that there could be ways to control a chaotic system, so that a binary information could be included in a chaotic signal and accordingly retrieved. This spawned a lot of work on the field [Kolumban et al., 1997; Kolumban et al., 1998; Kolumban and Kennedy, 2000], and allowed the design and evaluation of several kinds of communications systems in an attempt to provide practical alternatives. For example,

we can mention chaos-based modulations [Mariño, 1999; Maggio and De Feo, 2000; Schimming and Hasler, 2000; Kennedy et al., 2000a; Hasler and Schimming, 2002], channel coding based on chaos [Andersson, 1998; Chen and Wornell, 1998; Mariño et al., 2002a; Mariño et al., 2002b; López, 2003], or spread spectrum multiple access communications [Heidari-Bateni and McGillem, 1994; Schweizer and Hasler, 1996; Rovatti et al., 2001; Maggio et al., 2001; Laney et al., 2002; Argüello et al., 2002; Argüello et al., 2003; Setti et al., 2004; Tam et al., 2004; Kolumban, 2005]. A major event was the establishing of the symbolic dynamics principles [Schweizer, 1998; Maggio, 2000; Schweizer and Schimming, 2001a; Schweizer and Schimming, 2001b; Maggio and Galias, 2002; Boltt, 2003], where the chaotic sequence could be replaced by a corresponding symbolic sequence which admitted quantization and allowed its treatment at the receiver as a Markov chain decodable with known algorithms in communications theory [Proakis, 2001].

Nonetheless, the initial expectations were not always fulfilled to the desired extent once the chaotic sequence was introduced in a distorting communications channel [Kolumban et al., 2002; Abel and Schwarz, 2002; Dmitriev et al., 2003]. This was due to the fact many of the systems proposed were based on ad-hoc developments not very well understood, not easy to analyze or to generalize, or were based on simple alternatives leading to a very poor performance. These circumstances determined a relative decay in the interest originally generated. This situation has progressed a lot in recent times, because, together with a more accurate understanding of the underlying problems, issues and possibilities [Schimming, 2001; Callegari et al., 2002; Baptista et al., 2003; Hen and Merhav, 2004], a comprehensive work on the topic has shown that a growing convergence between the Chaos Theory and digital communications fields can fill in the steady gap that chaotic systems performance seemed to have with respect to standard communications systems [Schimming, 2002; Kozic, 2006].

This PhD thesis focuses on the mentioned recent developments and tries to push further the analogy between known digital communications systems and chaos-based modulators or chaos-based channel encoders. We are convinced that this analogy can greatly help in establishing practical systems with good performance and which could take real advantage of the properties of the chaotic signals in the channel. According to this, our purposes are twofold. By one side, we are interested on exploring the possibilities of already characterized chaos-based systems that use one-dimensional discrete maps to encode binary information. On the other hand, we try to propose new systems that, by combining the philosophy of binary channel coding and standard modulation techniques with chaos-based modulators, could produce a successful interaction.

According to our first objective, we will look into the chaos-based systems by encoding initial conditions which use chaotic one-dimensional discrete maps. We will try to clarify which is the role played by the invariant density of the chaotic data with respect to the chaotic dynamics of the underlying discrete chaotic map, and whether said invariant density can serve as a design criterion for such class of encoders. At the same time, we will see if piecewise nonlinear chaotic maps can also

---

be employed efficiently in this kind of setup [Escribano et al., 2006a]. The relative failure of the systems examined in point of performance face to the additive white Gaussian noise (AWGN) channel will lead us to consider which is the kind of channels where chaotic sequences of this sort can truly show their potential advantages.

Moreover, we will introduce the class of chaos coded modulations derived from chaotic encoders using maps controlled by small perturbations [Schimming et al., 2002], a generalization of which has shown to provide encoding systems with good coding gains [Kozic, 2006], and we will analyze the behaviour of the simplest chaos coded modulator example in two different kinds of dispersive environments, where, together with AWGN, the chaotic sequence will have to cope with other sources of impairment. This topic had been only partially addressed, and mainly in the context of other sorts of chaotic modulations [Kennedy et al., 2000a; Ciftei and Williams, 2001], or in multiuser situations [Lau and Tse, 2003] and the like. Little work exists so far about the results attainable with chaos coded modulations in channels other than the AWGN one. The results obtained will stress the fact that the dispersive environments are the ones where chaotic systems can make their best, even in the case of using chaotic modulators that perform poorly in AWGN channels.

Our next step will be related to the fact that there is a well established analogy between the class of chaos coded modulations and other standard coded modulations. Therefore, it is perfectly possible to consider the inclusion of chaos coded modulators in any communications system which employs any other kind of coded modulator [Escribano et al., 2005; Escribano et al., 2006c]. According to this, we will propose both serially and parallel concatenated systems where the inner encoder role, the one which sends samples to the channel, will be held by chaos coded modulators. The main reason to attempt this is, by one side, the fact that concatenated systems have proved to give good results both in non-dispersive and dispersive channels [Schlegel and Pérez, 2004], and, by the other side, the need to increase the redundancy level in the chaos coded setup in order to enhance the performance over chaos coded modulations with coding rate one sample per bit. We will see that concatenation increases the complexity of the overall system, but also provides a more desirable level of protection while being able to keep the good properties of chaos in the channel, even when the chaotic sequence is quantized to match symbolic dynamics.

First we address the possibility of serial concatenation and, once we have verified it is a valid alternative to increase the attainable performance of chaos coded modulations, we examine the possibility of parallel concatenation. With the background of the previous work, we will show that these new developments are pretty straightforward. The resulting chaos-based systems will also be good performing, though the case of parallel concatenation will require the design and evaluation of additional chaos coded modulators. The simulation results in all the cases will be accompanied, when possible, with bounds and predictions about the behaviour of the systems drawn with the help of standard tools taken from digital communications theory. This will be possible again thanks to the mentioned convergence of chaos-based communications and standard digital communications.

The objectives mentioned can only be attained if, together with the attention paid to the encoder side, we are able to adapt or propose decoding systems suitable for the situations we have enumerated. Many decoding strategies have been designed and studied in the past years in order to decode or estimate chaotic sequences under the effects of channel distortion [Papadopoulos and Wornell, 1995; Pantaleón et al., 2000; Pantaleón et al., 2002; Kisel et al., 2001; Luengo et al., 2002; Pantaleón et al., 2003; Xiaofeng et al., 2004; Luengo, 2006], but they will not be always the best possibility for the encoders considered here. While we will adapt and evaluate a number of known decoders for the kind of systems working on encoding initial conditions [Escribano et al., 2006a; Escribano et al., 2006b], we will propose scalable decoder modules able for iterative decoding of concatenated systems [Escribano et al., 2005]. In any case, we will not forget either that, depending on the kind of channel considered, such decoding systems could require certain arrangements.

With this background and this itinerary on mind, we will proceed in the next four chapters to the fulfillment of these objectives, to the extent they can be covered in one PhD thesis of this kind and size. However, we will prove that chaos-based communications are nearly to be ripe enough to take part in real world practical applications on equal grounds with existing systems.

## Chapter 2

# Coding and Decoding with Discrete Chaotic Maps

### 2.1. Introduction

Chapter 1 has contributed a bit to the history about the topic of chaos-based communications and has shown the path through which we have arrived at the current state of the art. We have mentioned that chaos-based modulations and channel encoders derived from chaotic systems attracted much attention. This was due, for example, to the foreseen possibility of employing them in multiuser systems as efficient means of sharing the access to the channel, but the interest dropped somewhat during a time due to the bad performance of the systems proposed in the additive white Gaussian noise (AWGN) channel [Abel and Schwarz, 2002; Dmitriev et al., 2003]. We have nevertheless witnessed the arising of new developments in chaotic direct modulations [Schimming and Hasler, 2002; Kozic et al., 2003a; Kozic and Schimming, 2005] that showed that what was lacking was a deeper understanding of the joint possibilities of digital communications and Chaos Theory, and that a most promising path lies in finding analogies between standard digital systems and chaos-based encoders.

The design of channel coding and decoding algorithms based on discrete chaotic maps is one of the possibilities studied during the last years. It is known in communications theory that the knowledge of the probability density function (pdf) of the data produced by an encoder or modulator is potentially useful in evaluating the performance of the communications systems, and thus the possibility of designing a map knowing the invariant pdf of the data is not a topic deprived of interest. The task of designing maps with desired pdf's had been addressed in the mid 90's [Baranovsky and Daems, 1995]. However, the trouble is still the design of the chaos-based encoding algorithm itself when the resulting map is piecewise nonlinear and when it does not match the convolutional encoder equivalence that greatly simplifies the encoding rules in the recent class of chaos coded modulated systems [Schimming and Hasler, 2002]. The possibility we have chosen here is to take advantage of an already known encoding algorithm based on the Bernoulli shift map (BSM) and, using the properties of conjugacy between maps, expand it to the whole class of discrete chaotic maps conjugated to the BSM.

The design of a map suitable for this framework will require some restrictions regarding the form of the invariant pdf and the symmetry properties of the map. The importance of this design procedure lies in the fact that such a device has been thoroughly studied for piecewise linear maps (PWL maps, which generate the so-called piecewise linear Markov maps [Schweizer and Schimming, 2001a; Schweizer and Schimming, 2001b]), whose pdf can be given exactly and readily thanks to the piecewise linearity of the systems involved. On the other hand, once we clarify the way to build a chaos-based modulation using a piecewise nonlinear discrete chaotic map with known invariant density, there remains the open topic of how the interaction between the data pdf and the chaotic trajectory can affect the performance. The question we want to answer is whether we can expect better behaviour with this piecewise nonlinear alternative face to the systems derived from piecewise linear maps.

According to all this, in this chapter we will follow closely the developments of [Escribano et al., 2006a] and [Escribano et al., 2006b]. The chapter is thus structured as follows. In Section 2.2 we introduce the encoding process chosen for the kind of discrete chaotic maps considered, including the method to design a map starting from its desired invariant probability density function. Section 2.3 is devoted to a brief description of the AWGN channel. Section 2.4 explains in detail the four decoding algorithms which will serve us to analyze the possibilities of the proposed encoding maps. Section 2.5 is devoted to the derivation of bounds for a simple case, but that will serve us to better understand the chaos-based encoding system. In Section 2.6, we plot and analyze the simulation results together with the proposed bounds. Finally, Section 2.7 gathers the conclusions.

## 2.2. Encoding binary data with discrete chaotic maps

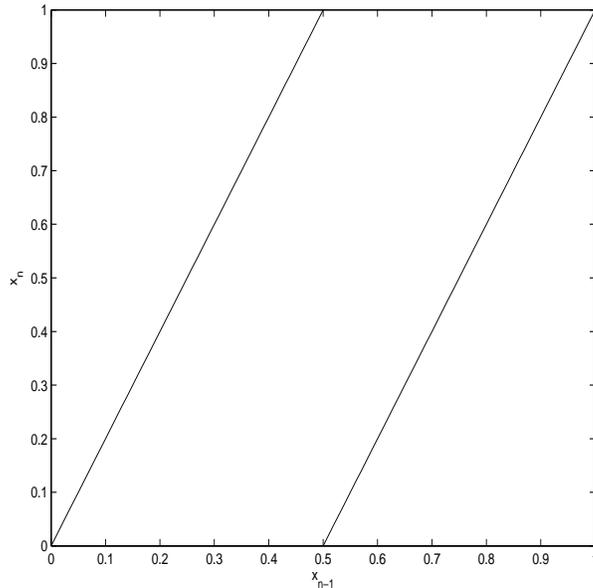
In order to encode a binary sequence of length  $N$  containing the information,  $b_n \in \{0, 1\}$ ,  $n = 1 \cdots N$ , we make use of one method extensively used [Ciftci and Williams, 2001; Baptista and López, 2002; Kozic et al., 2003a; Kozic et al., 2003b; Kozic and Schimming, 2005], which employs an effect of truncation which will be explained in the sequel. This method, based in other one known generally as encoding initial conditions, has been described for piecewise linear chaotic maps, and we extend it here for piecewise nonlinear chaotic maps. In all the developments that follow, the sequence  $b_n$  is a binary sequence, independently and identically distributed (*iid*), with  $P(b_n = 0) = P(b_n = 1) = 1/2$ . The maps we are interested in throughout this chapter are limited to the interval  $[0, 1]$  and leave said interval invariant

$$f(x) : [0, 1] \rightarrow [0, 1]. \quad (2.1)$$

The base for the encoding rule is the Bernoulli shift map (BSM from now on), whose expression is

$$x_n = f(x_{n-1}) = \begin{cases} 2x_{n-1} & \text{if } x_{n-1} < \frac{1}{2} \\ 2x_{n-1} - 1 & \text{if } x_{n-1} \geq \frac{1}{2} \end{cases}. \quad (2.2)$$

This map is depicted in Fig. 2.1. The BSM is also usually denoted as  $f(x) = 2x \bmod 1$ .



**Figure 2.1.** *Diagram of the Bernoulli shift map (BSM), with its two linear sections with slope +2 and its corresponding antisymmetry with respect to the vertical line  $x = 1/2$ .*

It is a well known property of the Bernoulli shift map [Chen and Wornell, 1998] that, if we define the symbolic state of the system  $r$  as the real number

$$r = \sum_{m=1}^N b_m 2^{-m}, \quad (2.3)$$

and we take as initial condition for the chaotic sequence the value  $x_0 = r$ , then the binary sequence is encoded into the chaotic sequence generated by  $x_0$ . In this situation, the iterations through the BSM take values

$$x_n = f(x_{n-1}) = f^n(x_0) = \sum_{m=n+1}^N b_m 2^{-m+n}, \quad (2.4)$$

where  $f^n(\cdot)$  is the result of applying the map  $n$  times,  $n = 0, \dots, N - 1$ . In this case, the binary information can simply be retrieved following

$$b_{n+1} = \left\lfloor x_n + \frac{1}{2} \right\rfloor, \quad (2.5)$$

where  $\lfloor x \rfloor$  represents the maximum integer just below  $x$ . This method is generally known as encoding initial conditions [Xiaofeng et al., 2004].

For a real system, where the length  $N$  of the binary message to be transmitted could reach thousands of bits even when working in packet mode, the proposed encoding process is not practical, since it would require an almost infinite precision to process the values of  $r$ ,  $x_0$  and the subsequent map iterations in the form of Eq. (2.4). However, the same principles can be used to encode blocks of just  $D \ll N$  bits at a time. If we define the truncated symbolic sequence at time  $n$  as

$$r'_n = \sum_{m=n+1}^{n+D} b_m 2^{-m+n}, \quad (2.6)$$

then the corresponding truncated value for  $x'_n$  in the BSM case is

$$x'_n = g(r'_n) = r'_n, \quad (2.7)$$

where  $g(r)$  is a function whose meaning will be seen in the following. This function is the identity in the case of the BSM,  $g(r) = r$ . Eq. (2.5) still holds by replacing  $x_n$  by  $x'_n$ . The resulting truncated sequence  $x'_n$  is close to the original one when  $D$  is large enough, and the process is equivalent to the addition of noise in order to control the chaotic sequence

$$x'_n = x_n + \eta_n. \quad (2.8)$$

The quantity  $\eta_n = -\sum_{m=n+D+1}^N b_m 2^{-m+n} = O(2^{-D-1})$  could be seen approximately as a noisy sequence whose power decreases with  $D$ . Note also that this truncation method is an instance of quantization of  $x_n$  down to a precision of  $D$  bits, so that there is only a total of  $2^D$  real-valued possible different values for  $x'_n$ . Therefore, the resulting sequence corresponds to what is called a discrete chaotic map [Setti et al., 2002]. In the following, we will refer to the controlled discrete chaotic sequence simply as  $x_n$  and the symbolic state sequence of Eq. (2.6) as  $r_n$ . Therefore, we have a pair of encoding functions that we rewrite for clarity

$$r_n = \sum_{m=n+1}^{n+D} b_m 2^{-m+n} \quad (2.9)$$

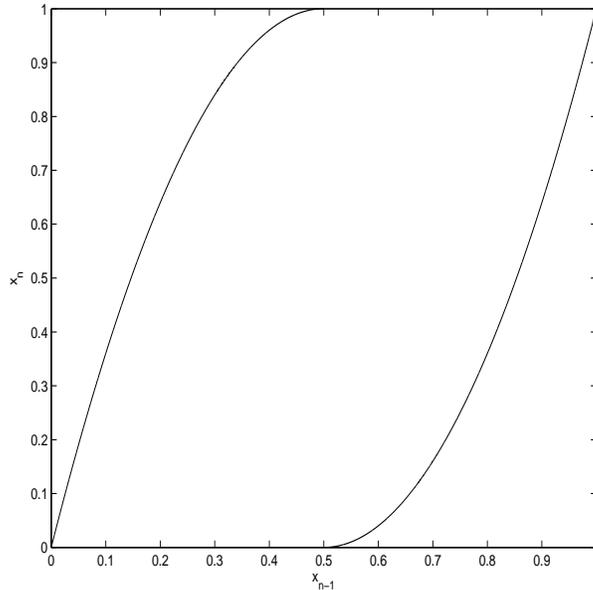
$$x_n = g(r_n). \quad (2.10)$$

They can be used to encode data with the BSM and with other chaotic maps whenever we can find a suitable expression for the corresponding  $g(r)$  function. The function  $x_0 = g(r)$ , when it exists for a specific map, makes it possible for the chaotic sequence generated by such map to carry the information bit sequence and recover it following Eq. (2.5).

Other map we will consider is the logistic map, defined as well in  $[0, 1]$  as

$$x_n = f(x_{n-1}) = 4x_{n-1}(1 - x_{n-1}). \quad (2.11)$$

However, this map is continuous in the definition interval, convex and symmetric with respect to  $x_n = 1/2$ . Here we are interested in antisymmetric maps, because



**Figure 2.2.** Diagram of the modified logistic map (MLM), with its two parabolic sections and its corresponding antisymmetry with respect to the vertical line  $x = 1/2$ .

the symmetric piecewise linear maps are known to offer potentially poor squared Euclidean distance properties between two different encoded sequences  $x_n$  and  $x_n^*$  [Kozic et al., 2003b], and no better results are to be expected with symmetric piecewise nonlinear maps<sup>1</sup>. The squared Euclidean distance between these chaotic sequences  $x_n$  and  $x_n^*$  is defined as

$$d_E^2 = \sum_{m=1}^N (x_m - x_m^*)^2, \quad (2.12)$$

and it has been shown to tend to 0 for the symmetric counterpart of the BSM, i.e., the tent map [Kozic et al., 2003b]. Therefore, we will use the antisymmetric version of the logistic map, given by

$$x_n = f(x_{n-1}) = \begin{cases} 4x_{n-1}(1 - x_{n-1}) & \text{if } x_{n-1} < \frac{1}{2} \\ 1 - 4x_{n-1}(1 - x_{n-1}) & \text{if } x_{n-1} \geq \frac{1}{2} \end{cases}, \quad (2.13)$$

and we will accordingly call it the *modified logistic map* (MLM). It is depicted in Fig. 2.2. The whole class of maps in  $[0, 1]$  that accomplish the antisymmetric condition

$$f(x) = 1 - f\left(x + \frac{1}{2}\right), \quad x \in \left[0, \frac{1}{2}\right], \quad (2.14)$$

<sup>1</sup>It has been shown that the performance of this kind of chaos-based encoding for high signal to noise ratios is linked to the minimum squared Euclidean distance between pairs of chaotic encoded sequences [Kozic, 2006].

is conjugated to the BSM and, as we will demonstrate in the next subsection,  $g(r)$  can be obtained directly from the conjugating function. In fact, if a map is conjugated to the BSM, we will see that the function  $g(r)$  exists and is related to the invariant density  $p(x)$  of such map through [Kennedy et al., 2000b]  $g(r) = F^{-1}(r)$ , where  $F(x) = \int_0^x p(t)dt$ ,  $x \in [0, 1]$ , is the distribution function of the corresponding invariant density.

The probability density function (or, more exactly, the invariant density) of the chaotic data  $x_n$  without truncation is uniform for the BSM, i.e.,  $p(x) = 1$  when  $x \in [0, 1]$ . For the MLM the pdf is the same as in the case of the logistic map itself, namely [Sprott, 2003]

$$p(x) = \frac{1}{\pi\sqrt{x(1-x)}} \quad x \in [0, 1]. \quad (2.15)$$

This is a consequence of the fact that for a one-dimensional map with two branches and two possible inverse values, the pdf has to meet the condition [Sprott, 2003]

$$p(x) = \left| \frac{p(f_1^{-1}(x))}{f'(f_1^{-1}(x))} \right| + \left| \frac{p(f_2^{-1}(x))}{f'(f_2^{-1}(x))} \right|, \quad (2.16)$$

where  $f_1^{-1}(x)$  and  $f_2^{-1}(x)$  are the respective two inverse values and  $f'(\cdot)$  is the first derivative of the map. The absolute value in the expression gives reason of the equal invariant densities for both symmetries.

Therefore, the distribution function of the MLM will be  $F(x) = 1 - (2/\pi)\text{acos}(\sqrt{x})$  and it is easy to verify that, in this case

$$g(r) = F^{-1}(r) = \cos^2\left(\frac{\pi}{2}(1-r)\right). \quad (2.17)$$

If we apply Eqs. (2.9) and (2.10) with this definition, the same decoding property stated in Eq. (2.5) holds and the chaotic sequence  $x_n$  produced carries the binary information while following the MLM dynamics. It is worth noting that the same happens with the BSM itself, since  $F(x) = x$ . We will illustrate in the next subsection how, following the conjugacy properties between this class of maps, we can design maps with the mentioned symmetry just by employing the desired pdf shape as main design criterion.

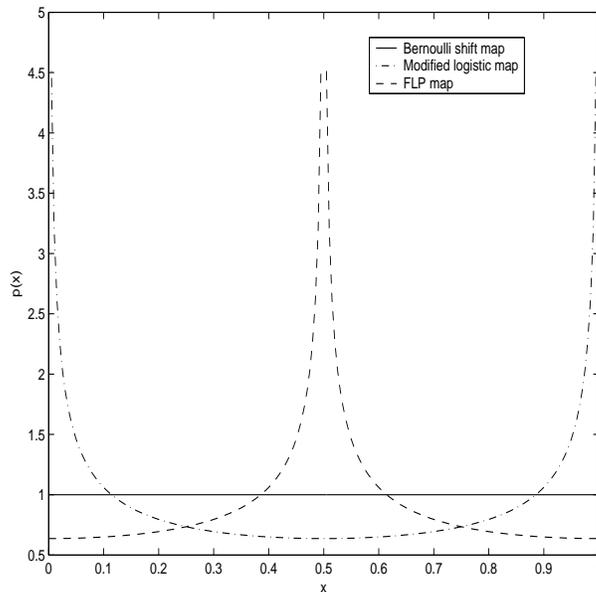
### 2.2.1. Map synthesis

Following [Kennedy et al., 2000b], and using the conjugation properties between maps that leave invariant the interval  $[0, 1]$  and comply with some symmetry conditions, we will show how to design new piecewise nonlinear maps. This will allow us to illustrate one possible method for map synthesis. The map obtained, together with the BSM and the MLM, will be used in the developments of the subsequent sections and will serve us to perform comparisons between the final bit error rate (BER) of the three schemes as a function of the distribution of the data and of the

map dynamics. For convenience, we choose as final density a pdf that has the same shape as the pdf of MLM, but with the maximum in the threshold point  $x = 1/2$ , i.e.,

$$\begin{aligned}
 p(x) &= \frac{1}{\pi \sqrt{(x + \frac{1}{2})(\frac{1}{2} - x)}} & x \in \left[0, \frac{1}{2}\right], \\
 p(x) &= \frac{1}{\pi \sqrt{(x - \frac{1}{2})(\frac{3}{2} - x)}} & x \in \left(\frac{1}{2}, 1\right].
 \end{aligned}
 \tag{2.18}$$

We will see that we can design a map conjugated to the BSM whose chaotic data will follow this pdf. The map we will obtain through this flipped logistic pdf will be called *flipped logistic pdf* map (abbreviated as FLPM). All the probability density functions (pdf's) involved are shown in Fig. 2.3. Note that they correspond to three different cases of interest when transmitting data with chaotic maps using the described framework. As we can recover directly the binary data with Eq. (2.5), we have a *critical point* in the threshold  $x = 1/2$  which serves to discriminate between the cases  $b_n = 0$  and  $b_n = 1$ . We can see that the BSM distributes the data indifferently in the complete definition interval, the MLM concentrates less data around the threshold, and the FLPM concentrates most of the data around it. This will be of consequence when evaluating the performance of the systems, though it will not give complete reason of the behaviour of the corresponding encoding systems.



**Figure 2.3.** Invariant probability density functions for the BSM (continuous line), MLM (dash-dotted line) and FLPM (dashed line).

We mentioned before that the function  $g(r)$  which relates the initial condition  $x_0$  to the symbolic state  $r$  as defined in Eq. (2.3) can be obtained starting from

the pdf of the map following  $g(r) = F^{-1}(r)$ . We are going to review the reason for this. In the case when the map is conjugated to the BSM, if the initial condition for the Bernoulli map is  $x_0^B = r$  and the initial condition for the conjugated map is  $x_0^C = g(r)$ , then all the samples are related through [Kennedy et al., 2000b]

$$f_C^n(x_0^C) = f_C^n(g(r)) = g(f_B^n(g^{-1}(x_0^C))) = g(f_B^n(r)) = g(f_B^n(x_0^B)), \quad (2.19)$$

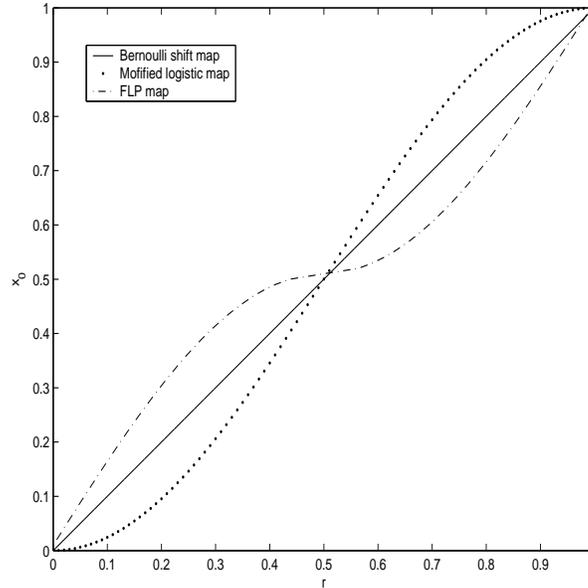
where  $f_C^n(x)$  is the  $n$ -th iteration of the conjugated map and  $f_B^n(x)$  is the  $n$ -th iteration of the Bernoulli shift map. The last expression stems from the definition of conjugacy between two maps  $f_1$  and  $f_2$  defined in the interval  $[0, 1]$  and leaving it invariant,

$$f_1(x) = g(f_2(g^{-1}(x))), \quad (2.20)$$

where  $g(x)$  is a monotonous growing function mapping the interval  $[0, 1]$  onto itself. Clearly, the truncated encoding process described in Eq. (2.9) makes use of the properties of Eq. (2.19). By definition of the distribution function of the conjugated map  $F_C(x)$ ,

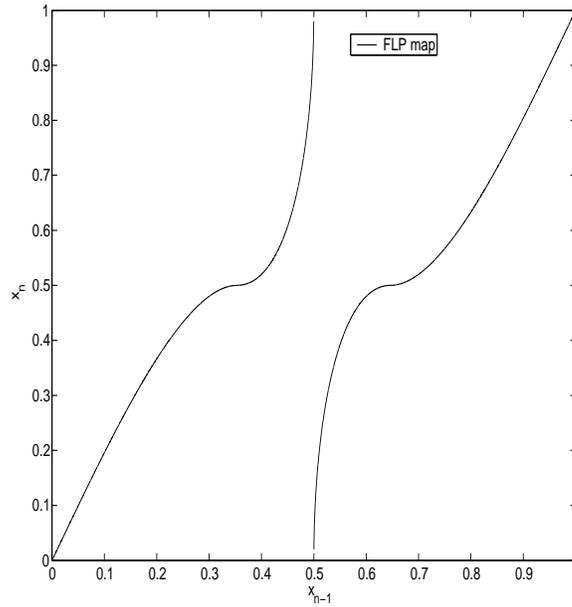
$$\begin{aligned} F_C(x) &= P(f_C^n(x_0^C) < x) = P(f_B^n(g^{-1}(t)) < g^{-1}(x)) = \\ &= P(f_B^n(x_0^B) < g^{-1}(x)) = F_B(g^{-1}(x)) = g^{-1}(x), \end{aligned} \quad (2.21)$$

since the invariant pdf of the BSM data is uniform in  $[0, 1]$ .



**Figure 2.4.** The different conjugating functions  $g(x)$  for the BSM (straight continuous line), the MLM (dotted line) and the FLPM (dash-dotted line).

Therefore, the next task in the design process will be to find the expression for  $f_C(x)$  starting from the pdf definition of Eq. (2.18) and using Eq. (2.19). Since



**Figure 2.5.** *The figure shows the flipped logistic pdf map (FLPM).*

$g(r) = F^{-1}(r)$  and the pdf is symmetric with respect to the point  $1/2$ ,  $g(r)$  has the following properties

$$g(0) = 0, \quad g\left(\frac{1}{2}\right) = \frac{1}{2}, \quad g(1) = 1 \quad (2.22)$$

and

$$g\left(\left[0, \frac{1}{2}\right]\right) = \left[0, \frac{1}{2}\right], \quad g\left(\left[\frac{1}{2}, 1\right]\right) = \left[\frac{1}{2}, 1\right]. \quad (2.23)$$

Besides, we know that  $g(r)$  is a monotonously growing function. Taking into account the Eq. (2.19) and the expression of the BSM, the new map  $f_C(x)$  will also have the following property

$$f_C(g(r)) = \begin{cases} g(2r) & \text{if } r < \frac{1}{2} \\ g(2r - 1) & \text{if } r \geq \frac{1}{2} \end{cases}. \quad (2.24)$$

As  $g(r)$  is invertible, taking  $g(r) = x$  and  $r = g^{-1}(x)$ , we arrive at

$$f_C(x) = \begin{cases} g(2g^{-1}(x)) & \text{if } x < \frac{1}{2} \\ g(2g^{-1}(x) - 1) & \text{if } x \geq \frac{1}{2} \end{cases}, \quad (2.25)$$

which is the general expression for the new map, provided that its pdf lies in the interval  $[0, 1]$  and is symmetric with respect to  $x = 1/2$ .

After some algebra, we can calculate  $F(x)$ ,  $g(r)$  and the map for the pdf defined in Eq. (2.18). The corresponding expressions are

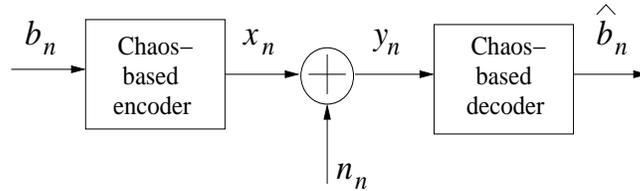
$$F(x) = \begin{cases} \frac{1}{2} - \frac{2}{\pi} \arccos \left( \sqrt{x + \frac{1}{2}} \right) & \text{if } x < \frac{1}{2} \\ \frac{3}{2} - \frac{2}{\pi} \arccos \left( \sqrt{x - \frac{1}{2}} \right) & \text{if } x \geq \frac{1}{2} \end{cases} \quad (2.26)$$

$$g(r) = \begin{cases} \cos^2 \left( \frac{\pi}{2} \left( \frac{1}{2} - r \right) \right) - \frac{1}{2} & \text{if } x < \frac{1}{2} \\ \cos^2 \left( \frac{\pi}{2} \left( \frac{3}{2} - r \right) \right) + \frac{1}{2} & \text{if } x \geq \frac{1}{2} \end{cases} \quad (2.27)$$

$$x_n = f(x_{n-1}) = \begin{cases} 4x_{n-1} \sqrt{\frac{1}{4} - x_{n-1}^2} & \text{if } 0 \leq x_{n-1} < \frac{1}{2\sqrt{2}} \\ 1 - 4x_{n-1} \sqrt{\frac{1}{4} - x_{n-1}^2} & \text{if } \frac{1}{2\sqrt{2}} \leq x_{n-1} < \frac{1}{2} \\ 4(1 - x_{n-1}) \sqrt{\frac{1}{4} - (1 - x_{n-1})^2} & \text{if } \frac{1}{2} \leq x_{n-1} < 1 - \frac{1}{2\sqrt{2}} \\ 1 - 4(1 - x_{n-1}) \sqrt{\frac{1}{4} - (1 - x_{n-1})^2} & \text{if } 1 - \frac{1}{2\sqrt{2}} \leq x_{n-1} \leq 1. \end{cases} \quad (2.28)$$

The different  $g(r)$  functions for the three maps involved are depicted in Fig. 2.4, and the FLPM can be seen in Fig. 2.5. We could now use these three maps to encode a binary sequence using the Eqs. (2.9) and (2.10) and see how efficient the resulting encoding systems are in terms of their BER performance when the chaotic samples are distorted by the communications channel. Note that, for the encoding process alone,  $g(r)$  is the only function needed, but, as we will see, the resulting map  $f(x)$  is required on the decoder side.

### 2.3. Channel model



**Figure 2.6.** Block diagram of the described system, consisting on a chaos-based encoder, an AWGN channel and a chaos-based decoder.

In normal communications systems, one of the most common source of impairment in the channel is the presence of additive white Gaussian noise (AWGN) [Proakis, 2001]. According to this and as a first approach, we will evaluate the performance of the proposed systems in a pure AWGN channel. In this case, if we send the chaotic samples  $x_n$  without further processing to the channel, the sequence arriving to the receiver side,  $y_n$ , will be described as

$$y_n = x_n + n_n, \quad (2.29)$$

where  $n_n$  are *iid* samples of AWGN following a Gaussian distribution with zero mean and power  $\sigma^2$ . The pdf of the noise samples will accordingly be

$$p(n) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{n^2}{\sigma^2}}. \quad (2.30)$$

The complete model of the communications system can be seen in Fig. 2.6, including the chaos-based encoder, the channel, and the decoder. The possibilities for this decoding block will be described in the next section.

## 2.4. Decoding in AWGN

### 2.4.1. Previous remarks

In order to take advantage of the potential properties of the chaotic sequence in AWGN, a suitable decoding method has to be proposed. In fact, if we decode in a symbol-by-symbol basis, by hard and directly deciding over each  $y_n$ ,

$$\begin{aligned} y_n < \frac{1}{2} &\rightarrow \hat{b}_{n+1} = 0 \\ y_n \geq \frac{1}{2} &\rightarrow \hat{b}_{n+1} = 1. \end{aligned} \quad (2.31)$$

no gain is achieved with respect to the direct transmission of the source message  $b_n$  as will be shown next. Since the pdf's of all the maps are known, it is straightforward to calculate the expressions for the bit error probability in each case, taking into account that

$$P_b = P\left(y_n < \frac{1}{2}, x_n \geq \frac{1}{2}\right) + P\left(y_n \geq \frac{1}{2}, x_n < \frac{1}{2}\right). \quad (2.32)$$

By symmetry

$$\begin{aligned} P_b &= 2P\left(y_n \geq \frac{1}{2}, x_n < \frac{1}{2}\right) = 2P\left(n_n \geq \frac{1}{2} - x_n, x_n < \frac{1}{2}\right) = \\ &= 2 \int_0^{\frac{1}{2}} p(x) \int_{\frac{1}{2}-x}^{\infty} p(n) dn dx = \int_0^{\frac{1}{2}} p(x) \operatorname{erfc}\left(\left(\frac{1}{2} - x\right) \frac{1}{\sqrt{2}\sigma}\right) dx, \end{aligned} \quad (2.33)$$

where we have used the known relation [Proakis, 2001]

$$\int_x^{\infty} p(n) dn = \frac{1}{2} \operatorname{erfc}\left(\frac{x}{\sqrt{2}\sigma}\right), \quad (2.34)$$

and the fact that  $x_n$  and  $n_n$  are independent. The resulting bit error probability can be calculated analytically only for the BSM encoder after some algebra, while for the other two map encoders it has to be calculated by numerical integration

[Abramowitz and Stegun, 1965]. The corresponding expressions are

$$P_b^B = \frac{1}{2} \operatorname{erfc} \left( \sqrt{\frac{3E_b}{N_0}} \right) + \frac{1}{\sqrt{12\pi \frac{E_b}{N_0}}} \left( 1 - e^{-3\frac{E_b}{N_0}} \right) \quad (2.35)$$

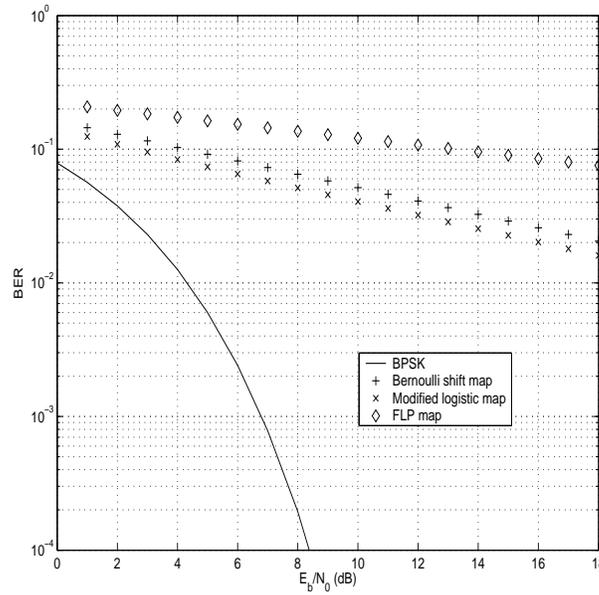
$$P_b^M = \int_0^{\frac{1}{2}} \frac{1}{\pi \sqrt{x(1-x)}} \operatorname{erfc} \left( (1-2x) \sqrt{2\frac{E_b}{N_0}} \right) dx \quad (2.36)$$

$$P_b^F = \int_0^{\frac{1}{2}} \frac{1}{\pi \sqrt{(x+\frac{1}{2})(\frac{1}{2}-x)}} \operatorname{erfc} \left( (1-2x) \sqrt{\frac{2}{3-\frac{8}{\pi}} \frac{E_b}{N_0}} \right) dx, \quad (2.37)$$

where  $E_b/N_0$  is the well known signal-to-noise ratio [Haykin, 2001] in terms of energy per bit to power spectral density of the noise. It is given by

$$\frac{E_b}{N_0} = \frac{P}{2\sigma^2}, \quad (2.38)$$

where  $P$  is the power of the chaotic sequence and  $\sigma^2$  is the power of the AWGN



**Figure 2.7.** Bit error probability in the case of direct decoding of the received sequence  $y_n$  in the AWGN channel for each of the possible encoders (based on the BSM, the MLM or the FLPM). The bit error probability of BPSK is shown for comparison.

process.  $P$  corresponds also to the variance of the sequence  $x_n$ , which in the case of the BSM, for  $D$  high enough to make the quantization noise negligible, is  $\sigma_x^2 = 1/12$ ; in the case of the MLM,  $\sigma_x^2 = 1/8$ ; and, in the case of the FLPM,  $\sigma_x^2 = 3/8 - 1/\pi$ . The expression  $P_b^B$  corresponds to the BSM,  $P_b^M$  to the MLM and  $P_b^F$  to the FLPM.

The results are depicted in Fig. 2.7, together with the bit error probability for binary phase shift keying (BPSK). In this case, the signal in the channel is  $x_n = 2b_{n+1} - 1$ , and the bit error probability is given by [Proakis, 2001]

$$P_b = \frac{1}{2} \operatorname{erfc} \left( \sqrt{\frac{E_b}{N_0}} \right). \quad (2.39)$$

It is clear that the best is to employ BPSK, since the distance between symbols is maximized with respect to the chaotic systems proposed. The worst system is the based on the FLPM, since a large number of samples lie around the threshold point  $x = 1/2$ , whereas the MLM slightly improves the performance of the BSM, as there is a reduced density of samples around such threshold. The slope of the bit error probability is so shallow for the chaos-based encoding systems because there is always a number of samples arbitrarily close to the threshold that can be decoded on error, no matter how low the noise power.

The last results are drawn assuming infinite precision in the chaos-based encoding (i.e.,  $N \rightarrow \infty$  and  $D \rightarrow \infty$ ), so that the data follow the invariant theoretical pdf's. But, since we will always be in a situation with  $D < N$  and  $N < \infty$ , we can look into the effect of a finite  $D$  in the system. In fact, for the BSM with a precision of  $D$  bits, the pdf will be

$$p(x) = \sum_{i=0}^{2^D-1} \frac{1}{2^D} \delta \left( x - \frac{i}{2^D} \right). \quad (2.40)$$

This means that each of the  $2^D$  possible samples occur with probability  $1/2^D$ . For  $D$  high enough, this pdf tends to the uniform pdf within the interval  $[0, 1]$ . Taking into account Eq. (2.32), the exact probability of error in the case of simple decoding will be

$$\begin{aligned} P_b &= \frac{1}{2^D} \sum_{i=0}^{2^{D-1}-1} \frac{1}{2} \operatorname{erfc} \left( \left( \frac{1}{2} - \frac{i}{2^D} \right) \frac{1}{\sigma\sqrt{2}} \right) + \\ &+ \frac{1}{2^D} \sum_{i=2^{D-1}}^{2^D-1} \left[ 1 - \frac{1}{2} \operatorname{erfc} \left( \left( \frac{1}{2} - \frac{i}{2^D} \right) \frac{1}{\sigma\sqrt{2}} \right) \right], \end{aligned} \quad (2.41)$$

where  $\sigma^2$  is again the power of the noise. To put this expression in terms of  $E_b/N_0$ , it should be taken into account that the signal power also differs from the infinite precision case. In fact, for BSM and taking into account its real invariant pdf of Eq. (2.40), it will be

$$\sigma_x^2 = \frac{1}{12} \frac{2^{2D} + 2^{D+3} - 3}{2^{2D}}. \quad (2.42)$$

In any case, for  $\sigma^2 \rightarrow 0$  (i.e.,  $E_b/N_0 \rightarrow \infty$  for fixed signal power), the only term left will be the one with  $i = 2^{D-1}$ , which makes  $1 - (1/2) \operatorname{erfc}(\cdot) = 1/2$  with independence of the noise power, and therefore the whole  $P_b$  will tend to a constant error floor of  $1/2^{D+1}$ .

This result has to be taken into account when designing the system, since, for  $E_b/N_0$  higher than the limit value where the error floor  $P_b = 1/2^{D+1}$  becomes dominant, the performance could be substantially different from the predictions made under the assumption of a continuous and uniformly distributed pdf. In practice, however, a moderate value of  $D$  between  $D = 15$  and  $D = 20$  will be enough to overcome these possible side effects, as this pushes such error floor  $E_b/N_0$  beyond the 40 dB and the difference between the finite precision signal power and the infinite precision signal power becomes negligible. Though drawn for the BSM, similar results can be found for the MLM and the FLPM.

## 2.4.2. Decoding algorithms

We have seen that the direct decoding of the chaotic sequence offers in all the cases a much poorer behaviour than uncoded transmission, due to the fact that the squared Euclidean distance between each of the samples and the thresholds can be almost arbitrarily small. In fact, with direct decoding we neglect the potential advantage of the system, which is the redundancy present in adjoining symbols, since each symbol  $x_n$  as defined in Eqs. (2.9) and (2.10) has  $D - 1$  bits in common with  $x_{n-1}$  and  $x_{n+1}$ ,  $D - 2$  with  $x_{n-2}$  and  $x_{n+2}$ , and so on. According to this, we will propose in the following a number of decoding methods that, based upon different principles, will try to exploit this redundancy.

### Heuristic decoding

The first decoding method was already employed in [Baptista and López, 2002], and we call it here *heuristic decoding*. It takes advantage of the property of the discrete chaotic systems, where two trajectories starting from points very close could easily diverge, while, on the contrary, two trajectories iterated backwards through the map starting from two different points could finally merge [Spratt, 2003].

Therefore, to decode the symbol  $x_n$ , we look ahead  $M - 1$  received symbols,  $y_{n+1}, \dots, y_{n+M-1}$ , and rebuild the possible trajectories for the corresponding map that have as ending point  $y_{n+M-1}$ . The proposed maps have two possible solutions for the inverse problem as a function of the branch chosen,  $f_1^{-1}(x)$  and  $f_2^{-1}(x)$ , and therefore, we will have a total of  $2^{M-1}$  possible trajectories ending in  $y_{n+M-1}$ . For example, for  $M = 2$  and the Bernoulli shift map, we have

$$\begin{aligned} z_n^0 &= f_1^{-1}(y_{n+1}) = \frac{y_{n+1}}{2} \\ z_n^1 &= f_2^{-1}(y_{n+1}) = \frac{y_{n+1}}{2} + \frac{1}{2} \end{aligned} \quad (2.43)$$

Previously to this first step, the values of  $y_{n+M-1}$  have to be normalized within the interval  $[0, 1]$  for consistency reason, since the maps are defined in  $[0, 1]$ . As a consequence, any time we write  $y_{n+M-1}$  in the heuristic decoding and related algorithms, we are implicitly assuming

$$\begin{aligned} y_{n+M-1} > 1 &\rightarrow y_{n+M-1} = 1 \\ y_{n+M-1} < 0 &\rightarrow y_{n+M-1} = 0. \end{aligned} \quad (2.44)$$

In the case of the MLM and for  $M = 2$ , the inverse possible values are calculated as

$$\begin{aligned} z_n^0 &= f_1^{-1}(y_{n+1}) = \frac{1}{2} - \frac{1}{2}\sqrt{1 - y_{n+1}} \\ z_n^1 &= f_2^{-1}(y_{n+1}) = \frac{1}{2} + \frac{1}{2}\sqrt{y_{n+1}}. \end{aligned} \quad (2.45)$$

For the FLPM, and when  $y_{n+1} < 1/2$

$$\begin{aligned} z_n^0 &= f_1^{-1}(y_{n+1}) = \frac{1}{2}\sqrt{\frac{1}{2} - \frac{1}{2}\sqrt{1 - 4y_{n+1}^2}} \\ z_n^1 &= f_2^{-1}(y_{n+1}) = 1 - \frac{1}{2}\sqrt{\frac{1}{2} + \frac{1}{2}\sqrt{1 - 4y_{n+1}^2}}. \end{aligned} \quad (2.46)$$

For the FLPM, and when  $y_{n+1} \geq 1/2$

$$\begin{aligned} z_n^0 &= f_1^{-1}(y_{n+1}) = \frac{1}{2}\sqrt{\frac{1}{2} + \frac{1}{2}\sqrt{1 - 4(1 - y_{n+1})^2}} \\ z_n^1 &= f_2^{-1}(y_{n+1}) = 1 - \frac{1}{2}\sqrt{\frac{1}{2} - \frac{1}{2}\sqrt{1 - 4(1 - y_{n+1})^2}}. \end{aligned} \quad (2.47)$$

We denote the resulting filtered sequences as  $z_k^{s_l}$ , where  $k = n, \dots, n + M - 2$ ,  $l = 1, \dots, 2^{M-1}$  and  $s_l$  is a vector with length  $M - 1$  defining the  $l$ -th trajectory, starting at time  $n$  and ending at time  $n + M - 1$ . For example, we can write  $s_l = (0, 1, 1, 0, 0, \dots, 0, 1)$ , with 0 and 1 in each case identifying the inverse in the  $k$ -th position (with the meaning implicit in the Eqs. (2.43), (2.45), (2.46) and (2.47)). Then, the vector  $s_l$  contains the possible symbolic sequences associated with the corresponding map which end in  $y_{n+M-1}$ , since it stores the information about what branch of the map we take at any time, the one at  $x < 1/2$  (corresponding to  $f_1^{-1}(x)$ ), or the one at  $x > 1/2$  (corresponding to  $f_2^{-1}(x)$ ) [Schweizer and Schimming, 2001a; Schweizer and Schimming, 2001b].

We need a criterion to decide which is the best candidate sequence  $z_n^{s_l}$  over all possible  $2^{M-1}$  symbolic trajectories  $s_l$ . The most natural choice is to take the one whose trajectory is closest to the received one. That is, take sequence  $z_n^{s_{d_n}}$  which minimizes the squared Euclidean distance with the received one,

$$\sum_{k=n}^{n+M-2} (z_k^{s_{d_n}} - y_k)^2 \leq \sum_{k=n}^{n+M-2} (z_k^{s_l} - y_k)^2, \quad (2.48)$$

over all possible  $l = 1, \dots, 2^{M-1}$  different trajectories, with  $d_n \in \{1, \dots, 2^{M-1}\}$ . The received bit  $\hat{b}_n$  can then be hard decoded following

$$\begin{aligned} z_n^{s_{d_n}} < \frac{1}{2} &\rightarrow \hat{b}_{n+1} = 0 \\ z_n^{s_{d_n}} \geq \frac{1}{2} &\rightarrow \hat{b}_{n+1} = 1. \end{aligned} \quad (2.49)$$

This algorithm takes advantage of the redundancy present between  $x_n$  and the subsequent  $M - 1$  symbols, but it does not take into account the redundancy present between  $x_n$  and the rest of the sequence.

### Recursive heuristic decoding

One possible way to overcome the mentioned problem of not taking into account all the redundancy encoded in each chaotic sample is applying the heuristic algorithm recursively. To achieve this, once the sequence of best  $z_n^{s^{dn}}$  has been calculated for each  $n = 0, \dots, N - 1$  in a first step, the same heuristic decoding algorithm can be applied by replacing the received values  $y_n$  by their corresponding filtered best matching values,  $z_n^{s^{dn}}$ . Now the redundancy present in the whole sequence, from  $y_n$  to the end, is propagated backwards and we can expect to get better results. According to this description, this second algorithm will be called the *recursive heuristic* algorithm.

One can expect that these two algorithms will do reasonably well for limited noise, since, in this case, it is more probable that the received sequence  $y_n, \dots, y_{n+M-1}$  is close to  $x_n, \dots, x_{n+M-1}$ , and the sequence  $z_k^{s^{dn}}$ , with  $k = n \cdot \dots \cdot n + M - 2$  chosen as the closest to the received one, will remain closest as well to the sent one. Nevertheless, the simulation results will show that there is an error floor which depends on the decoding complexity (the length of the decoding block  $M$  and the number of iterations). This phenomenon is related to the fact that, when the decoding relies basically on the calculation of an initial condition (in this case,  $x_n$  as a function of the channel distorted values  $y_n, \dots, y_{n+M-1}$ ), a threshold effect appears [Hen and Merhav, 2004].

### Maximum likelihood Viterbi decoding

The third algorithm is a sequence decoding algorithm based on the *Maximum likelihood* (ML) criterion already proposed in [Kisel et al., 2001], but with a sliding window framework [Escribano et al., 2006b]. It is an adaptation of the well known Viterbi algorithm [Viterbi, 1967]. We will briefly review it here. This ML algorithm work on a symbolic dynamics basis [Schweizer and Schimming, 2001a; Schweizer and Schimming, 2001b], and therefore a quantization of the phase space  $[0, 1]$  is needed. In the two previous algorithms we saw that we were implicitly applying symbolic dynamics while rebuilding backwards the trajectories  $s_l$  of inverse branches chosen, because taking the  $f_1^{-1}(x)$  branch or the  $f_2^{-1}(x)$  branch implied being placed at the subinterval  $[0, \frac{1}{2})$  (encoded bit  $b = 0$ ) or  $[\frac{1}{2}, 1]$  (encoded bit  $b = 1$ ).

The present and the next algorithm need also an analogous setup with a division in subintervals, but a higher number of the same will be required to rightly capture the signal dynamics. According to this, we take the interval  $[0, 1]$  and split it into a series of non-overlapping subintervals  $I_i$  with limits  $\frac{i}{p}$  and  $\frac{i+1}{p}$  for  $i = 0, \dots, p - 1$  and centers in  $c_i = \frac{i}{p} + \frac{1}{2p}$ . These subintervals are additionally defined by

$$[0, 1] = \cup_{i=0}^{p-1} I_i, \quad (2.50)$$

$$I_i \cap I_j = \emptyset, \quad i \neq j. \quad (2.51)$$

The number of subintervals  $p$  is taken as an even number, so that the threshold point  $x = \frac{1}{2}$  is the upper limit of one subinterval and the lower one of another.

Thus, with the only knowledge of the subinterval  $I_i$  in which a sample lies, we can say whether this sample has to be decoded as a 1 or a 0. Moreover, if we replace the original sequence by the sequence of subintervals where the corresponding sample lies<sup>2</sup>, we get a symbolic representation of the sequence that can be described as a first order Markov process, with a corresponding transition matrix  $\mathbf{T}$  [Kisel et al., 2001; Papoulis and Pillai, 2002]. The term  $t_{ij}$  in this matrix means the transition probability between the subinterval  $i$  and the subinterval  $j$ . In the case of the BSM and its linear sections with slope +2, each subinterval is mapped exactly into two contiguous intervals with equal probability (which is proportional to the length of the original subinterval mapped to the destination subinterval). For example, in the case of  $p = 4$ , this transition matrix is:

$$\mathbf{T} = \frac{1}{2} \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}.$$

In the case of the MLM and the FLPM, this is not so straightforward, and the transition matrix has to be built by considering the quotient between the length of the intersection between the mapped interval  $f(I_i)$  and the destination interval  $I_j$  to the length of the image. This can be readily calculated once we know the expression for the map  $f(x)$  as

$$t_{ij} = \frac{\text{length}(f(I_i) \cap I_j)}{\text{length}(f(I_i))}, \quad (2.52)$$

where  $i = 0, \dots, p-1$  and  $j = 0, \dots, p-1$ . If the interval  $I_i$  is defined by the lower limit  $a_l^i$  and by the upper limit  $a_u^i$ , then  $\text{length}(I_i) = a_u^i - a_l^i$ .

In the decoding process, we consider the candidate sequence of subinterval centers within a window of length  $L$ ,  $d_k^i$ ,  $k = 0, \dots, L-1$ , where  $d_k^i = c_i$  if  $x_{n+k}$  lies in  $I_i$ . Moreover, we define the state of the Markov process within the window of length  $L$  as  $s_k = i$  at time  $k = 0, \dots, L-1$  if  $x_{n+k}$  lies in  $I_i$ . The starting state  $s_0$  is considered to be initially 0, but, as the sliding window proceeds with the decoding, the initial state value will be taken from the previously calculated values for each decoding block of length  $L$ , and in this way the redundancy of the whole sequence is propagated forward. To apply the Viterbi algorithm in the form described in [Kisel et al., 2001], we take a block of  $L$  received symbols,  $y_n, \dots, y_{n+L-1}$  and perform the following operations

- Initialize the algorithm:

$$\begin{aligned} \delta_0^i &= \pi_i b_i(y_n) & 0 \leq i < p \\ \phi_0^i &= 0 & 0 \leq i < p \end{aligned} \quad (2.53)$$

<sup>2</sup>In an equivalent way, the chaotic sequence can be replaced by the sequence of centers  $c_i$  of the subintervals visited.

- Perform a forward pass:

$$\delta_k^j = \max_{0 \leq i < p} \{ \delta_{k-1}^i t_{ij} b_j(y_{n+k}) \} \quad (2.54)$$

$$0 < k < L \quad 0 \leq j < p$$

$$\phi_k^j = \arg \max_{0 \leq i < p} \{ \delta_{k-1}^i t_{ij} b_j(y_{n+k}) \} \quad (2.55)$$

$$0 < k < L \quad 0 \leq j < p$$

- Terminate:

$$\hat{s}_{p-1}^n = \arg \max_{0 \leq i < p} \{ \delta_{p-1}^i \} \quad (2.56)$$

- Perform a backward pass:

$$\hat{s}_k^n = \phi_{k+1}^{\hat{s}_{k+1}^n} \quad k = L-2, \dots, 0, \quad (2.57)$$

where  $t_{ij}$  corresponds to the transition probabilities of the Markov process,  $b_j(y_k)$  corresponds to the transition probabilities of the channel<sup>3</sup>

$$b_j(y_k) = p(y_k | c_j) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(y_k - c_j)^2}{2\sigma^2}}, \quad (2.58)$$

and  $\pi_i$  is the *a priori* probability of  $x_n$  belonging to interval  $I_i$ . These probabilities can be readily calculated from the invariant density of the map as

$$\pi_i = F\left(\frac{i+1}{P}\right) - F\left(\frac{i}{P}\right), \quad (2.59)$$

where  $F(x)$  is the distribution function. The sequence  $d_k^{\hat{s}_k^n}$ ,  $k = 0, \dots, L-1$ , is the ML quantized sequence corresponding to the block of received symbols  $y_n, \dots, y_{n+L-1}$ . To proceed with the algorithm, we store  $d_0^{\hat{s}_0^n}$ , decide over

$$\hat{b}_{n+1} = \left\lfloor d_0^{\hat{s}_0^n} + \frac{1}{2} \right\rfloor, \quad (2.60)$$

and perform the same algorithm over the following block of  $L$  symbols,  $y_{n+1}, \dots, y_{n+L}$ . The main difference with the algorithm in [Kisel et al., 2001] is the application of this sliding window scheme in order to limit the memory requirements of the sequence estimation algorithm [Escribano et al., 2006b]. Note that, contrary to the heuristic decoding and the recursive heuristic decoding, we do not force the values of the received sequence  $y_n$  to belong to the  $[0, 1]$  interval. We will not do it either for the next algorithm.

<sup>3</sup>Note that the decoder requires the knowledge of the noise variance  $\sigma^2$ . This variance is easy to find in the case of an AWGN channel by standard methods and so its calculation is not addressed here [Proakis, 2001].

### Maximum a posteriori BCJR decoding

The algorithm previously described performs Maximum Likelihood Sequence Estimation (MLSE), which is not optimal for minimizing the bit error probability. Since we know the pdf's of the sent and received data, we can also resort to *Maximum A Posteriori* (MAP) sequence decoding, since the MAP framework is known to be better in lowering the final BER in communications systems [Haykin, 2001]. The possibility of MAP estimation of a chaotic sequence has already been addressed in its optimal form [Pantaleón et al., 2000; Pantaleón et al., 2002; Luengo et al., 2002; Pantaleón et al., 2003; Luengo, 2006], but the resulting algorithms are not always feasible from a practical sequence decoding point of view. We then propose our own alternative that makes use of the BCJR algorithm<sup>4</sup> derived in [Bahl et al., 1974] for the estimation of the *a posteriori* probabilities of the states and transitions of a first order Markov chain. In our case, we can adapt the algorithm in a pretty straightforward way, starting from the same definitions previously established for the case of Viterbi decoding. Therefore, we have a Markov chain with  $p$  possible states and the same transition matrix  $\mathbf{T}$  of the ML case. We also reproduce the sliding window scheme and we take a decoding block of  $L$  received symbols  $y_n, \dots, y_{n+L-1}$ . Again, we say that the state  $s_k$  at time  $k = 0, \dots, L - 1$  is  $s_k = i$  when  $x_{n+k}$  lies in the interval  $I_i$ . The trellis defined by the Markov chain is not initialized nor ended at a given state, and this has to be taken into account in the initialization steps of this MAP algorithm.

To build the algorithm, we have to calculate several probability functions ( $\gamma$ ,  $\alpha$ ,  $\beta$  and  $\lambda$ ) as stated in [Bahl et al., 1974].  $\gamma$  is defined as the transition probability at time  $n$  between two subsequent states when the received sample is  $y_n$  and follows the general expression

$$\begin{aligned} \gamma_n(i, j) &= P(s_n = j, y_n | s_{n-1} = i) = \\ &= \sum_x P(s_n = j | s_{n-1} = i) P(x_n = x | s_{n-1} = i, s_n = j) p(y_n | x). \end{aligned} \quad (2.61)$$

As we operate with the quantized symbolic sequence, we replace again the possible values of  $x_n$  by the values of the center of the intervals  $c_j$ . So, in a transition from  $s_{n-1} = i$  to  $s_n = j$ , the only possible quantized chaotic output is  $c_j$  regardless of  $s_{n-1}$ , and  $P(x_n = x | s_{n-1} = i, s_n = j) = 1$  when  $x_n \in I_j$  and is 0 in the rest of cases.

$P(s_n = j | s_{n-1} = i)$  is the transition probability  $t_{ij}$  and  $p(y_n | c_j) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(y_n - c_j)^2}{2\sigma^2}}$  is the channel output probability. Once we have defined  $\gamma$ , the evaluation of the rest of probability functions is straightforward following [Bahl et al., 1974]. Taking into account that we apply the algorithm in a sliding window scheme, it consists in the following steps

---

<sup>4</sup>Named after the initials of the authors, Bahl, Cocke, Jenilek and Raviv.

- Calculate the probability function:

$$\gamma_k^n(i, j) = P(s_k = j, y_{n+k} | s_{k-1} = i) = t_{ij} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(y_{n+k} - c_j)^2}{2\sigma^2}} \quad (2.62)$$

$$k = 1, \dots, L \quad i, j = 0, \dots, p-1$$

- Initialize and forward calculate the probability function:

$$\alpha_0^n(j) = \lambda_1^{n-1}(j) \quad (2.63)$$

$$\alpha_k^n(j) = P(s_k = j, (y_n, \dots, y_{n+k-1})) = \sum_i \alpha_{k-1}^n(i) \gamma_k^n(i, j) \quad (2.64)$$

$$k = 1, \dots, L \quad i, j = 0, \dots, p-1$$

- Initialize and backward calculate the probability function:

$$\beta_L^n(j) = \alpha_L^n(j) \quad (2.65)$$

$$\beta_k^n(j) = P((y_{n+k} \dots y_{n+L-1}) | s_k = j) = \sum_i \beta_{k+1}^n(i) \gamma_{k+1}^n(j, i) \quad (2.66)$$

$$k = L-1, \dots, 0 \quad i, j = 0, \dots, p-1$$

- Finally compute the *a posteriori* probabilities:

$$\lambda_k^n(i) = P(s_k = i, (y_n \dots y_{n+L-1})) = \alpha_k^n(i) \beta_k^n(i) \quad (2.67)$$

$$k = 1, \dots, L \quad i = 1, \dots, p$$

To decode the bit at time  $n+1$ ,  $n = 0, \dots, N-1$ , we take the state  $i_{max}$  that maximizes the probability  $\lambda_1^n(i)$ , and decode according to  $\hat{b}_{n+1} = \lfloor c_{i_{max}} + \frac{1}{2} \rfloor$ , where  $c_{i_{max}}$  is the center of the interval  $I_{i_{max}}$ . Note that, in the initializing step for  $\alpha$ , we take the resulting *a posteriori* values  $\lambda$  from the preceding decoding block. For  $y_0$ , we initialize equiprobably  $\alpha_0^1(i) = \frac{1}{p}$ ,  $i = 0 \dots p-1$ , taking into account that no previous evidence can be used in this case. The algorithm proceeds in the same way by taking the next block of received symbols  $y_{n+1}, \dots, y_{n+L}$  in the sliding window scheme.

## 2.5. Performance analysis

Finding out possible performance bounds for the bit error probability in the cases of the four decoding methods proposed could be a very difficult task. Nevertheless, the simple analysis of the heuristic decoding when  $M = 2$  symbols will allow us to gain some insight on the encoding and decoding processes. For higher values of  $M$ , the theoretical analysis becomes quickly unfeasible. When  $M = 2$ , we can consider basically three cases

- $y_{n+1} < 0$

Recall that this value is normalized as  $y_{n+1} = 0$ . In this case,  $z_n^0 = 0$  and  $z_n^1 = 1/2$ , and we get

$$P_{b1} = P\left(y_{n+1} < 0, (z_n^1 - y_n)^2 < (z_n^0 - y_n)^2, x_n < \frac{1}{2}\right) + P\left(y_{n+1} < 0, (z_n^0 - y_n)^2 < (z_n^1 - y_n)^2, x_n \geq \frac{1}{2}\right) \quad (2.68)$$

- $y_{n+1} > 1$

Recall that this value is normalized as  $y_{n+1} = 1$ . Therefore,  $z_n^0 = 1/2$  and  $z_n^1 = 1$  and the decoded symbol will always be  $b_{n+1} = 1$ , so

$$P_{b2} = P\left(y_{n+1} > 1, x_n < \frac{1}{2}\right) \quad (2.69)$$

- $0 \leq y_{n+1} \leq 1$

Now, no restriction or normalization is applied, and

$$P_{b3} = P\left(0 \leq y_{n+1} \leq 1, (z_n^1 - y_n)^2 < (z_n^0 - y_n)^2, x_n < \frac{1}{2}\right) + P\left(0 \leq y_{n+1} \leq 1, (z_n^0 - y_n)^2 < (z_n^1 - y_n)^2, x_n \geq \frac{1}{2}\right) \quad (2.70)$$

Knowing that, in the BSM case,  $y_{n+1} = x_{n+1} + n_{n+1} = f(x_n) + n_{n+1}$ ,  $z_n^0 = y_{n+1}/2$ ,  $z_n^1 = y_{n+1}/2 + 1/2$ ,  $y_n = x_n + n_n$ , and that  $n_n$  and  $n_{n+1}$  are independent and identically distributed, we arrive at the following result after some algebra

- $y_{n+1} < 0$

$$P_{b1}^B = \frac{1}{4} \int_0^{\frac{1}{2}} \operatorname{erfc}\left(2x\sqrt{12\frac{E_b}{N_0}}\right) \operatorname{erfc}\left(\left(\frac{1}{4} - x\right)\sqrt{12\frac{E_b}{N_0}}\right) dx + \frac{1}{2} \int_{\frac{1}{2}}^1 \operatorname{erfc}\left(2x\sqrt{12\frac{E_b}{N_0}}\right) \left[1 - \frac{1}{2} \operatorname{erfc}\left(\left(\frac{1}{4} - x\right)\sqrt{12\frac{E_b}{N_0}}\right)\right] dx \quad (2.71)$$

- $y_{n+1} > 1$

$$P_{b2}^B = \frac{1}{4} \operatorname{erfc}\left(\sqrt{12\frac{E_b}{N_0}}\right) + \frac{1}{2\sqrt{48\pi\frac{E_b}{N_0}}} \left(1 - e^{-12\frac{E_b}{N_0}}\right) \quad (2.72)$$

- $0 \leq y_{n+1} \leq 1$

$$\begin{aligned}
P_{b3}^B &= \frac{1}{\sqrt{48\pi\frac{E_b}{N_0}}} \int_0^1 \left(\frac{1}{2} - \frac{x}{2}\right) \left[ \operatorname{erfc} \left( \left(\frac{1}{4} - \frac{x}{2}\right) \sqrt{12\frac{E_b}{N_0}} \right) + \right. \\
&\quad \left. + \operatorname{erfc} \left( \left(\frac{1}{4} + \frac{x}{2}\right) \sqrt{12\frac{E_b}{N_0}} \right) \right] dx \tag{2.73}
\end{aligned}$$

The total bit error error probability will be  $P_b^B = P_{b1}^B + P_{b2}^B + P_{b3}^B$ .

We have also addressed this problem with the MLM system. Taking into account its own  $p(x)$  and the corresponding expressions for  $z_n^0$  and  $z_n^1$ , we arrive at

- $y_{n+1} < 0$

$$\begin{aligned}
P_{b1}^M &= \frac{1}{2} \int_0^{\frac{1}{2}} \frac{1}{\pi\sqrt{x(1-x)}} \operatorname{erfc} \left( 2 \left(\frac{1}{4} - x\right) \sqrt{2\frac{E_b}{N_0}} \right) \cdot \\
&\quad \cdot \left[ 1 - \frac{1}{2} \operatorname{erfc} \left( 2(-4x(1-x)) \sqrt{2\frac{E_b}{N_0}} \right) \right] dx + \\
&+ \int_{\frac{1}{2}}^1 \frac{1}{\pi\sqrt{x(1-x)}} \left[ 1 - \frac{1}{2} \operatorname{erfc} \left( 2 \left(\frac{1}{4} - x\right) \sqrt{2\frac{E_b}{N_0}} \right) \right] \cdot \\
&\quad \cdot \left[ 1 - \frac{1}{2} \operatorname{erfc} \left( 2(-4x(1-x)) \sqrt{2\frac{E_b}{N_0}} \right) \right] dx \tag{2.74}
\end{aligned}$$

- $y_{n+1} > 1$

$$P_{b2}^M = \frac{1}{2} \int_0^{\frac{1}{2}} \frac{1}{\pi\sqrt{x(1-x)}} \operatorname{erfc} \left( 2(1 - 4x(1-x)) \sqrt{2\frac{E_b}{N_0}} \right) dx \tag{2.75}$$

- $0 \leq y_{n+1} \leq 1$

$$\begin{aligned}
P_{b3}^M &= \int_0^{\frac{1}{2}} p(x) \int_{-h(x)}^{1-h(x)} \sqrt{\frac{2 E_b}{\pi N_0}} e^{8 \frac{E_b}{N_0} z^2} \cdot \\
&\cdot \operatorname{erfc} \left( \left( 1 + \frac{1}{2} \sqrt{h(x) + z} - \frac{1}{2} \sqrt{1 - h(x) - z} - x \right) \sqrt{2 \frac{E_b}{N_0}} \right) dz dx + \\
&\quad + \int_{\frac{1}{2}}^1 p(x) \int_{-1+h(x)}^{h(x)} \sqrt{\frac{2 E_b}{\pi N_0}} e^{8 \frac{E_b}{N_0} z^2} \cdot \\
&\cdot \left[ 2 - \operatorname{erfc} \left( \left( 1 + \frac{1}{2} \sqrt{h(x) + z} - \frac{1}{2} \sqrt{1 - h(x) - z} - x \right) \sqrt{2 \frac{E_b}{N_0}} \right) \right] dz dx \\
p(x) &= \frac{1}{\pi \sqrt{x(1-x)}} \quad h(x) = 4x(1-x) \tag{2.77}
\end{aligned}$$

Again, the total bit error probability is  $P_b^M = P_{b1}^M + P_{b2}^M + P_{b3}^M$ . All the equations drawn for the BSM and MLM cases, with the exception of Eq. (2.69), require numerical integration [Abramowitz and Stegun, 1965].

On the other hand, in the case of the ML decoding algorithm, an MLSE estimation of the bit error probability based on the minimum squared Euclidean distance between all the possible encoded sequences could be given as a good approximation when  $E_b/N_0 \rightarrow \infty$ . This was done for the BSM and the tent map in [Kozic et al., 2003b]. However, it was also shown there that finding this minimum squared Euclidean distance is not an easy task, either for the finite precision or the infinite precision chaos-based encoding systems. Therefore, we recall here only that a tight upper bound for the bit error probability of the BSM encoding system under the assumption of MLSE decoding is

$$P_b^B \leq \operatorname{erfc} \left( \sqrt{\frac{E_b}{N_0}} \right). \tag{2.78}$$

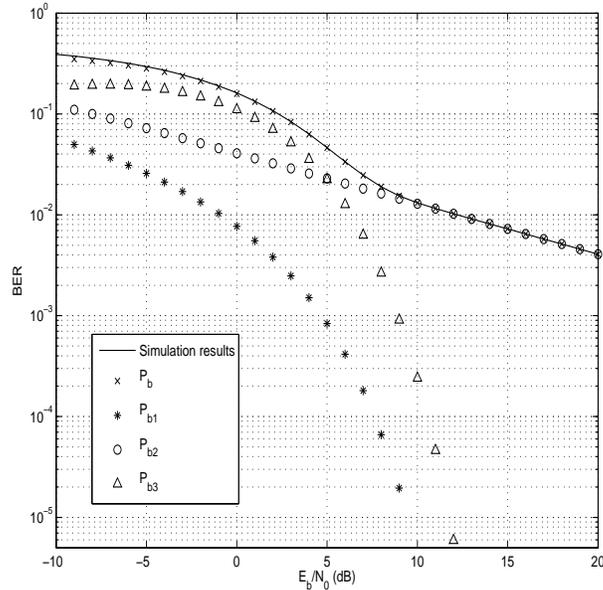
Together with BPSK, we will plot for comparison purposes the bit error probability of the well known differential chaos shift keying (DCSK) system, whose expression is [Abel and Schwarz, 2002]

$$P_b^{\text{DCSK}} = \frac{1}{2} \operatorname{erfc} \left( \sqrt{4 \left( \frac{E_b}{N_0} \right)^{-1} + 4L \left( \frac{E_b}{N_0} \right)^{-2}} \right), \tag{2.79}$$

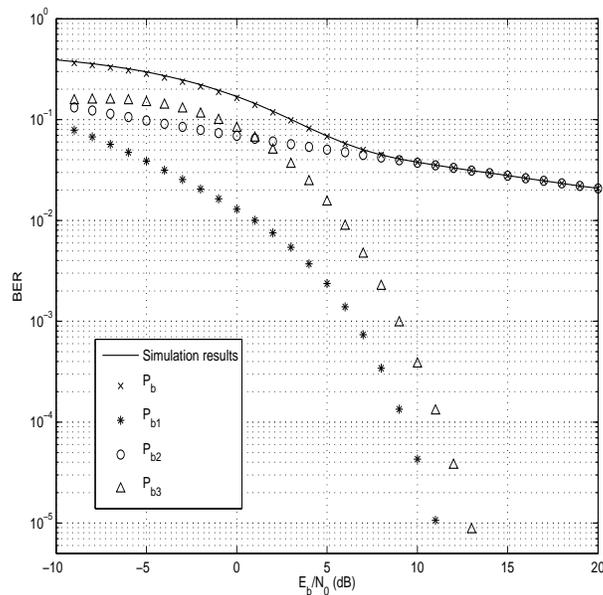
where  $L$ , called spreading factor, is the number of chaotic samples sent to the channel for each input information bit.

## 2.6. Simulation results

In all cases, except for Fig. 2.18, we have chosen  $D = 20$  bits per symbol. This guarantees that  $2^{-D}$  is small enough to make the difference between  $x_n$  and



**Figure 2.8.** Theoretical bit error probability and simulation results for the BSM encoder when performing heuristic decoding with  $M = 2$ .

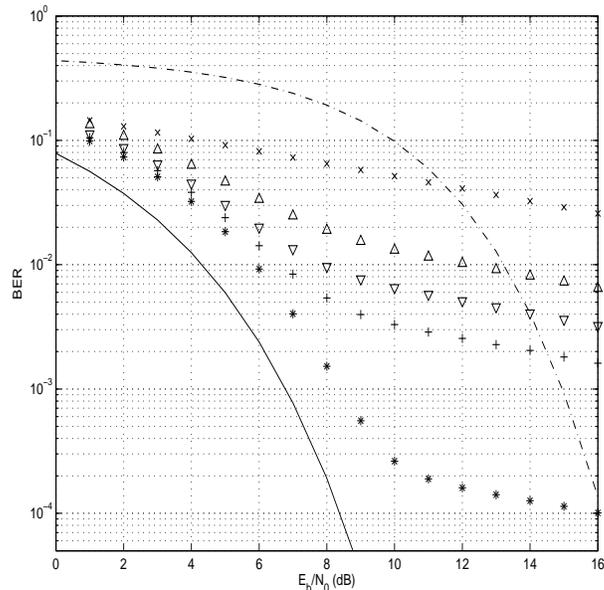


**Figure 2.9.** Theoretical bit error probability and simulation results for the MLM encoder when performing heuristic decoding with  $M = 2$ .

$x'_n$  negligible in practice, while making it possible to use the encoding scheme with an arbitrary block size  $N$ , as a value  $D \approx N$  would make the truncation process useless. In any case, this condition is always met, since we have performed all the simulations with a data block size of  $N = 100000$ . For comparison, we plot in several of the following diagrams the bit error probability of BPSK [Proakis, 2001], and of

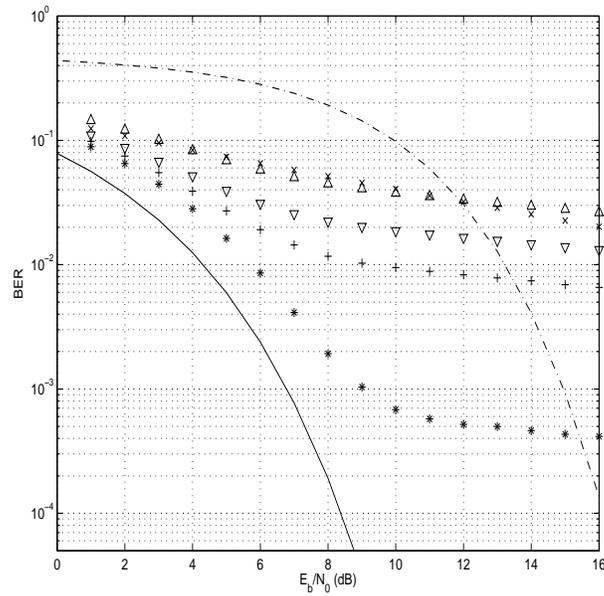
DCSK with  $L = 20$  [Abel and Schwarz, 2002; Lau and Tse, 2003]. This means that 20 chaotic symbols are produced for each information bit  $b_n$ . We will verify that, in most of the cases, the performance of the chaotic systems proposed remains between the bit error probability curves of BPSK and DCSK.

In Figs. 2.8 and 2.9, we have depicted the simulation results for the BSM and MLM systems when decoding with the heuristic algorithm with  $M = 2$ , together with the bounds drawn in the preceding section. We also show the individual contributions to the bit error probability,  $P_{b1}$ ,  $P_{b2}$  and  $P_{b3}$ . As it may be seen, Eqs. (2.72) and (2.75) give reason of the behaviour of the system as  $E_b/N_0 \rightarrow \infty$ , and, due to their polynomial dependence on  $E_b/N_0$ , they are responsible of the residual BER that remains for large signal to noise ratios. The reason for this is that  $y_{n+1} > 1$  leads to inverse values  $1/2$  and  $1$  as stated, both values decoded as  $\hat{b}_{n+1} = 1$  and this will always lead to an error if  $x_n < 1/2$  was sent. The other terms of Eqs. (2.71) and (2.74) affect the performance only in the region of lower  $E_b/N_0$ . For example, in the BSM system,  $y_{n+1} < 0$  leads to the inverse values  $1/2$  and  $0$ , and so, for limited noise, if  $x_n < 1/2$  and  $y_{n+1} \cong 2x_n + n_{n+1} < 0$ , it is most probable that  $x_n$  is near  $0$ ,  $(z_n^0 - y_n)^2$  is also near  $0$  and the symbol is decoded successfully with high probability. The terms of Eqs. (2.73) and (2.76) are dominant in the range of middle  $E_b/N_0$ , and also drop fast as the  $E_b/N_0$  grows, implying that, for a decreasing noise power, the decoding process is able to decode the symbol properly.

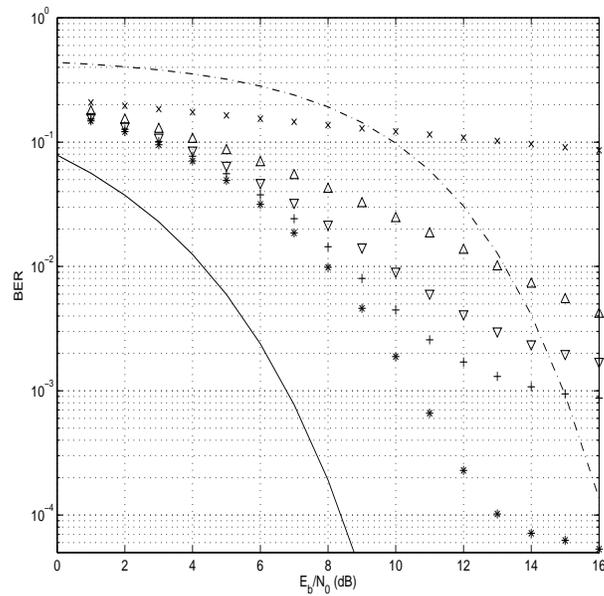


**Figure 2.10.** BER for the BSM encoder in the case of heuristic decoding, with  $D = 20$  bits and different  $M$ . 'x':  $M = 1$ . 'Δ':  $M = 2$ . '▽':  $M = 3$ . '+':  $M = 4$ . '\*':  $M = 8$ . The performance of BPSK (continuous line) and DCSK (dash-dotted line) with  $L = 20$  are shown for comparison.

If we look into Figs. 2.10, 2.11 and 2.12, where we show the results for the heuristic decoding with different  $M$ , we appreciate two different regions in the resulting

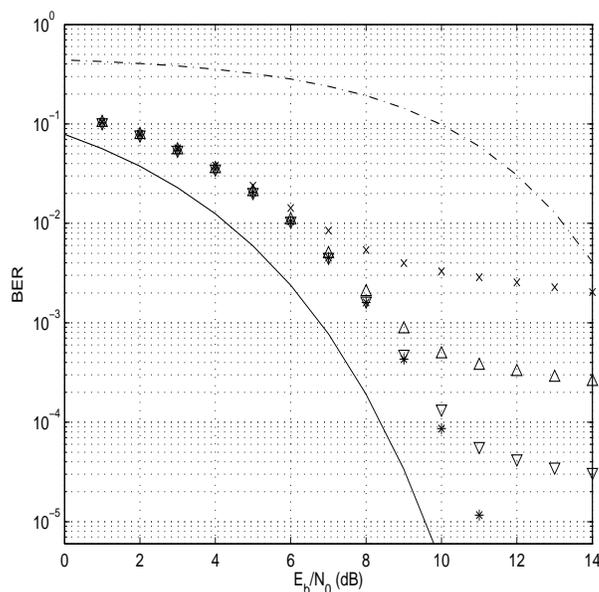


**Figure 2.11.** BER for the MLM encoder in the case of heuristic decoding, with  $D = 20$  bits and different  $M$ . 'x':  $M = 1$ . ' $\Delta$ ':  $M = 2$ . ' $\nabla$ ':  $M = 3$ . '+':  $M = 4$ . '\*':  $M = 8$ . The performance of BPSK (continuous line) and DCSK (dash-dotted line) with  $L = 20$  are shown for comparison.



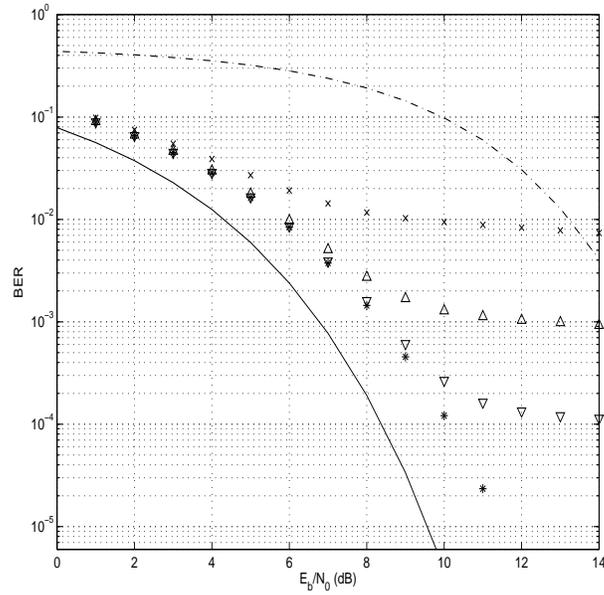
**Figure 2.12.** BER for the FLPM encoder in the case of heuristic decoding, with  $D = 20$  bits and different  $M$ . 'x':  $M = 1$ . ' $\Delta$ ':  $M = 2$ . ' $\nabla$ ':  $M = 3$ . '+':  $M = 4$ . '\*':  $M = 8$ . The performance of BPSK (continuous line) and DCSK (dash-dotted line) with  $L = 20$  are shown for comparison.

BER graphs for the three encoding maps: one waterfall region, tending to a curve some dB away from the BPSK case, and one floor region, whose value becomes lower with increasing  $M$ , thus establishing a tradeoff between decoding complexity and desired BER. The main reason for this improvement is that the samples can be correctly decoded with higher probability as the number of consecutive samples taken into account grows, so that there is more information available to decide over the symbol and the redundancy is employed more efficiently. Going back again to Figs. 2.8 and 2.9, we verify that it is the term  $P_{b2}$  the most affected by the increasing of  $M$ . Extrapolating from the  $M = 2$  case, we can think that the reason for this error floor is associated with the reference samples  $y_{n+M-1}$  lying outside the interval  $[0, 1]$ . We can deduce thus that the increase in decoding complexity by increasing  $M$  and the using of a higher quantity of redundancy to decode each symbol can compensate for the errors induced when the reference symbol lies outside the interval  $[0, 1]$ . As a consequence of this, the value of the error floor decreases. On the other hand, the performance in the waterfall region does not improve with increasing  $M$ , but tends to a curve very close to  $P_{b3}$  for the  $M = 2$  case, which is related to the samples lying inside  $[0, 1]$ . Therefore, the errors induced by samples with reference within  $[0, 1]$  seem to be unaffected by the availability of extra redundancy when  $M > 2$ .

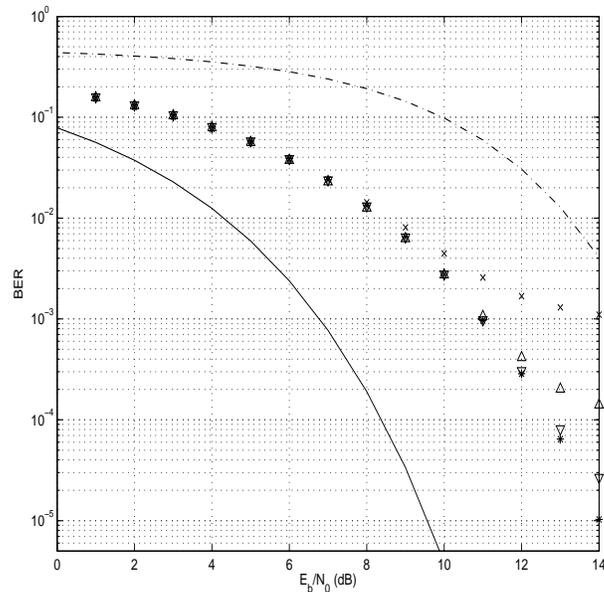


**Figure 2.13.** BER for the BSM encoder in the case of recursive heuristic decoding, with  $D = 20$  bits,  $M = 4$  symbols and different number of iterations. 'x': 1 iteration. 'Δ': 2 iterations. '▽': 3 iterations. '\*': 7 iterations. The performance of BPSK (continuous line) and DCSK (dash-dotted line) with  $L = 20$  are shown for comparison.

In Figs. 2.13, 2.14 and 2.15, we can see the results obtained when using the recursive heuristic scheme. In this case, with a fixed complexity ( $D = 20$ ,  $M = 4$ ), we obtain better results than in the former case: the floor region falls down very

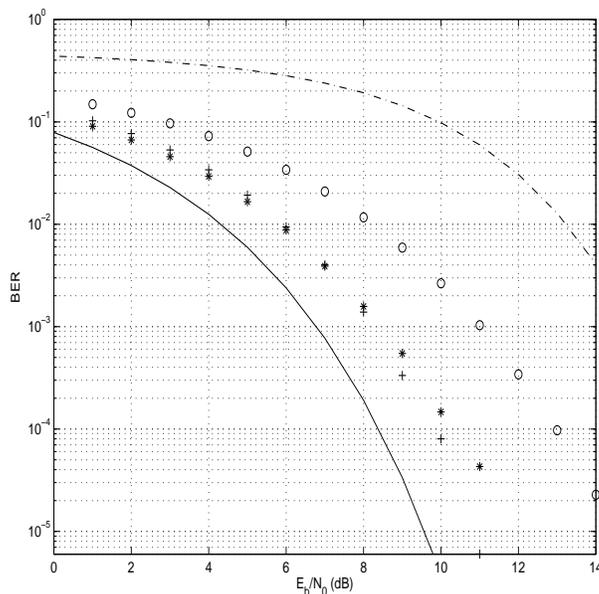


**Figure 2.14.** BER for the MLM encoder in the case of recursive heuristic decoding, with  $D = 20$  bits,  $M = 4$  symbols and different number of iterations. 'x': 1 iteration. 'Δ': 2 iterations. '▽': 3 iterations. '\*': 7 iterations. The performance of BPSK (continuous line) and DCSK (dash-dotted line) with  $L = 20$  are shown for comparison.



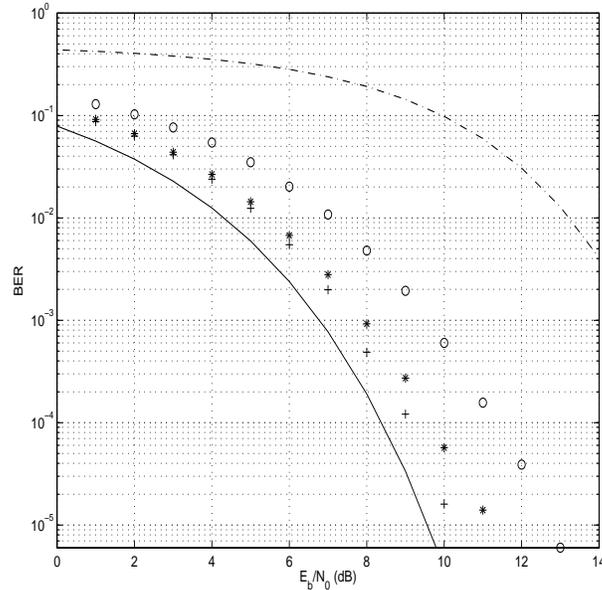
**Figure 2.15.** BER for the FLPM encoder in the case of recursive heuristic decoding, with  $D = 20$  bits,  $M = 4$  symbols and different number of iterations. 'x': 1 iteration. 'Δ': 2 iterations. '▽': 3 iterations. '\*': 7 iterations. The performance of BPSK (continuous line) and DCSK (dash-dotted line) with  $L = 20$  are shown for comparison.

fast with each iteration, achieving the bounding curve some dB away from the BPSK BER in 7 iterations, at least to a simulated BER of  $10^{-5}$ . This is a great improvement, since the complexity of the decoder changes from  $O(2^M)$  to  $O(q2^M)$ , with  $q$  meaning the number of iterations, and the recursive algorithm performs well with moderate values of  $M$ . Therefore, we can reach the same performance as in the heuristic decoding case, but at a lower complexity, just by performing a number of iterations that only increase the processing time linearly, instead of exponentially. The recursive alternative is then the only way to make heuristic decoding potentially applicable in a real system.



**Figure 2.16.** BER in the case of ML Viterbi decoding, with  $p = 32$  states and  $L = 10$  symbols. '+' : BSM encoder. '\*' : MLM encoder. 'o' : FLPM encoder. The performance of BPSK (continuous line) and DCSK (dash-dotted line) with  $L = 20$  are shown for comparison.

In the case of Viterbi decoding (Fig. 2.16), we can see that, for  $p = 32$ ,  $L = 10$ , the results are practically the same as those obtained in the recursive heuristic decoding case with 7 iterations and  $M = 4$  (i.e., without error floor region down to  $10^{-5}$ ). With the heuristic algorithms, we saw that we can achieve the performance given by the waterfall curve at the expense of a higher  $M$  or a higher number of iterations, and the results will be always affected by the error floor, even when it could be low enough. However, the Viterbi decoding converges to the limit performance of the waterfall curve from the start. The BCJR decoding results shown in Fig. 2.17 lead to an even closer performance to the BPSK BER, practically coinciding with the expected limit performance calculated through the minimum squared Euclidean distance analysis in the BSM case [Kozic et al., 2003a; Kozic et al., 2003b]. This bound was given in Eq. (2.78) and it is two times the expected bit error probability of BPSK. In fact, the MAP algorithm proposed is near to the optimum for the type

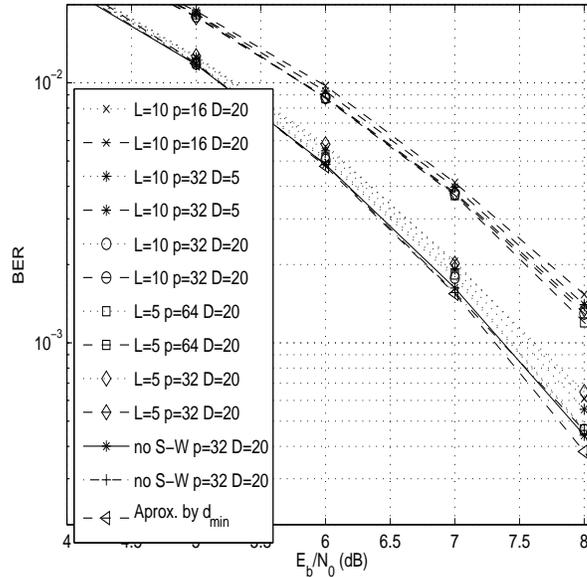


**Figure 2.17.** BER in the case of MAP BCJR decoding, with  $p = 32$  states and  $L = 10$  symbols. '+' : BSM encoder. '\*' : MLM encoder. 'o' : FLPM encoder. The performance of BPSK (continuous line) and DCSK (dash-dotted line) with  $L = 20$  are shown for comparison.

of encoding used here and sheds light on how to improve the decoding systems for the more efficient encoding schemes recently proposed [Kozic et al., 2003a; Kozic et al., 2003b; Kozic, 2006]. The complexity of the Viterbi and BCJR algorithms grows as  $O(Lp)$ , with  $p$  being a power of 2, and it exhibits the same behaviour as the recursive heuristic algorithm, with a linear and an exponential term. Nevertheless, the ML method is more complex than the heuristic alternatives, and the MAP one is even more complex than the ML, in the sense of the number of operations (sums and multiplications) needed for a comparable set of parameters.

After examining all the results, we see that the best performance is obtained with the BSM system, and not by the MLM, even though with simple decoding it was the opposite. The FLPM system, as expected, is the worst of the three. This means that the pdf as design factor for piecewise nonlinear encoding maps is not a good choice, and that the dynamics of the map is the most important factor for the final performance, regardless of the distribution of the data. The results also point to the better suitability of the piecewise linear maps in the AWGN channel.

In Fig. 2.18 we explore further the behaviour of the ML Viterbi decoding and the MAP BCJR decoding algorithms [Escribano et al., 2006b]. We illustrate only the case of the BSM system with different combinations of  $D$ ,  $L$  and  $p$ , and even the case of decoding a whole block of data without the sliding window framework. We can see that the improvement achieved by increasing  $p$  or  $L$  is in general not very high. The results show almost a constant gap between the Viterbi and the BCJR alternatives in all the cases, though the MAP algorithm is more affected by



**Figure 2.18.** *Case study of BER with MAP BCJR decoding vs ML Viterbi decoding for the BSM encoder.*

changes in the parameters. We see also that the MAP algorithm can be efficient in approaching the limit performance of BSM, at the expense only of increasing  $L$  with  $p$  fixed or viceversa. Both for the Viterbi and the BCJR algorithms, the non-sliding window results are the same and keep closest to the limit of the performance attainable, so that in this case the gap between the MAP and the ML results is filled and we can conclude that the sliding window device imposes an almost fixed penalty to the performance of the ML decoding. Note finally how the case when the quantization in the encoding process matches the symbolic representation in the decoding process<sup>5</sup> does not offer better results than the case where the chaotic sequence is closer to the real one ( $D = 20$ ), specially with MAP decoding.

## 2.7. Concluding remarks

We have described a communications system that uses chaos-based encoding and decoding blocks, and we have shown a method to design, under some constraints, an encoding chaotic map starting with the desired invariant density. We have also reviewed four possible decoding algorithms suitable for the chaos-based encoding proposed. Using these decoding algorithms, we have tried to gain insight into the behaviour of the system, its performance and possibilities.

The maps proposed, with different properties respecting their dynamics and invariant density, have not proved to be good performing, at least in the sense of behaving better than uncoded BPSK. This is a well established fact for the BSM encoding system [Kozic, 2006]. Nevertheless, even the map with the worst

<sup>5</sup>That is to say, when  $D = 5$  and  $p = 32$ .

bit error probability for simple decoding, the FLPM, was able to perform better than other standard chaotic based systems [Abel and Schwarz, 2002; Lau and Tse, 2003; Xiaofeng et al., 2004]. Nowadays, there exist a number of techniques able to provide good coding gains with chaos-based communications systems similar to the one presented here [Schimming and Hasler, 2002; Escribano et al., 2006c; Kozic et al., 2006], but the developments of this chapter serve us to gain an invaluable insight into the working-out and possibilities of communicating binary data with discrete chaotic maps.

The design task accomplished had as objective to verify how important can be the map dynamics face to the invariant density, and how such factors, when considered together, could affect the BER on the decoder side. Though the invariant density of the BSM concentrates more data in the region of the deciding threshold, this does not degrade the performance with respect to the MLM system, which, respecting the pdf shape, could be seen as a case closer to BPSK than the BSM system. This means clearly that the implicit dynamics of the maps involved is more important for decoding than the pdf of the encoded sequence, and that the piecewise linear dynamics offers advantages over the piecewise nonlinear dynamics at least in a channel like the AWGN one. The FLPM exhibits a BER in the case of direct decoding (see Fig. 2.7) worse than the other two cases, and this property holds when decoding the sequence with any of the algorithms proposed. This means that the concentration of data around the threshold point imposes a severe penalty that cannot be easily overcome. In some way, we can say that the pdf can account for the general behaviour (a pdf with more data around the critical point leads surely to worse results), but not for the details (a uniform pdf can offer better results than a pdf designed to avoid the accumulation of data around the critical point).

With respect to the decoding algorithms proposed, we can draw two important consequences that should not be overlooked in this kind of chaos-based communications:

1. The received samples which, after the addition of the Gaussian noise, lie outside the definition interval, play a major role in the final BER performance when only a limited number of samples is involved in the decoding of each of the bits, and
2. The best possible performance for a given chaos-based encoding is only attainable if the evidence stored in the whole sequence is effectively used and propagated during the decoding process, thus helping to minimize the effect of the outlying samples.

With the heuristic decoding there was a tradeoff between complexity and error floor not easy to manage, while the recursive variant, and the ML and MAP decoding algorithms offered lower complexity alternatives leading to much better results, since they were able to propagate the evidence more efficiently. We have also shown that MAP decoding, though more time consuming, leads to better results than ML decoding when using a sliding window framework, and this gives us a clue that a

---

possible concatenation of chaotic systems with iterative MAP decoding, as is done in turbocodes or in serial concatenation of channel codes, can possibly improve the behaviour of chaos-based communications systems. Chapters 4 and 5 will show the accurateness of this hint.



## Chapter 3

# Chaos Coded Modulations over Dispersive Channels

### 3.1. Introduction

We have seen in Chapter 2 that encoding binary data with antisymmetric piecewise nonlinear maps cannot do better than the simple system based on piecewise linear BSM, at least when we choose as design criterion only the invariant density of the data. The results pointed towards a better behaviour of the piecewise linear systems in the AWGN channel. Recent work has focused on a kind of chaotic encoding stemming from the description of piecewise linear maps in terms of a related trellis coded modulation (TCM), and has provided a generalization of this framework to chaotic encoding systems without the need of a proper discrete map equivalence [Schimming and Hasler, 2002; Kozic et al., 2003a; Kozic et al., 2006]. Moreover, these systems have proved to be more efficient than uncoded BPSK and exhibit BER performances comparable to TCM. The key to this success has been the approaching of the fields of digital communications and Chaos Theory, so that nowadays there is a number of tools to study and design chaos-based encoding systems suitable for potential applications in communications systems.

We then drop the possibility to look deeper into the piecewise nonlinear maps and we will proceed to study this new kind of chaos-based encoding systems, which are based initially in a chaotic piecewise linear map driven (or controlled) by small perturbations and whose general principles are close to the encoding rules of the BSM system seen in the last chapter. These systems are generally designed as chaos coded modulations (CCM) [Kozic, 2006]. Together with the review of this class of chaos-based encoders, the main task we are interested in addressing here is whether the simplest CCM (the one based on the BSM) can offer still advantages in some kind of channel different from the AWGN channel. The results of Chapter 2 suggested that the redundancy contained in the BSM chaotic sequence cannot be fully exploited in AWGN. The coded modulated characteristics of the chaotic signal give the clue that maybe this redundancy can be useful face to dispersive environments, as it happens with TCM [Anderson and Svensson, 2003].

According to this, this chapter deals with the study of the BSM CCM system in two different kinds of dispersive channels. The first one is a time invariant and

frequency selective channel where the main source of distortion is intersymbol interference (ISI) modeled by means of a filter [Proakis, 2001]. The second one is a time varying and frequency non-selective channel where the main source of distortion is flat fading [Haykin, 2001]. The analysis performed and the simulation results will highlight the fact that the lack of good results in the AWGN channel is not enough reason to neglect the chaos-based encoding systems, and that the commonplace of the suitability of chaotic signals in dispersive channels can in fact become true with a good scheme for coding and decoding.

With these goals in view, this chapter is structured as follows. Section 3.2 is devoted to the description of the CCM system in its simplest form, and to the derivation of a decoder suited to this sort of chaotic encoding. Section 3.3 describes the ISI channel, addresses the analysis of the bit error probability of the BSM CCM system, and shows by means of some simulation results the potential coding gain achievable in this kind of channel. Section 3.4 reproduces the structure of Section 3.3, but this time with the flat fading channel. Finally, Section 3.5 is devoted to the conclusions.

## 3.2. System model

In the previous chapter, we reviewed a method for encoding binary data using chaotic maps based on the discretization of the chaotic sequence generated by the BSM and using the conjugacy properties between the BSM and the class of one-dimensional antisymmetric maps. We also used the invariant pdf of the chaotic samples to design maps and to define the conjugating function needed to set a practical encoding scheme. We have seen how the information was carried in the associated symbolic sequence, so that the bit of interest could be exactly recovered in absence of distortion or noise.

We have also reviewed and proposed some decoding methods. The best ones were those which involved the whole chaotic sequence in the decoding process and which were based on the principles of ML or MAP sequence estimation. This was possible because, at the decoder side, after discretizing the definition interval of the encoding chaotic map, the chaotic sequence could be substituted by the sequence of intervals visited (i.e., its symbolic sequence), and this sequence could be finally seen as a Markov process with a finite number of states and a series of binary transitions determined by the information bit sequence.

This point of view allowed the uncoupling of the discretization level at the encoding side and at the decoding side. In this way, at the encoder it could be set as high as possible to keep the discretized chaotic sequence closest to the real one, and as low as possible at the decoder to reduce the complexity of the decoding algorithm. This helped to overcome some side effects which appeared with the decoding algorithms based on the estimation of the initial condition when the quantization level in the encoder was not high enough.

Nevertheless, for the decoding methods based on ML or MAP sequence estimation, there were no such side effects nor such drawbacks, and there is no need to

keep the difference between the quantization level at the encoder and at the decoder. So, from this point onwards, we will use an encoding method which requires a decoding algorithm matched to the same discretization level and whose extension and generalization to multidimensional systems has provided good BER results even in the AWGN channel. It also has the advantage of being closely related to the well known TCM systems [Ungerboeck, 1982; Kozic, 2006], and thus can offer the good properties of combined coded and modulation systems [Benedetto et al., 1988]. In the next subsections, we review briefly this encoding method and propose a related MAP decoding module which will allow us to derive in the following chapters new interesting developments with potential application in chaos-based communications.

### 3.2.1. Encoder

The encoder, as described in terms of its *map view* [Kozic et al., 2003b], is a chaos-based encoder driven by small perturbations [Schimming et al., 2002]. It follows the recursion

$$z_n = f(z_{n-1}) + b_n \cdot 2^{-Q}, \quad (3.1)$$

$$x_n = 2z_n - 1, \quad (3.2)$$

where  $Q$  is a natural number which we will see is related to the quantization level of the chaotic signal, and  $f(z) : [0, 1] \rightarrow [0, 1]$  is a chaotic map, which in our examples will belong to the subset of the piecewise linear maps (PWLM) where the linear sections have always slope 2 or  $-2$  [Sprott, 2003]. Throughout this chapter, the map employed will be the BSM,  $f(z) = 2z \bmod 1$ . Note that under these conditions the small perturbation manifests itself after  $Q - 1$  iterations. As seen in Eq. (3.2), the chaotic encoded signal sent to the channel  $x_n$  is defined in the  $[-1, 1]$  interval for symmetry reasons. It is easy to show that a recursion like the one in Eq. (3.1) leaves the finite set

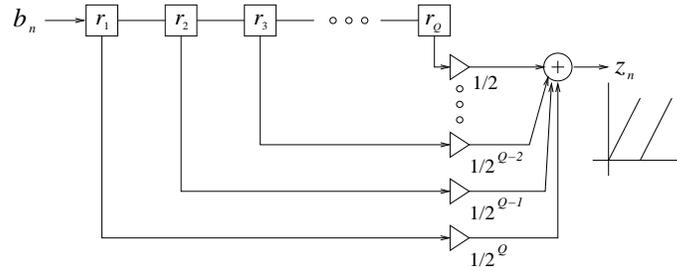
$$S_Q = \{i \cdot 2^{-Q} | i = 0, \dots, 2^Q - 1\} \quad (3.3)$$

invariant and, therefore, we can restrict Eq. (3.1) to  $S_Q$  taking as initial condition  $z_0$  any point in  $S_Q$  (e.g.  $z_0 = 0$ ) [Kozic et al., 2006]. In this way, we have a chaotic sequence quantized over  $2^Q$  possible values which coincides exactly with its symbolic representation [Schweizer and Schimming, 2001b] and that can be described as a trellis encoded sequence, with a state given by a shift register of  $Q$  positions and two possible transitions in a Markov chain determined by the input bit  $b_n$ . When  $Q \rightarrow \infty$ , Eq. (3.1) becomes simply the recursion by the chaotic map  $f(z)$  without control. This kind of encoding is designed generally as chaos coded modulation (CCM) and has an equivalent representation in terms of a *trellis encoder view* [Kozic et al., 2006], since it is closely related to TCM [Ungerboeck, 1982]. When we are restricted to the values of  $S_Q$ , the encoding for the BSM map, for example, can

simply be performed following

$$\begin{aligned}
 r_i &= r_{i-1} \quad i = Q, \dots, 2, \\
 r_1 &= b_n, \\
 z_n &= \sum_{i=1}^Q 2^{-(Q+1-i)} \cdot r_i,
 \end{aligned} \tag{3.4}$$

where  $r_i$  are memory positions storing successive values of the input bit  $b_n$ . The related *trellis encoding like* structure can be seen in Fig. 3.1.



**Figure 3.1.** *Trellis encoder view of the BSM based CCM.*

For the kind of maps mentioned and when  $b_n$  is an *iid* binary sequence, the invariant pdf will be uniform in  $[0, 1]$  for  $z_n$ , and uniform in  $[-1, 1]$  for  $x_n$  [Spratt, 2003], so that the power of this signal will be 4 times the power of  $z_n$ , and, using the result of Eq. (2.42) [Escribano et al., 2006a], we can use in our calculations

$$P \approx \frac{1}{3}. \tag{3.5}$$

This power  $P$  corresponds to the case when  $Q \rightarrow \infty$ , but the difference with respect to the actual value when  $Q \geq 4$  is negligible for our purposes. The minimum squared Euclidean distance between all possible chaotic encoded sequences for this setup will correspond to the binary error events with weight 1 [Kozic et al., 2003b]. For the BSM based CCM, two sequences  $z_n$  and  $z'_n$  differing in only one bit will determine a loop of length  $L = Q + 1$  in their trellis representation (see Fig. 3.5), and so up to  $L - 1$  nonzero values of  $|z_n - z'_n|$ . This difference is an increasing power of  $1/2$  starting from  $(1/2)^Q$  when the paths start to diverge and ending in  $1/2$  before the merging of the paths. So the minimum squared Euclidean distance will be

$$d_{\min}^2 = 4 \sum_{n=m}^{L+m-1} (z_n - z'_n)^2 = 4 \sum_{i=1}^Q \frac{1}{4^i} = \frac{4}{3} \left(1 - \frac{1}{4^Q}\right). \tag{3.6}$$

When needed, we will use the value  $d_{\min}^2 \approx 4/3$  for  $Q \geq 4$  without incurring in much error.

In recent work [Kozic, 2006], this setup has been expanded to the design of multidimensional chaos-based modulators which have no associated corresponding

discrete map and whose rate is lower than  $R = 1$  sample per bit. The developments we will show in this and the following chapters are easily extended to this kind of modulators, since both our encoder and decoder are designed under the assumption of an equivalent trellis representation, which is common to all such class of chaos-based modulations.

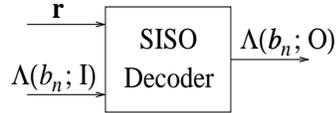
### 3.2.2. Decoder

According to what has been said at the beginning of this section, and following the principles of CCM shown in the preceding description, we will use for decoding an algorithm matched to the CCM trellis representation. In this case, the symbolic dynamics of the decoder is limited to the  $2^Q$  values within  $S_Q$  and to the possible transitions between the states represented by these discretized values. The decoding method has been chosen taking into account the following facts:

- As seen in Chapter 2, it has to be able to work over a potentially large block of chaotic samples to overcome the problem of estimating a chaotic sample based only on a limited number of successive samples.
- We will prefer a decoding algorithm based on MAP sequence estimation, since our first objective is to reduce the BER and MAP decoding was shown to give better results than ML decoding in Chapter 2.
- It has to be easily adaptable for concatenated iterative decoding.

This last condition stems from the fact that, as this kind of chaos-based modulation is in fact a coded modulation, we will study in the following chapters the possibility of exploiting successfully the serial and parallel concatenation of CCM's as is made with other standard coded modulations [Robertson and Wörz, 1998; Caire et al., 1998]. One possible decoding algorithm which meets the enumerated conditions is the additive Soft-Input Soft-Output (SISO) decoding algorithm proposed in [Benedetto et al., 1996] for binary block and convolutional codes. Though in its simplified forms it is suboptimal with respect to the multiplicative MAP BCJR algorithm [Bahl et al., 1974], it is less complex and does not suffer from numerical problems related with the required multiplications. We could thus process large blocks and we are not forced to resort to sliding window adaptations of the algorithm as seen in the previous chapter. The core of the algorithm follows the additive log-MAP algorithm of [Benedetto et al., 1996], but a few arrangements are needed to adapt it to CCM. This kind of SISO modules for sequence MAP decoding of chaos coded modulated signals have already been employed in communications schemes in [Escribano et al., 2005; Escribano et al., 2006c]. In the following, we will review the algorithm and point out the particularities.

In Fig. 3.2 we have depicted the SISO module with its inputs and outputs. The quantities  $\Lambda(b_n; \cdot)$  represent log probability ratios over the binary data  $b_n$ . The parameter I in  $\Lambda(\cdot; \cdot)$  stands for input, and O for output. The vector  $\mathbf{r} = (r_1, \dots, r_N)$



**Figure 3.2.** Input and output scheme of the SISO decoding module.

represents a block of  $N$  received samples. The sequence  $r_n$  corresponds to the encoded sequence  $x_n$  after having suffered the distortion of the channel, i.e.,

$$r_n = q_n(\mathbf{x}, \Theta), \quad (3.7)$$

where the function  $q_n(\mathbf{x}, \Theta)$  is the channel function at time  $n$ . It will include at least AWGN and possibly other sources of impairment. The expression in Eq. (3.7) remarks the fact that the output  $r_n$  will depend in general on the CCM signal block  $\mathbf{x} = (x_1, \dots, x_N)$  (or only on  $x_n$  if the channel has no memory) and on the channel parameters, which are denoted implicitly under the dummy vector  $\Theta$ .

Let us denote as  $s_i = i$  the state corresponding to the  $i$ -th element of  $S_Q$ ,  $i = 0, \dots, 2^Q - 1$ , see Eq. (3.3). If the encoder is initially at state  $s_j$  and an input bit  $b_n$  drives it at state  $s_k$  at time  $n$ , the output will be given by  $z_n = k \cdot 2^{-Q}$  and  $x_n = 2z_n - 1$ . According to these definitions, the output log probability ratio  $\Lambda(b_n; O)$  will be calculated as follows

$$\Lambda(b_n; O) = \log \frac{\sum_{s_j \xrightarrow{b_n=1} s_k} \exp(\alpha_n(s_j) + \pi(x_n; I) + \beta_n(s_k))}{\sum_{s_j \xrightarrow{b_n=0} s_k} \exp(\alpha_n(s_j) + \pi(x_n; I) + \beta_n(s_k))}, \quad (3.8)$$

where the summations are taken over all possible transitions  $s_j \xrightarrow{b_n=b} s_k$  between states where  $b_n$  takes the value  $b$  and the corresponding output is  $x_n$ . The quantities  $\alpha_n(\cdot)$  and  $\beta_n(\cdot)$  are probabilities for the corresponding states at time  $n$  which will be reviewed in the following.  $\pi(x_n; I) = \log [p(r_n|x_n)]$  is the transition log probability of the channel. In the AWGN channel, this transition probability is given by [Proakis, 2001] (see Subsection 2.4.2)

$$p(r_n|x_n) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(r_n - x_n)^2}{2\sigma^2}\right), \quad (3.9)$$

where, as usual,  $\sigma^2$  is the power of the Gaussian noise, related to the signal to noise ratio as

$$\sigma^{-2} = \frac{1}{2P} \frac{E_b}{N_0}. \quad (3.10)$$

Therefore, the log transition metric for the algorithm would be  $\pi(x_n; I) = A(r_n - x_n)^2 + B$ , where  $A$  and  $B$  are constants which depend on the signal to noise ratio  $E_b/N_0$ . For channels with other impairment sources, we will have to change this channel metric accordingly. Note that  $x_n$  is the quantized value for the chaotic sample taken as candidate to calculate the distance with respect to the received

sample  $r_n$  [Escribano et al., 2006b]. This is analogous to the replacing of the actual chaotic sample by the interval center values in Subsection 2.4.2.

If  $s_j$  denotes again a starting state and  $s_k$  an ending state for a transition at time  $n$  given by the bit  $b_n = b$  and whose corresponding output is  $x_n$ , the probabilities  $\alpha_n(\cdot)$  and  $\beta_n(\cdot)$  are calculated through a forward-backward algorithm according to

$$\alpha_n(s_k) = \log \left( \sum_{s_j} \exp(\alpha_{n-1}(s_j) + \pi(x_n; \mathbf{I}) + \pi(b_n = b; \mathbf{I})) \right) + h_\alpha, \\ n = 1, \dots, N, \quad (3.11)$$

$$\beta_{n-1}(s_j) = \log \left( \sum_{s_k} \exp(\beta_n(s_k) + \pi(x_n; \mathbf{I}) + \pi(b_n = b; \mathbf{I})) \right) + h_\beta, \\ n = N, \dots, 1. \quad (3.12)$$

The constants  $h_\alpha$  and  $h_\beta$  are normalization constants needed to avoid overflows in the algorithm, and  $\pi(b_n = b; \mathbf{I}) = \log [p(b_n = b; \mathbf{I})]$  is the *a priori* input log probability for  $b_n = b$ . Since the input log probability ratio  $\Lambda(b_n; \mathbf{I})$  is defined as

$$\Lambda(b_n; \mathbf{I}) = \log \frac{p(b_n = 1; \mathbf{I})}{p(b_n = 0; \mathbf{I})}, \quad (3.13)$$

then

$$\pi(p(b_n = 1; \mathbf{I})) = \Lambda(b_n; \mathbf{I}) - \log(1 + \exp(\Lambda(b_n; \mathbf{I}))), \quad (3.14)$$

$$\pi(p(b_n = 0; \mathbf{I})) = -\log(1 + \exp(\Lambda(b_n; \mathbf{I}))). \quad (3.15)$$

When there is no iterative decoding,  $p(b_n = b; \mathbf{I}) = 1/2$ ,  $b = 0, 1$ , and  $\Lambda(b_n; \mathbf{I}) = 0$ . To compute Eqs. (3.11) and (3.12), we need the initial values for probabilities  $\alpha_0(\cdot)$  and  $\beta_N(\cdot)$ . As we will always start with  $z_0 = 0$ , the encoder at the initial state will be in the zero state. Therefore  $\alpha_0(s_j = 0) = 0$  and  $\alpha_0(s_j \neq 0) = -\infty$ . With respect to  $\beta_N(\cdot)$ , we will not perform trellis termination, so that it will be initialized equiprobably,  $\beta_N(s_k) = \log(1/2^Q)$ .

The log probability ratio  $\Lambda(b_n; \mathbf{O})$  can be employed as input log probability ratio  $\Lambda(b_n; \mathbf{I})$  for other SISO module in a concatenated scheme, or, when using CCM alone, hard decoded to give the estimated bit  $\hat{b}_n$  as

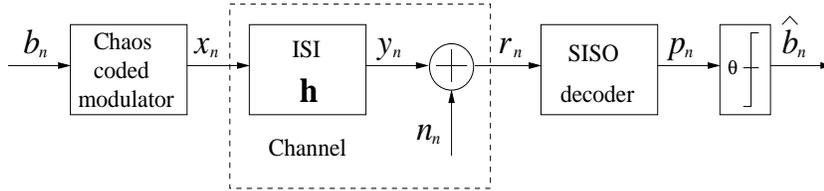
$$\Lambda(b_n; \mathbf{O}) > 0 \Rightarrow \hat{b}_n = 1, \\ \Lambda(b_n; \mathbf{O}) < 0 \Rightarrow \hat{b}_n = 0. \quad (3.16)$$

From now onwards, we will use the SISO module in the described log-MAP form, without any further simplification in the calculation of the logarithm of the sum of exponentials (as is made for example in its max-log-MAP variant [Benedetto et al., 1996]).

### 3.3. Chaos Coded Modulations over ISI channels

We have mentioned that we will consider at least the presence of AWGN in the channel, but also other possible kinds of impairment. As the target of this chapter is the evaluation of the CCM scheme in dispersive channels, face to which the chaotic sequences are supposed to offer potential good properties, we will first try a channel with intersymbol interference (ISI). The ISI can easily appear in a broadband modulation like CCM due to the usual presence of filters at the encoder or at the decoder needed to comply with band and interference restrictions. Though there are a large number of strategies to cope with this kind of distortion, we are interested in verifying if this coded modulation can be robust against a certain degree of ISI without the application of additional equalization techniques.

#### 3.3.1. Channel model



**Figure 3.3.** Block diagram of the communications system.

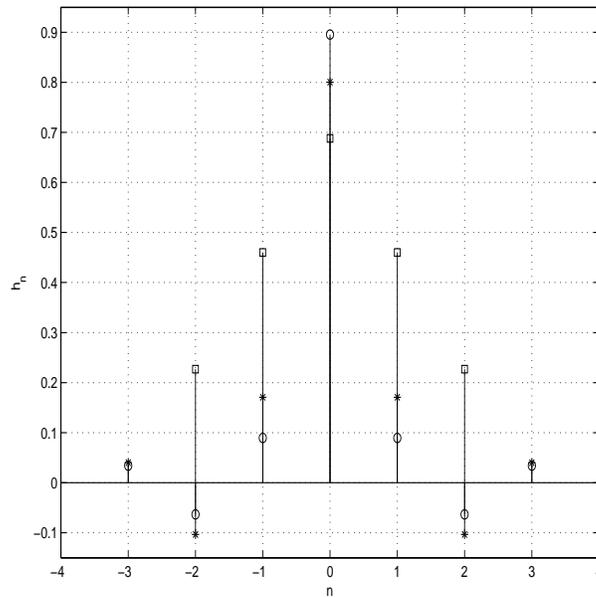
The channel is a conventional ISI channel with AWGN, where the ISI is simulated by a finite impulse response (FIR) linear filter [Oppenheim and Schaffer, 1989] given by a set of coefficients  $\mathbf{h} = (h_{-M}, \dots, h_M)$  [Proakis, 2001]. We show the complete communications system in Fig. 3.3, where we have depicted the chaos-based encoder, the channel model, the decoder and the decisor. This channel model corresponds to the case where, due to band constraints, some bandlimiting filtering has to be applied on the encoder or the decoder side (or both). As we do not perform iterative decoding and the *a priori* bit probabilities are the same, the SISO block only shows one input, i.e.,  $\Lambda(b_n; \mathbf{I}) = 0$ . For convenience, we have denoted the sequence of output log probability ratios  $\Lambda(b_n; \mathbf{O})$  as  $p_n$ . The hard decoding of these samples  $p_n$  is represented in Fig. 3.3 by a slicer with threshold  $\theta = 0$ .

The impulse response  $\mathbf{h}$  of the filter is normalized following  $\sum_{m=-M}^M |h_m|^2 = 1$ , so that it does not affect the signal to noise ratio at the receiver. The AWGN process adds *iid* samples  $n_n$  that are Gaussian distributed with mean  $\eta = 0$  and power  $\sigma^2$ . In our examples, we will consider three possible degrees of distortion, given by the impulse responses  $\mathbf{h}^l$  for low ISI,  $\mathbf{h}^m$  for moderate ISI and  $\mathbf{h}^h$  for high ISI. Their coefficients are shown in Table 3.1 and depicted in Fig. 3.4.

The low ISI filter represents a lowpass FIR filter with normalized cutoff frequency 0.9, the moderate ISI filter has normalized cutoff frequency 0.8, and the impulse response for high ISI was taken from [Proakis, 2001]. With respect to the decoding algorithm, we will use the metrics adapted to the AWGN channel as seen

$m$	$\mathbf{h}^l$	$\mathbf{h}^m$	$\mathbf{h}^h$
-3	0.034	0.040	0.0
-2	-0.063	-0.103	0.227
-1	0.089	0.171	0.460
0	0.896	0.800	0.688
1	0.089	0.171	0.460
2	-0.063	-0.103	0.227
3	0.034	0.040	0.0

**Table 3.1.** ISI FIR filter coefficients.



**Figure 3.4.** Coefficients for the ISI filters; 'o':  $\mathbf{h}^l$ ; '\*':  $\mathbf{h}^m$ ; '□':  $\mathbf{h}^h$ .

in the previous subsection, without further processing as we are not interested in performing equalization.

Taking into account these definitions, the signal at the output of the channel (see Fig. 3.3) will be

$$r_n = q_n(\mathbf{x}, \Theta) = y_n + n_n = \sum_{m=-M}^M h_m x_{n+m} + n_n. \quad (3.17)$$

In the next subsection we study theoretically the performance of the system in this kind of ISI channel and attempt to derive a bound for the bit error probability.

### 3.3.2. Performance analysis and bounds

To establish comparisons with the performance of the chaos coded modulated system, we will take into account also the case of uncoded binary phase shift keying (BPSK) over the same channel, as was done for the AWGN channel in Chapter 2. Recall that in the BPSK case the input to the channel is simply given by  $x_n^* = 2b_n - 1$ . Let  $\mathbf{x}^* = (x_{n-M}^*, \dots, x_{n+M}^*)$  denote an *iid* sequence of  $2M + 1$  BPSK symbols. There are a total of  $2^{2M+1}$  possible sequences  $\mathbf{x}^*$ . It is easy to show that the bit error probability under threshold decoding of the received value  $y_n = \sum_{m=-N}^N h_m x_{n+m}^* + n_n$  is given by [Prabhu, 1973]

$$P_b = \frac{1}{2^{2M+1}} \left[ \sum_{\mathbf{x}^* | x_n^* = -1} \operatorname{erfc} \left( -\sqrt{\frac{E_b}{N_0}} \sum_{m=-M}^M h_m x_{n+m}^* \right) + \sum_{\mathbf{x}^* | x_n^* = +1} \operatorname{erfc} \left( \sqrt{\frac{E_b}{N_0}} \sum_{m=-M}^M h_m x_{n+m}^* \right) \right], \quad (3.18)$$

where the summations are taken over all possible sequences  $\mathbf{x}^*$  when  $x_n^*$  takes the corresponding value  $-1$  or  $+1$ . This bit error probability can be calculated only for low and moderate values of  $M$ , but it is also possible to give good bounds for the case when  $M \rightarrow \infty$  [Prabhu, 1973].

In the case of our CCM system, we need to compute the error probability in terms of the associated error events, since we perform sequence decoding [Lin and Costello, Jr., 2004]. Therefore, there will be an error event when, having sent the sequence  $\mathbf{x}$ , the decoder chooses  $\mathbf{x}' \neq \mathbf{x}$ , whose values diverge from a common state at time  $m$  and eventually merge again after  $L$  steps in (possibly) other different common state<sup>1</sup>, i.e.,  $x_n \neq x'_n$ ,  $n = m, \dots, L + m - 2$  ( $x_{m+L-1} = x'_{m+L-1}$  because the ending states are the same). Since the *a priori* probabilities are the same, the MAP decoding performed by the SISO module is equivalent to ML decoding, and, as the decoding is based upon the metric  $(r_n - x_n)^2$  (see Eq. (3.9)), an error event of the described type will happen when [Proakis, 2001]

$$\sum_{n=m}^{L+m-1} (r_n - x'_n)^2 < \sum_{n=m}^{L+m-1} (r_n - x_n)^2. \quad (3.19)$$

This is equivalent to

$$\sum_{n=m}^{L+m-1} ((y_n - x'_n) + n_n)^2 < \sum_{n=m}^{L+m-1} ((y_n - x_n) + n_n)^2, \quad (3.20)$$

<sup>1</sup>This is consequence of the nonlinearity of this kind of coded modulation, which makes the analysis more complex, since the error events cannot be simply calculated assuming that the all-zero codeword has been sent. Note also that we are assuming implicitly  $L \ll N$ , so that these error events appear randomly at any point of the sequence.

and, after a little algebra, to

$$\sum_{n=m}^{L+m-1} (y_n - x'_n)^2 - \sum_{n=m}^{L+m-1} (y_n - x_n)^2 < \sum_{n=m}^{L+m-1} 2(x_n - x'_n)n_n = A, \quad (3.21)$$

where  $y_n$  was defined in Eq. (3.17) and we have defined the random variable (RV)  $A$  for convenience. Since  $n_n$  are *iid* samples of a Gaussian process  $\mathcal{N}(0, \sigma^2)$ , and  $x_n$  and  $x'_n$  are known,  $A$  is a Gaussian RV with mean

$$\eta_A = E \left[ 2 \sum_{n=m}^{L+m-1} (x_n - x'_n)n_n \right] = 0, \quad (3.22)$$

and variance

$$\sigma_A^2 = E \left[ 4 \left( \sum_{n=m}^{L+m-1} (x_n - x'_n)n_n \right)^2 \right] = 4\sigma^2 \sum_{n=m}^{L+m-1} (x_n - x'_n)^2 = 4\sigma^2 d_E^2, \quad (3.23)$$

where

$$d_E^2 = \sum_{n=m}^{n=L+m-1} (x_n - x'_n)^2 \quad (3.24)$$

is the squared Euclidean distance between sequences  $\mathbf{x}$  and  $\mathbf{x}'$ . Recall that, when  $A$  is a Gaussian RV [Haykin, 2001], we have

$$P(A > w) = \frac{1}{2} \operatorname{erfc} \left( \frac{w}{\sqrt{2}\sigma_A} \right). \quad (3.25)$$

Then, recalling Eqs. (3.21) and (3.10), we can derive the error event probability (EEP) for an input binary error event loop of length  $L$  as

$$P_e(\mathbf{x} \rightarrow \mathbf{x}'|\mathbf{x}) = \frac{1}{2} \operatorname{erfc} \left( \sqrt{\frac{d_{eq}^2 E_b}{4P N_0}} \right), \quad (3.26)$$

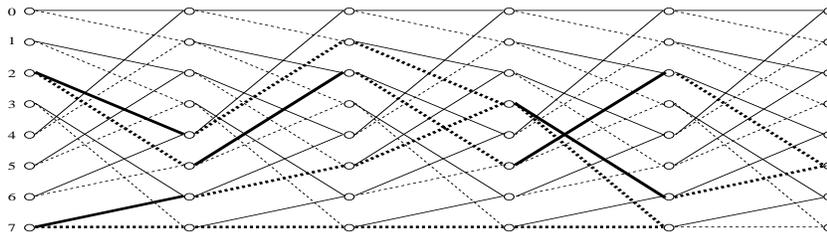
where  $P = 1/3$  is the power of the CCM signal, and  $d_{eq}^2$  is an *equivalent squared Euclidean distance* in the ISI channel between the sequences  $\mathbf{x}'$  and  $\mathbf{x}$ , given by

$$d_{eq}^2 = \left( \frac{\sum_{n=1}^M (y_n - x'_n)^2 - \sum_{n=1}^M (y_n - x_n)^2}{d_E} \right)^2. \quad (3.27)$$

We have defined this equivalent distance in analogy with the EEP in a channel where the only source of distortion is AWGN [Proakis, 2001]

$$P_e(\mathbf{x} \rightarrow \mathbf{x}'|\mathbf{x}) = \frac{1}{2} \operatorname{erfc} \left( \sqrt{\frac{d_E^2 E_b}{4P N_0}} \right). \quad (3.28)$$

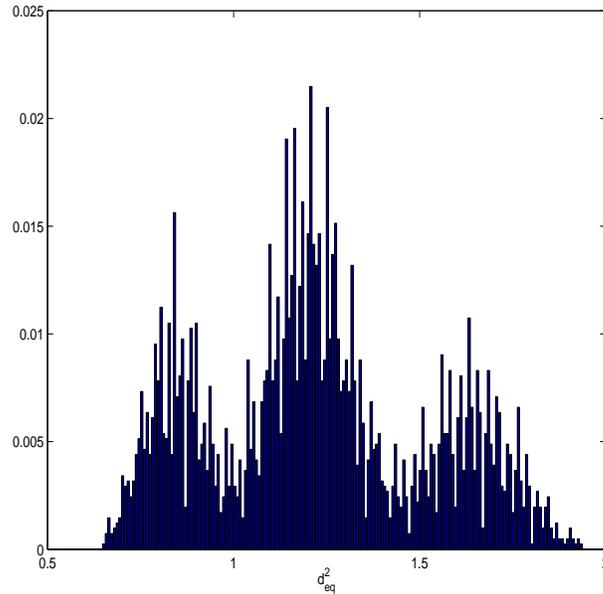
It is clear that the evaluation of the bit error probability starting from the event error probability and using union bound techniques [Proakis, 2001; Lin and Costello, Jr., 2004] is unfeasible even for moderate values of  $N$ ,  $Q$  and  $M$ . Since we cannot use the linearity property and assume that the all-zero codeword has been sent, the number of possible error paths for a sequence of length  $N$  under ML decoding makes the required calculations prohibitive. Nevertheless, we can resort to the theory of rare events, so that, for high  $E_b/N_0$  values and low ISI, we can assume that the dominant error events will be of the minimum  $d_{eq}^2$  kind. Exploring the error events for minimum  $d_{eq}^2$  is also quite a burdensome task for a coded modulation not admitting linear simplifications like other TCM schemes [Schlegel, 1991; Carlisle et al., 1994; Li et al., 2004; Ryan and Tang, 2004], but we have found out in our simulations that such error events for the BSM CCM in low or moderate ISI are the ones involving an encoded message of the kind  $\dots b_{n-1}01b_{n+2}\dots$  and a decoded message with two adjacent bit errors  $\dots b_{n-1}10b_{n+2}\dots$ , and viceversa. In Figure (3.5) we can see the trellis for  $Q = 3$ . We have highlighted with boldline two pairs of paths starting from the same state and merging after some steps. We have two merging paths with length  $L = Q + 1$  leading to 1 bit error, 1111 and 0111 (these are the kind of error paths leading to minimum squared Euclidean distance events for a BSM CCM in the AWGN channel [Kozic et al., 2006]), and two merging paths with length  $L = Q + 2$  leading to 2 bit errors, 01101 and 10101. With independence of  $Q$  and of the preceding or following bits, in the case of low or moderate ISI, the simulations have pointed out that the error events are mainly of the second kind for large  $E_b/N_0$ . Even the error paths leading to 2 adjacent bit errors and beginning with 00 or with 11 lead to much higher  $d_{eq}^2$  values and less frequent error events. We can say that the effect of the ISI FIR filter is a transformation of the error event spectrum, so that the minimum squared distance now is associated with input binary error events of Hamming weight 2 and a special structure [Schlegel, 1991; Li et al., 2004].



**Figure 3.5.** *Chaos coded modulation trellis for the BSM encoder with  $Q = 3$ . Continuous lines: input bit 0; dashed lines: input bit 1.*

According to this, we have only to focus on the error events of the described kind to get a bound of the bit error probability in the high signal to noise ratio region. We have two possibilities: we could compute the bound based on the minimum  $d_{eq}^2$  over all such error events, or compute the bound taking into account all possibilities for such error events. Recall again that, since the resulting chaos coded modulation is not linear, we cannot assume that the all-zero codeword has been sent and make the

calculations as a function of the associated binary error event alone. The presence of the ISI FIR filter makes things worse, preventing any simple relationship between the values of  $d_{eq}^2$  and the error events starting with 01 against 10, and viceversa. Now we have to examine all the possible cases, starting from all the possible states, not just the all-zero state. Nevertheless, if we consider the sequence  $\mathbf{x}$  encoding the binary message  $\dots, b_{n-1}, 0, 1, b_{n+2}, \dots$  and a corresponding sequence  $\mathbf{x}'$  encoding the erroneous binary message  $\dots, b_{n-1}, 1, 0, b_{n+2}, \dots$ , we have only to evaluate  $2^{4M+2Q-1}$  potentially different  $d_{eq}^2$  values. Since  $y_n = \sum_{m=-M}^M h_m x_{n+m}$  and  $x_n$  is uniquely determined by  $b_{n-Q+1}, \dots, b_n$ , it is easy to verify that only the binary samples  $b_{n-2M-Q+1}, \dots, b_{n-1}, b_n^*, b_{n+1}^*, b_{n+2}, \dots, b_{n+2M+Q+1}$  determine values of  $x_k, x'_k$  and  $y_k$  that make quantities  $(y_k - x_k)^2 \neq (y_k - x'_k)^2$ , for  $k = n - M, \dots, n + M + Q$ . It is the same for the 10  $\rightarrow$  01 case, thus yielding a total of  $2^{4M+2Q}$  possibilities. In our examples, where  $M$  takes values of 2 and 3, and  $Q$  between 4 and 6, this evaluation, though burdensome, is computationally feasible<sup>2</sup>. In Figure (3.6) we plot the histogram of the corresponding  $d_{eq}^2$  for the case of low ISI. It can be seen that the distribution of  $d_{eq}^2$  is centered around  $d_E^2 = 4/3$ , which we have seen is the approximated value of the minimum squared Euclidean distance for this kind of system in AWGN channels [Kozic et al., 2006].



**Figure 3.6.** Normalized histogram of  $d_{eq}^2$  for  $Q = 4$  and for the error patterns under evaluation with ISI filter  $\mathbf{h}^l$ .

If we consider that all the error events for high signal to noise ration involve the absolute minimum of  $d_{eq}^2$ , the bit error probability can be bounded by [Proakis,

<sup>2</sup>In the case with  $Q = 6$  and  $M = 3$ , the possibilities are  $2^{24}$

2001]

$$P_b \leq \beta_{\min} \frac{1}{2} \operatorname{erfc} \left( \sqrt{\frac{d_{eq\min}^2 E_b}{4P N_0}} \right). \quad (3.29)$$

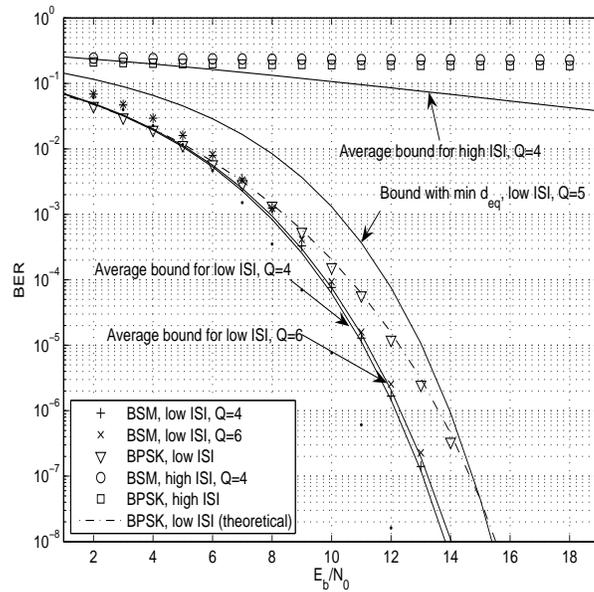
The bit enumerator function  $\beta_{\min}$  has to take into account the fact that the error events lead to 2 adjacent error bits, and that only half of such possible error patterns can be involved (01 *vs* 10 and viceversa, but not 00 *vs* 11 and viceversa), with independence of the starting and ending state. Thus  $\beta_{\min} = 2/2 = 1$ . In Fig. (3.6) we have depicted the histogram of the  $2^{4M+2Q}$  possible values for  $d_{eq}^2$ , and we see that this bound would be very conservative, as there are only a few number of error events leading to the absolute minimum of  $d_{eq}^2$ . A tighter bound can be evaluated if we consider all possibilities and assume that all of them are equally likely. This makes sense, since all binary messages are equiprobable with this setup. In this case, the bit error probability can be approximated by the average bound

$$P_b \approx \frac{\beta_{\min}}{2^{4M+2Q+1}} \sum_{d_{eq}^2} \operatorname{erfc} \left( \sqrt{\frac{d_{eq}^2 E_b}{4P N_0}} \right), \quad (3.30)$$

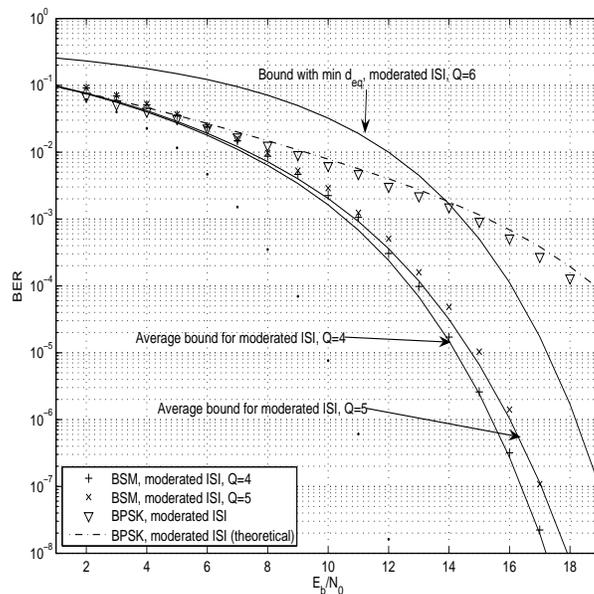
where the sum is taken over the  $2^{4M+2Q}$  possible values of  $d_{eq}^2$ . As before,  $\beta_{\min} = 2$ .

### 3.3.3. Simulation results

In Figs. 3.7 and 3.8 we can see the results for the proposed chaos coded modulation system and for uncoded BPSK, together with the proposed bounds. All simulations have been run with bit and symbol blocks of length  $N = 10000$ , and the BER computed for a minimum of 100 frames with errors. We can see that, for the cases with low and moderate ISI, the behaviour of the chaos coded modulation is better than uncoded BPSK for high signal to noise ratios (specially in the case of moderate ISI), while with AWGN alone the BPSK system is known to perform better than the BSM chaotic system [Kozic et al., 2006]. We have already verified it in Chapter 2. This can be explained because the chaotic signal may be seen as a case of *precoding* for the ISI channel, and this can be fully exploited by a good decoder, even in the absence of equalization [Laroia et al., 1993; Laroia, 1996]. On the other side, we can see that the influence of  $Q$  is slight, which is a desirable feature in such chaos coded system [Kozic et al., 2006]. We also see that the proposed average bound is very tight when the ISI level is not high, while, as foreseen, the simpler bound based upon the minimum value of  $d_{eq}^2$  is much looser. When the ISI level is high, the behaviour of the BSM CCM and of BPSK is equally poor, and the bounds are useless, since there are a large number of error events involved, even for high  $E_b/N_0$ .



**Figure 3.7.** Simulation results and bounds for low and high ISI. The bounds for the BSM CCM are depicted with continuous lines. The theoretical bit error probability for BPSK is represented with a dash-dotted line. The performance of the BSM CCM in the AWGN channel, when  $Q = 5$ , is shown with dotted line as a reference.

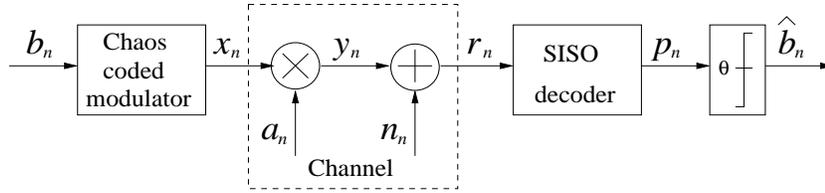


**Figure 3.8.** Simulation results and bounds for moderate ISI. The bounds for the BSM CCM are depicted with continuous lines. The theoretical bit error probability for BPSK is represented with a dash-dotted line. The performance of the BSM CCM in the AWGN channel, when  $Q = 5$ , is shown with dotted line as a reference.

### 3.4. Chaos Coded Modulations over frequency-non selective fading channels

After having seen the effect of ISI on the chaotic signal, we will consider a radio environment, where one of the main sources of distortion is the attenuation of the signal together with the fact that it can reach the receiver from different paths, thus leading to a self interference phenomenon. This multipath situation gives rise to fading [Biglieri et al., 1998], which can affect the amplitude of the received signal in different ways. We will only focus here on one class of fading, which we review briefly in the following subsection.

#### 3.4.1. Channel model



**Figure 3.9.** Block diagram of the communications system.

The channel includes AWGN together with frequency non-selective (flat) fading [Biglieri et al., 1998]. This fading process is described by an uncorrelated sequence of amplitudes  $\mathbf{a} = (a_1, \dots, a_N)$  for each block of  $N$  chaos coded modulated samples  $\mathbf{x} = (x_1, \dots, x_N)$ . We assume that the fading process is slow enough so that the fading amplitudes remain constant throughout a symbol period, but change from symbol to symbol. Each  $a_n$  follows a Rician probability density function (pdf) given by

$$p(a) = 2a(1 + K)e^{-a^2(1+K)-K} I_0 \left( 2a\sqrt{K(K+1)} \right), \quad a \geq 0, \quad (3.31)$$

where  $I_0(\cdot)$  is the zeroth-order modified Bessel function of the first kind, and  $K$  is the ratio of specular to diffuse energy [Proakis, 2001]. The Rice distribution is usually given in terms of two parameters,  $\nu_R$  and  $\sigma_R$ , and they are related to  $K$  as

$$K = \frac{\nu_R^2}{2\sigma_R^2}, \quad (3.32)$$

so that  $\nu_R^2$  is directly related to the energy of the specular component and  $\sigma_R^2$ , to the energy of the diffuse component. This model describes well the radio propagation anomaly caused by a partial cancellation of the signal by itself when it reaches the receiver from at least two different paths. The Rician fading occurs when one of the paths, typically a line of sight signal (specular component), is much stronger than the others (diffuse component). When there is no line of sight signal,  $K = 0$  and we have Rayleigh fading. When the specular component becomes dominant and

$K \rightarrow \infty$ , the channel tends to be an AWGN channel without fading. The mean and variance of the related Rician process are given by

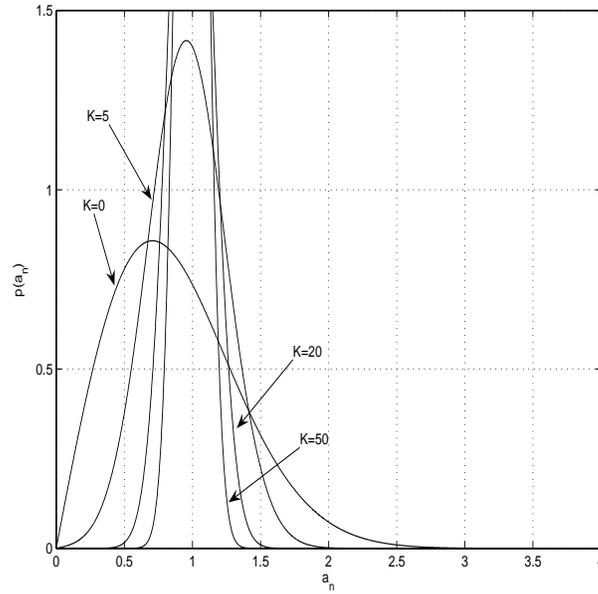
$$\eta_a = \frac{1}{2} \sqrt{\frac{\pi}{1+K}} e^{-\frac{K}{2}} \left[ (1+K) I_0 \left( \frac{K}{2} \right) + K I_1 \left( \frac{K}{2} \right) \right], \quad (3.33)$$

$$\sigma_a^2 = 1 - \eta_a^2, \quad (3.34)$$

where  $I_1(\cdot)$  is the first-order modified Bessel function of the first kind. The pdf is normalized so that  $E[a^2] = 1$  to make sure that the signal to noise ratio at the transmitter and the decoder side are the same. The channel is further described by the presence of the AWGN process, which adds *iid* Gaussian samples  $n_n$  with zero mean and power  $\sigma^2$ . As seen in Fig. 3.9, which is completely analogous to Fig. 3.3 with the exception of the channel description, the signal arriving at the decoder will be given by

$$r_n = q_n(\mathbf{x}, \Theta) = q_n(x_n, \Theta) = y_n + n_n = a_n x_n + n_n. \quad (3.35)$$

In Fig. 3.10 we show the pdf of the fading amplitudes for the cases we will consider in our simulations,  $K = 0, 5, 20, 50$ . We see that the Rayleigh case is the worst case and how  $p(a) \rightarrow \delta(a - 1)$  as  $K \rightarrow \infty$ .



**Figure 3.10.** Rician pdf for several specular to diffuse energy ratios.

In a channel with fading, two important possible situations arise. In the most favorable one, the sequence of fading amplitudes  $\mathbf{a}$  is known to the decoder due to the presence of some channel estimation method. In this case, we say to have perfect channel state information (CSI) and the decoding algorithm in the SISO module uses the metric  $(r_n - a_n x_n)^2$  [Biglieri et al., 1998] instead of the simple  $(r_n - x_n)^2$  AWGN metric. In the other situation, the receiver has only information, if any, about some parameters of the channel, such as the mean  $\eta_a$ , and we do not have perfect CSI.

In this case, the metric for the SISO module will be taken as  $(r_n - \eta_a x_n)^2$  [Hall and Wilson, 1998]. In the next subsection, we will examine both situations theoretically and try to derive bounds for the bit error probability.

### 3.4.2. Performance analysis and bounds

To calculate the bounds of the bit error probability we have to take into account that, as the *a priori* probabilities are the same, the MAP decoding is again equivalent to ML decoding. Let us recall that there will be an error event when, having sent the sequence  $\mathbf{x}$ , the decoder chooses  $\mathbf{x}' \neq \mathbf{x}$ , both sequences starting in the same state and merging again in (possibly) other state after  $L$  steps. In the following subsections we will develop the bounds for the case with perfect CSI and for the case without CSI under the assumption of ML decoding and using the corresponding decoder metrics. We will rely also on the calculation of the EEP (which is the same as the pairwise error probability between two words differing in an error event), since this is the usual approach for coded modulations in the fading channel [Biglieri et al., 1995; Sayana and Gelfand, 2004; Kambo et al., 2005]. As in the Section 3.3, we will use the performance of BPSK over the same channel to establish comparisons and show that there is a potential coding gain in dispersive channels contrary to the AWGN channel case. The bound for BPSK over a Rician fading channel can be easily calculated [Proakis, 2001]

$$P_b \leq \frac{1}{2} \frac{1 + K}{1 + K + \frac{E_b}{N_0}} \exp\left(-\frac{K \frac{E_b}{N_0}}{1 + K + \frac{E_b}{N_0}}\right). \quad (3.36)$$

For the special case of  $K = 0$  (Rayleigh fading), there exists an exact expression for the bit error probability [Haykin, 2001]

$$P_b = \frac{1}{2} \left(1 - \sqrt{\frac{\frac{E_b}{N_0}}{1 + \frac{E_b}{N_0}}}\right). \quad (3.37)$$

#### Performance with CSI

We have mentioned that the metric for the decoding algorithm in the case with perfect CSI is calculated as a function of  $(r_n - a_n x_n)^2$ , so that, under ML decoding, the decoder will choose the sequence  $\mathbf{x}'$  instead of  $\mathbf{x}$  when [Proakis, 2001]

$$\sum_{n=m}^{L+m-1} (r_n - a_n x'_n)^2 < \sum_{n=m}^{L+m-1} (r_n - a_n x_n)^2, \quad (3.38)$$

for an error loop of length  $L$  starting at time  $m$ . This is equivalent to

$$\sum_{n=m}^{L+m-1} (a_n(x_n - x'_n) + n_n)^2 < \sum_{n=m}^{L+m-1} n_n^2, \quad (3.39)$$

and, after some algebra, to

$$\sum_{n=m}^{L+m-1} a_n^2(x_n - x'_n)^2 = A < 2 \sum_{n=m}^{L+m-1} n_n a_n(x_n - x'_n) = B \quad (3.40)$$

where  $A$  and  $B$  are RVs whose meaning will be clear in the sequel. As  $\mathbf{x}$ ,  $\mathbf{x}'$  and  $\mathbf{a}$  are known and  $n_n$  are *iid* samples of a Gaussian process  $\mathcal{N}(0, \sigma^2)$ ,  $B$  is a Gaussian random variable with mean

$$\eta_B = E \left[ 2 \sum_{n=m}^{L+m-1} n_n a_n(x_n - x'_n) \right] = 0, \quad (3.41)$$

and variance

$$\sigma_B^2 = E \left[ 4 \left( \sum_{n=m}^{L+m-1} n_n a_n(x_n - x'_n) \right)^2 \right] = 4\sigma^2 \sum_{n=m}^{L+m-1} a_n^2(x_n - x'_n)^2 = 4\sigma^2 A. \quad (3.42)$$

Taking into account the result of Eq. (3.25), the EEP conditioned to the fading amplitudes for the perfect CSI case will be

$$P_e(\mathbf{x} \rightarrow \mathbf{x}' | \mathbf{x}, \mathbf{a}) = \frac{1}{2} \operatorname{erfc} \left( \sqrt{\frac{1}{4P} \frac{E_b}{N_0} A} \right) \leq \frac{1}{2} \exp \left( -\frac{1}{4P} \frac{E_b}{N_0} A \right), \quad (3.43)$$

where we have made use of the inequality  $\operatorname{erfc}(x) \leq \exp(-x^2)$  [Proakis, 2001].

The unconditioned EEP will be then upper bounded by

$$P_e(\mathbf{x} \rightarrow \mathbf{x}' | \mathbf{x}) = E_{\mathbf{a}} [P_e(\mathbf{x} \rightarrow \mathbf{x}' | \mathbf{x}, \mathbf{a})] \leq E_{\mathbf{a}} \left[ \frac{1}{2} \exp \left( -\frac{1}{4P} \frac{E_b}{N_0} A \right) \right], \quad (3.44)$$

where the subindex  $\mathbf{a}$  means that the expectation has to be taken over the RVs  $a_n$ . Since  $a_n$  follows a Rician distribution, each  $a_n^2(x_n - x'_n)^2$  is a noncentral  $\chi^2$  RV. Let us denote as  $\Psi_n(s)$  the characteristic function [Papoulis and Pillai, 2002] of such noncentral  $\chi^2$  RV. Following [Proakis, 2001] and recalling that the Rician RV  $a_n$  is scaled by  $(x_n - x'_n)$ , and taking into account that  $K = \nu_R^2 / (2\sigma_R^2)$ , this characteristic function can be written as

$$\Psi_n(s) = \frac{1 + K}{1 + K - s(x_n - x'_n)^2} \exp \left( \frac{K s(x_n - x'_n)^2}{1 + K - s(x_n - x'_n)^2} \right). \quad (3.45)$$

As the RV  $A$  is a sum of *iid* RVs, its characteristic function will be

$$\Psi_A(s) = \prod_{n=m}^{L+m-1} \Psi_n(s). \quad (3.46)$$

By definition of the characteristic function [Proakis, 2001]

$$E[\exp(sA)] = \Psi_A(s), \quad (3.47)$$

so that

$$E_{\mathbf{a}} \left[ \exp \left( -\frac{1}{4P} \frac{E_b}{N_0} A \right) \right] = \prod_{n=m}^{n=L+m-1} \Psi_n \left( -\frac{1}{4P} \frac{E_b}{N_0} \right). \quad (3.48)$$

Finally, taking into account that  $x_{m+L-1} = x'_{m+L-1}$ , we can write the EEP as

$$P_e(\mathbf{x} \rightarrow \mathbf{x}' | \mathbf{x}) \leq \frac{1}{2} \prod_{n=m}^{L+m-2} \frac{1+K}{1+K + \frac{1}{4P} \frac{E_b}{N_0} (x_n - x'_n)^2} \cdot \exp \left( -\frac{K \frac{1}{4P} \frac{E_b}{N_0} (x_n - x'_n)^2}{1+K + \frac{1}{4P} \frac{E_b}{N_0} (x_n - x'_n)^2} \right). \quad (3.49)$$

For an error event with  $L < \infty$ , when  $E_b/N_0 \rightarrow \infty$ , this EEP tends to

$$P_e(\mathbf{x} \rightarrow \mathbf{x}' | \mathbf{x}) \rightarrow \frac{1}{2} \frac{(1+K)^{L-1}}{\left( \frac{1}{4P} \frac{E_b}{N_0} \right)^{L-1} \prod_{n=m}^{L+m-2} (x_n - x'_n)^2} \exp(-(L-1)K). \quad (3.50)$$

The factor

$$d_P^2(\mathbf{x}, \mathbf{x}') = \prod_{n=m}^{L+m-2} (x_n - x'_n)^2, \quad (3.51)$$

is the product distance associated to the error event [Anderson and Svensson, 2003]. The number of samples with  $x_n \neq x'_n$  is the Hamming distance ( $d_H(\mathbf{x}, \mathbf{x}') = L-1$ ) of such error event [Anderson and Svensson, 2003]. Therefore, for the fading channel with CSI, the determinant factor in the high signal to noise ratio is not the minimum squared Euclidean distance, but seems to be the pair  $d_P^2(\mathbf{x}, \mathbf{x}')$  and  $d_H(\mathbf{x}, \mathbf{x}')$  instead. This is specially true for traditional coded modulations, but we will show with the BSM example that this pair is not so important in CCM systems, since normally they cannot be increased at the same time.

To give a bound for the bit error probability, we see in Eq. (3.49) that the most probable error events when  $E_b/N_0 \rightarrow \infty$  will be those with minimum  $(x_n - x'_n)^2$ ,  $n = m, \dots, m+L-1$ , and those error events will be the ones related with the minimum of  $d_E^2$  for the BSM CCM. These error events, in terms of the binary input, are given by  $b_m, b_{m+1}, \dots, b_{Q+m}$  vs  $b_m^*, b_{m+1}, \dots, b_{Q+m}$ , with  $b_m \neq b_m^*$ , and are of length  $L = Q+1$ . They lead to  $(x_n - x'_n)^2 = (1/4)^{m+Q-1-n}$ ,  $n = m, \dots, m+Q-1$ , with independence of the previous values of  $b_n$  and  $x_n$ , and so the bit error probability could be approximated in the high signal to noise ratio region by

$$P_b \approx \beta_{\min} \frac{1}{2} \prod_{i=0}^{Q-1} \frac{1+K}{1+K + \frac{1}{4P} \frac{E_b}{N_0} \frac{1}{4^i}} \exp \left( -\frac{K \frac{1}{4P} \frac{E_b}{N_0} \frac{1}{4^i}}{1+K + \frac{1}{4P} \frac{E_b}{N_0} \frac{1}{4^i}} \right), \quad (3.52)$$

where  $\beta_{\min} = 1$  is the bit enumerator associated to these error events with  $d_{\min}^2 = 4/3$ , since there is only one bit on error. Note that the product distance and the

Hamming distance in this case are

$$d_P^2(\mathbf{x}, \mathbf{x}') = \prod_{i=0}^{Q-1} \frac{1}{4^i}, \quad (3.53)$$

$$d_H^2(\mathbf{x}, \mathbf{x}') = Q. \quad (3.54)$$

We could think of managing the BER by changing the quantization level  $Q$ , but in fact there is a tight tradeoff: while  $Q \rightarrow \infty$  leads to a Hamming distance tending to  $\infty$ , the product distance tends to 0, so that, at the end, the result of the bound in Eq. (3.52) will remain basically the same and independent of the values of  $Q$  when  $Q \geq 5$ .

### Performance without CSI

In the case without CSI, we have seen that the metric for the decoder is calculated as a function of  $(r_n - \eta_a x_n)^2$ , and so an error event loop of length  $L$  will occur when

$$\sum_{n=m}^{L+m-1} (r_n - \eta_a x'_n)^2 < \sum_{n=m}^{L+m-1} (r_n - \eta_a x_n)^2 \quad (3.55)$$

which is equivalent to

$$\sum_{n=m}^{L+m-1} ((a_n x_n - \eta_a x'_n) + n_n)^2 < \sum_{n=m}^{L+m-1} ((a_n x_n - \eta_a x_n) + n_n)^2, \quad (3.56)$$

and, after some algebra, to

$$\sum_{n=m}^{L+m-1} (a_n x_n - \eta_a x'_n)^2 - \sum_{n=m}^{L+m-1} (a_n x_n - \eta_a x_n)^2 = A < 2\eta_a \sum_{n=m}^{L+m-1} n_n (x_n - x'_n) = B, \quad (3.57)$$

where again  $A$  and  $B$  are RVs defined for convenience. The sequences  $\mathbf{x}$  and  $\mathbf{x}'$  are known, and so the RV  $B$  is a weighted sum of *iid* Gaussian variables. Therefore,  $B$  is a Gaussian RV with mean

$$\eta_B = E \left[ 2\eta_a \sum_{n=m}^{L+m-1} n_n (x_n - x'_n) \right] = 0, \quad (3.58)$$

and variance

$$\sigma_B^2 = E \left[ 4\eta_a^2 \left( \sum_{n=m}^{L+m-1} n_n (x_n - x'_n) \right)^2 \right] = 4\eta_a^2 \sigma^2 \sum_{n=m}^{L+m-1} (x_n - x'_n)^2 = 4\eta_a^2 \sigma^2 d_E^2. \quad (3.59)$$

With these definitions, and using Eq. (3.25), the EEP conditioned to the fading amplitudes for the case without CSI will be

$$P_e(\mathbf{x} \rightarrow \mathbf{x}' | \mathbf{x}, \mathbf{a}) = \frac{1}{2} \operatorname{erfc} \left( \sqrt{\frac{1}{4P} \frac{E_b}{N_0} \frac{A}{\eta_a d_E}} \right) \leq \frac{1}{2} \exp \left( -\frac{1}{4P} \frac{E_b}{N_0} \frac{A^2}{\eta_a^2 d_E^2} \right). \quad (3.60)$$

The RV  $A$  is a weighted sum of several *iid* RVs, and if  $L \geq 5$ , by virtue of the Central Limit Theorem [Papoulis and Pillai, 2002], it could be well approximated by a Gaussian RV with mean

$$\begin{aligned} \eta_A &= E_{\mathbf{a}} \left[ \sum_{n=m}^{L+m-1} (a_n x_n - \eta_a x'_n)^2 - \sum_{n=m}^{L+m-1} (a_n x_n - \eta_a x_n)^2 \right] = \\ &= \eta_a^2 \sum_{n=m}^{L+m-1} (x_n - x'_n)^2 = \eta_a^2 d_E^2, \end{aligned} \quad (3.61)$$

and variance

$$\begin{aligned} \sigma_A^2 &= E_{\mathbf{a}} \left[ \left( \sum_{n=m}^{L+m-1} (a_n x_n - \eta_a x'_n)^2 - \sum_{n=m}^{L+m-1} (a_n x_n - \eta_a x_n)^2 - \eta_A \right)^2 \right] = \\ &= 4\sigma_a^2 \eta_a^2 \sum_{n=m}^{L+m-1} x_n^2 (x_n - x'_n)^2. \end{aligned} \quad (3.62)$$

Let us define the variable  $\theta(\mathbf{x}, \mathbf{x}')$  as

$$\theta(\mathbf{x}, \mathbf{x}') = \sum_{n=m}^{L+m-1} x_n^2 (x_n - x'_n)^2. \quad (3.63)$$

If  $\rho$  is a constant and  $B$  a Gaussian RV, when

$$\rho < \frac{1}{2\sigma_A^2}, \quad (3.64)$$

it is easy to show that

$$E [\exp(\rho B^2)] = \frac{1}{\sqrt{1 - 2\rho\sigma_A^2}} \exp\left(\frac{\rho\eta_A^2}{1 - 2\rho\sigma_A^2}\right). \quad (3.65)$$

Taking this into account and recalling Eq. (3.60), the bound for the unconditioned EEP will be

$$\begin{aligned} P_e(\mathbf{x} \rightarrow \mathbf{x}' | \mathbf{x}) &= E_{\mathbf{a}} [P_e(\mathbf{x} \rightarrow \mathbf{x}' | \mathbf{x}, \mathbf{a})] \leq \\ &\leq \frac{1}{2} \frac{1}{\sqrt{1 + \frac{2}{P} \frac{E_b}{N_0} \sigma_a^2 \frac{\theta(\mathbf{x}, \mathbf{x}')}{d_E^2}}} \exp\left(-\frac{\frac{1}{4P} \frac{E_b}{N_0} \eta_a^2 d_E^2}{1 + \frac{2}{P} \frac{E_b}{N_0} \sigma_a^2 \frac{\theta(\mathbf{x}, \mathbf{x}')}{d_E^2}}\right). \end{aligned} \quad (3.66)$$

The condition of Eq. (3.64) is always met, since, as seen in Eq. (3.60)

$$\rho = -\frac{1}{4P} \frac{E_b}{N_0} \frac{1}{\eta_a^2 d_E^2} < 0. \quad (3.67)$$

In the high signal to noise ratio region, when  $E_b/N_0 \rightarrow \infty$ , this bound tends to

$$P_e(\mathbf{x} \rightarrow \mathbf{x}'|\mathbf{x}) \rightarrow \frac{1}{2} \frac{1}{\sqrt{\frac{2}{P} \frac{E_b}{N_0} \sigma_a^2 \frac{\theta(\mathbf{x}, \mathbf{x}')}{d_E^2}}} \exp\left(-\frac{\eta_a^2 d_E^4}{8\sigma_a^2 \theta(\mathbf{x}, \mathbf{x}')}\right). \quad (3.68)$$

According to this,  $P_e(\mathbf{x} \rightarrow \mathbf{x}'|\mathbf{x}) \rightarrow 0$  when  $E_b/N_0 \rightarrow \infty$ , but the simulations will show the appearance of an error floor whose value decreases as  $K$  grows. To obtain a bound for this error floor we need to refine the calculations. In fact, when  $E_b/N_0 \rightarrow \infty$ , the right hand side of the inequality in Eq. (3.57), which depends only on the noise and not on the fading, becomes negligible with respect to the value of the left hand side<sup>3</sup>. In this case, the EEP will depend only with good approximation on the fading amplitudes, and, dropping the right hand side, the inequality in Eq. (3.57) becomes

$$2 \sum_{n=m}^{L+m-1} a_n \eta_a x_n (x_n - x'_n) = C < \sum_{n=m}^{L+m-1} \eta_a^2 (x_n^2 - x_n'^2). \quad (3.69)$$

Again, the left hand side of this inequality can be approximated by a Gaussian RV  $C$  for  $L \geq 5$ , with mean

$$\eta_C = E_{\mathbf{a}} [2a_n \eta_a x_n (x_n - x'_n)] = 2\eta_a^2 \sum_{n=m}^{L+m-1} x_n (x_n - x'_n), \quad (3.70)$$

and variance

$$\sigma_C^2 = E_{\mathbf{a}} \left[ \left( 2 \sum_{n=m}^{L+m-1} a_n \eta_a x_n (x_n - x'_n) - \eta_C \right)^2 \right] = 4\eta_a^2 \sigma_a^2 \sum_{n=m}^{L+m-1} x_n^2 (x_n - x'_n)^2. \quad (3.71)$$

Now, taking into account that [Haykin, 2001]

$$P(C < z) = 1 - \frac{1}{2} \operatorname{erfc} \left( \frac{z - \eta_C}{\sqrt{2}\sigma_C} \right), \quad (3.72)$$

the EEP, after some algebra, will tend to

$$\begin{aligned} P_{e_{floor}}(\mathbf{x} \rightarrow \mathbf{x}'|\mathbf{x}) &\rightarrow 1 - \frac{1}{2} \operatorname{erfc} \left( -\frac{\eta_a d_E^2}{2\sigma_a \sqrt{2\theta(\mathbf{x}, \mathbf{x}')}} \right) = \\ &= \frac{1}{2} \operatorname{erfc} \left( \frac{\eta_a d_E^2}{2\sigma_a \sqrt{2\theta(\mathbf{x}, \mathbf{x}')}} \right), \end{aligned} \quad (3.73)$$

where  $\theta(\mathbf{x}, \mathbf{x}')$  is again defined as in Eq. (3.63).

With respect to the bit error probability, the analysis is more involved in this case with respect to the perfect CSI case, since, as seen in Eqs. (3.66) and (3.73), the

<sup>3</sup>The power of the signal  $x_n$  remains constant and so  $n_n \rightarrow 0$  for high signal to noise ratio.

event error probability deepens on the specific values of  $x_n$  through  $\theta(\mathbf{x}, \mathbf{x}')$ , not only on the squared Euclidean distance between the samples  $x_n$  and  $x'_n$ . Nevertheless, for the intermediate values of  $E_b/N_0$ , we will see by means of the BSM example that the bound of Eq. (3.66) calculated with the error events associated with the minimum of  $d_E^2$  can be tight enough, and the same will happen with the bound of Eq. (3.73) for high values of  $E_b/N_0$ . We have seen that these error events for the BSM chaos coded modulator involve only a bit on error and they have length  $L = Q + 1$ . In this case, we have  $2^Q$  equiprobable possible values for the initial value  $x_m$  in the error loop, and  $2^Q$  possible sequences  $x_{m+1}, \dots, x_{Q+m}$  for each  $x_m$ , given by the  $2^Q$  possible values of  $b_{m+1}, \dots, b_{Q+m}$ . The erroneous sequence  $x'_n, n = m, \dots, L+m-1$  is completely determined by  $x_n$  and the Hamming weight 1 binary error event. The sequences  $\mathbf{x} = (x_m, \dots, x_{L+m-1})$  are equiprobable for an *iid*  $b_n$  sequence, and so the average bit error probability for error events with  $d_E^2 = d_{\min}^2$  could be calculated as

$$P_b \approx \beta_{\min} \frac{1}{2} \sum_{\mathbf{x} \in \mathbf{X}} \frac{1}{2^{2Q}} \frac{1}{\sqrt{1 + \frac{2}{P} \frac{E_b}{N_0} \sigma_a^2 \frac{\theta(\mathbf{x})}{d_{\min}^2}}} \exp \left( -\frac{\frac{1}{4P} \frac{E_b}{N_0} \eta_a^2 d_{\min}^2}{1 + \frac{2}{P} \frac{E_b}{N_0} \sigma_a^2 \frac{\theta(\mathbf{x})}{d_{\min}^2}} \right), \quad (3.74)$$

where  $\mathbf{X}$  is the set of the  $2^{2Q}$  possible sequences of length  $Q$  under the constraints imposed by the chaos coded modulation. Again,  $\beta_{\min} = 1$  and  $d_{\min}^2 = 4/3$ . Note that

$$\theta(\mathbf{x}) = \sum_{n=m}^{m+Q-1} \frac{1}{4^{m+Q-1-n}} x_n^2 \quad (3.75)$$

now depends only on  $\mathbf{x}$ , since the difference between the unequal values of  $x_n$  and  $x'_n$  is a power of  $1/4$  for these error patterns in the BSM encoder. When we are in the error floor region, we can also calculate the average bound by assuming the same kind of error events, and accordingly

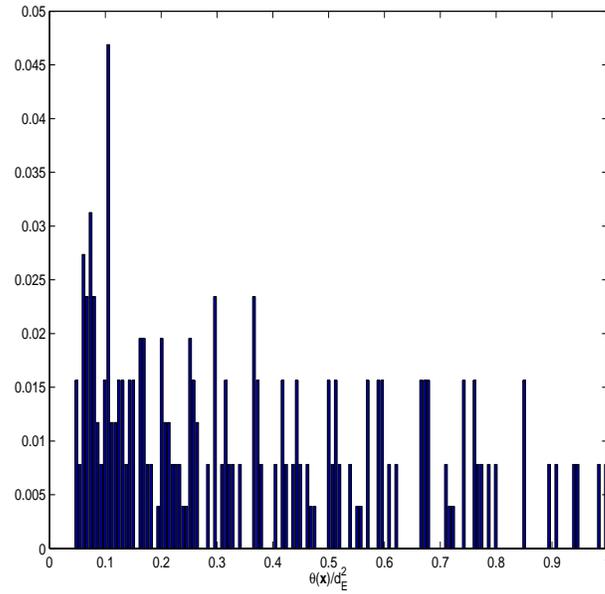
$$P_{b_{\text{floor}}} \approx \beta_{\min} \frac{1}{2} \sum_{\mathbf{x} \in \mathbf{X}} \frac{1}{2^{2Q}} \operatorname{erfc} \left( \frac{\eta_a d_{\min}^2}{2\sigma_a \sqrt{2\theta(\mathbf{x})}} \right), \quad (3.76)$$

where  $\mathbf{x}$ ,  $\beta_{\min}$ ,  $d_{\min}^2$  and  $\theta(\mathbf{x})$  take the same values as before.

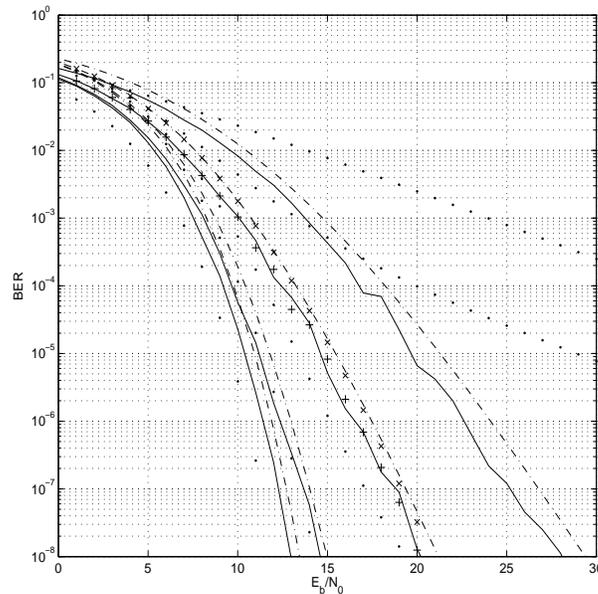
In Fig. 3.11 we have depicted the normalized histogram of the  $2^{2Q}$  possible values of  $\theta(\mathbf{x})/d_E^2$  in the case when  $Q = 5$  for the error patterns with  $d_E^2 = d_{\min}^2$ . We have verified that, with growing  $Q$ , the histogram tends to approximate an exponential decay between a minimum value around 0.05 and a maximum value around 1.0 (the maximum possible value of  $\theta(\mathbf{x})$  is  $d_{\min}^2$  because  $|x_n| \leq 1$ ). In any case, there is a concentration of values not negligible far from the minimum value, so that, as in the bound given for the ISI channel, the best approximation can only be given with the average bound, not only with the absolute minimum of  $\theta(\mathbf{x})/d_E^2$ .

### 3.4.3. Simulation results

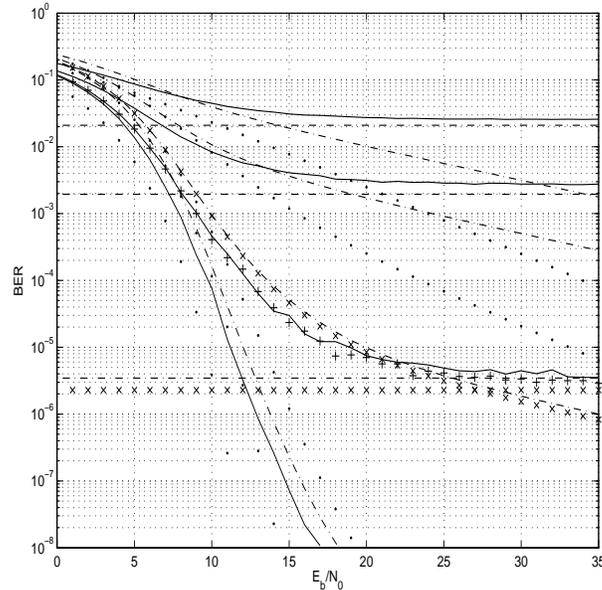
In Figs. 3.12 and 3.13 we can see the simulation results for several  $Q$  and  $K$  parameters, together with the proposed bounds. In all cases, we have simulated



**Figure 3.11.** Normalized histogram of  $\theta(\mathbf{x})/d_E^2$  for  $Q = 5$  in the BSM CCM system for the error patterns under evaluation.



**Figure 3.12.** Results for the case with CSI. From left to right,  $K = 50, 20, 5, 0$ . The simulation results for  $Q = 4$  are shown with continuous lines, the corresponding bounds with dash-dotted lines. The simulation result and the bound for  $Q = 6$  and  $K = 5$  are shown with '+' and 'x' respectively. Dotted lines represent the performance of BPSK, from left to right:  $K = \infty, 50, 20, 5, 0$ .



**Figure 3.13.** Results for the case without CSI. From left to right:  $K = 50, 20, 5, 0$ . The simulation results for  $Q = 4$  are shown with continuous lines, the corresponding bounds with dash-dotted lines. The simulation result and the bounds for  $Q = 5$  and  $K = 20$  are shown with '+' and 'x' respectively. Dotted lines represent the performance of BPSK, from left to right:  $K = \infty, 50, 20, 5, 0$ .

information frames of size  $N = 10000$  and we have calculated the BER after a minimum of 100 blocks with errors. The samples of the Rician process have been generated using the rejection method [Press et al., 1992]. The bit error probabilities of BPSK for several cases are shown for comparison, according to Eqs. (3.36) and (3.37).

In the CSI case (Fig. 3.12), we see that the BSM CCM exhibits a potential coding gain with respect to BPSK for  $K < \infty$ . Moreover, the bounds calculated with the  $d_{\min}^2$  events are tight enough to give reason of the BER slope for high  $E_b/N_0$ . We can also see that the influence of  $Q$  is small, both for the simulation results and for the bounds, which we know is a desirable feature in this kind of chaos coded modulations, where we are interested in keeping the coding gain with the lowest possible complexity.

On the other side, when there is no CSI (Fig. 3.13), there is no coding gain in any case with respect to uncoded BPSK. Nevertheless, we can see that the proposed bounds for the intermediate  $E_b/N_0$  region are tight enough, and that the error floor is appropriately described by the error floor bound<sup>4</sup>. As with the CSI case, we can verify that the influence of  $Q$  is small both for the simulations and for the bounds. Note that the  $Q = 4$  case is the minimum we can have to match the assumptions made while developing the bounds.

<sup>4</sup>The error floor bound for  $K = 50$  does not appear since it lies below  $10^{-9}$ .

### 3.5. Concluding remarks

Throughout this chapter, we have analyzed the performance of one of the most simple chaos-based encoding methods when the communication channel includes, besides AWGN, some amount of dispersive impairment. First of all, we have revised the encoding framework and we have limited ourselves to piecewise linear discrete maps. Though we have only described in detail the case of BSM, there is a wide range of chaos coded modulation systems, both based on chaotic maps or not [Kozic et al., 2006], that can be designed and analyzed under the same principles. These chaos coded modulations (CCM) have the special feature, compared to what was seen in Chapter 2, that the same trellis description of the resulting coded modulation is used at the encoder side and at the decoder side, in complete analogy with convolutional encoders or trellis coded modulated systems. According to this new framework, we have also proposed a MAP decoding method matched to CCM that will allow us to straightforwardly employ CCM within concatenated systems, as we will see in the subsequent chapters.

For the time invariant frequency selective impairment (ISI), we have drawn possible bounds for the BSM CCM, and we have shown by simulation that these bounds can explain reasonably well the behaviour of the BER when  $E_b/N_0 \rightarrow \infty$  if the ISI level is low enough that the error events still consist on input binary error events of limited Hamming weight. This has the positive consequence that such bounds can help in design tasks. Moreover, we have found out that the BER of the BSM CCM outperforms uncoded BPSK in the cases when equalization is not mandatory (low and moderate ISI), which is the opposite situation with respect to the AWGN channel. This means that CCM can offer potentially interesting properties in this kind of dispersive channels, even when the same CCM may be not so good in the absence of dispersive phenomena.

The same circumstances are emphasized by the results of BSM CCM in presence of frequency non-selective time varying impairment (flat fading). In this case, we have also drawn bounds for the situations when there is side information at the decoder, and when there is not, and we have verified that they can match the simulation results with the needed degree of accuracy. Apart from allowing us to gain insight into the behaviour of the chaotic encoding system, they offer again potential application in design tasks. When we have CSI, the BER with BSM CCM is again better than with uncoded BPSK, and so we verify that this kind of dispersion may possibly be managed with CCM schemes. Nonetheless, the case without CSI is catastrophic for the BSM CCM, and this points towards the need for more refined schemes if we want to use CCM in this kind of environment.

It is evident that ISI itself can be successfully managed by a number of equalization strategies [Li et al., 1995; Proakis, 2001] and that against fading we can use diversity techniques [Goldsmith, 2005], but the main point of this chapter was not to search for additional improvements, but only to show what kind of channels could be best suited to take advantage of the chaos-based encoding systems. In fact, there had been a commonplace for a long time that chaos could do well in dispersive envi-

ronments, but the poor results of CCM and related systems in AWGN had for some time prevented further research on this topic [Abel and Schwarz, 2002; Dmitriev et al., 2003], and therefore the main stress had been placed in multiple access systems or other kind of chaos-based modulations [Kennedy et al., 2000a; Setti et al., 2004]. With the evidence that CCM can be useful in usual dispersive channel, we will explore in the next chapters the possibility of designing more efficient higher dimensional chaos-based systems using the known principles of concatenated coding [G. D. Forney, 1966].

## Chapter 4

# Serially Concatenated Chaos Coded Modulations

### 4.1. Introduction

In Chapter 3 we have introduced a framework for chaos coded modulations which could be associated with an equivalent trellis encoder view analogous to Ungerboeck's trellis coded modulation (TCM) [Ungerboeck, 1982]. We have shown with bounds and simulations that this CCM scheme can offer some degree of robustness in dispersive channels, but the final performance of the example chosen was not good enough to lead to practical systems. One of the paths recently opened to improve the results of CCM in the AWGN channel has been the increasing of the chaotic map dimensionality and the expansion of the framework to chaotic systems without an underlying chaotic map [Schimming and Hasler, 2003; Kozic and Schimming, 2005; Kozic et al., 2006], together with the design of new decoding strategies [Kozic and Hasler, 2006]. Other possible way to increase the dimensionality and potential performance of the chaotic system is concatenation [G. D. Forney, 1966], and this is the alternative we will follow in this and the next chapter.

The task of designing concatenating systems with CCM is easy to address because the TCM equivalence of CCM allows us to evaluate this kind of chaos coded modulations in the same environments where the other usual coded modulations have found application. This offers the additional advantage face to old approaches in chaos-based communications that a number of well known and established tools from digital communications theory will be at hand to help in the design and evaluation tasks. On the other side, we can take advantage of the fact that serially concatenated channel codes (SCCC) [Benedetto et al., 1998; Schlegel and Pérez, 2004] and serially concatenated trellis coded modulations (SCTCM) [Divsalar and Pollara, 1997; Narayanan and Stüber, 1999; Altunbas and Narayanan, 2001; Tullberg and Siegel, 2001; Pfister and Siegel, 2003; Tullberg and Siegel, 2005a; Tullberg and Siegel, 2005b; Mitra and Lampe, 2006; Howard and Schlegel, 2006] have shown good results in AWGN and fading channels [Vucetic, 1993; Yuan et al., 2002; Gulati and Narayanan, 2003; Schlegel and Pérez, 2004; Goldsmith, 2005], because they can offer a robust solution where the changes in the channel does not affect much at the decoder side [Biglieri et al., 1998]. This is due to the fact that serial concate-

nation somewhat uncouples the squared Euclidean distance spectrum of the inner encoder and the Hamming distance spectrum of the outer encoder and increases this latter at the expense of a little reduction on the former. The serial concatenation of a channel encoder and a CCM can also be compared to bit-interleaved coded modulation (BICM) [Caire et al., 1998], though in this case the inner encoder is not properly speaking a trellis encoder. Nevertheless, we will see that they have several interesting analogies, like the dramatic improvement achieved by iterative decoding [Li and Ritchey, 1997]. This will allow us to exploit in our analysis some developments originally intended for BICM.

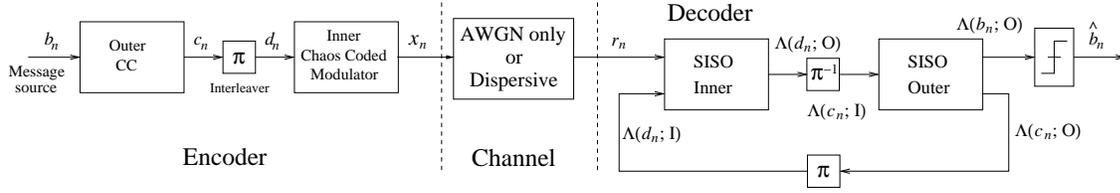
According to this and with the aim of further exploiting the good properties of CCM in dispersive channels, we can think of modifying the SCTCM system by replacing the TCM usually employed as inner encoder by a CCM. Such concatenated systems where the coded modulation is replaced by a chaos coded modulation were proposed in [Escribano et al., 2005; Escribano et al., 2006c], and the results under iterative decoding assessed the good possibilities of this new approach with respect to traditional chaos-based systems, specially when some conditions are met in the CCM module.

These are the reasons why we comprehensively address in this chapter the concatenated systems with binary convolutional channel codes and CCM joined by means of a bit interleaver, and show how we can study their behaviour in AWGN, ISI and frequency non-selective fading channels with the help of known developments in SCTCM or BICM. We will show that we can reach a good performance with low complexity outer channel codes and with the simplest CCM already used in Chapter 3. This, together with the characteristics of the chaotic signal in the channel, which has the advantage of being easily generated [Lau and Tse, 2003], allows us foresee potential applications for these systems.

Accordingly, the chapter is structured as follows. In Section 4.2 we will look into the concatenated system and its particularities, including the description of the concatenated encoder, the channel model and the iterative decoder. Section 4.3 addresses the convergence analysis of the decoding algorithm by means of the extrinsic information transfer chart tool. In Section 4.4 we will explain how to draw bounds for the bit error probability under the assumption of a binary input-output symmetric channel. Section 4.5 shows the simulation results and validates the bounds and predictions of Section 4.3. Section 4.6 is devoted to the conclusions.

## 4.2. System model

As stated, this chapter deals with an extension of SCTCM, where we make use of chaos coded modulations (CCM) instead of the usual coded modulations based on trellis codes or the like [Caire et al., 1998; Escribano et al., 2006c]. According to this, we will call this system serially concatenated chaos coded modulations (SCCCM). The scheme of the complete system can be seen in Fig. 4.1, and we will describe it with the needed detail in the next subsections.



**Figure 4.1.** Block diagram of the serially concatenated encoder, the channel and the iterative decoder.

### 4.2.1. Serially concatenated encoder

On the left hand side of Fig. 4.1, we have depicted the serially concatenated encoder. We have as outer encoder a binary convolutional code (CC) of rate  $R$ , which accepts as input an independent and identically distributed (*iid*) binary sequence  $b_n$ , and produces a convolutionally encoded binary sequence  $c_n$ . As the chaos coded modulator will work with input bit blocks of size  $N$ , and as we perform trellis termination for the convolutional encoder [Proakis, 2001], each input block for the inner encoder will be produced by a binary sequence  $\mathbf{b} = (b_1, \dots, b_D)$  of size  $D = R \cdot N - \nu$ , where  $\nu = m + 1$  is the constraint length of the convolutional encoder, and  $m$  is its memory length [Lin and Costello, Jr., 2004].

In analogy with the model of SCTCM, we introduce an interleaver between the outer and the inner encoders. This interleaver performs a permutation on the CC output sequence  $c_n$ , so that the input CCM data  $d_n$  is handed to the inner encoder in a different order with respect to  $c_n$ . Since a typical  $N_1 \times N_2$  block interleaver<sup>1</sup> leads almost always to a bad behaviour due to its regularity when no termination is applied to the inner encoder [Hokfelt et al., 2001], the interleaver considered here will be an S-random interleaver [Divsalar and Pollara, 1995; Dolinar and Divsalar, 1995], where the permutation function is chosen in a semi-random basis. This function will map each index  $j$  into an index  $\pi(j)$ , which means that the bit in position  $j$ ,  $c_j$ , at the output of the outer encoder will be taken as bit in position  $\pi(j)$ ,  $d_{\pi(j)}$ , at the input of the inner encoder. The index permutation is chosen according to the following algorithm:

- Choose an integer  $S$ .
- For index  $\pi(j)$  corresponding to position  $j$  draw a random number  $t$  between 1 and  $N$ .
- If  $t$  has not been chosen before, verify if the  $S$  previously chosen indexes lie at least at a distance of  $S$  from  $t$ , i.e.,  $|\pi(p) - t| > S$  for  $p = j - S, \dots, j - 1$ .
- If  $t$  satisfies the conditions, keep it as  $\pi(j) = t$  and proceed until all  $N$  indexes are chosen.

<sup>1</sup>We understand as  $N_1 \times N_2$  block interleaver an interleaver consisting in an  $N_1 \times N_2 = N$  matrix where the data is written columnwise and read rowwise or viceversa.



where  $\bar{d} = d \otimes 1$  and  $\otimes$  is the binary XOR operation. The background for this becomes more clear if we look into the trellis encoder view of this CCM [Kozic et al., 2006], which can be developed through the associated symbolic dynamics as described in the previous chapter [Schweizer and Schimming, 2001a; Schweizer and Schimming, 2001b]. In fact, when  $z_0 = 0$ , this encoder based upon the BSM can be again seen as a shift register with  $Q$  memory positions storing  $Q$  successive values of  $g(d_n, z_n)$ , and where the output  $z_n$  is calculated as (see Fig. 4.2)

$$\begin{aligned} r_i &= r_{i-1} \quad i = Q, \dots, 2, \\ r_1 &= g(d_n, z_{n-1}), \\ z_n &= \sum_{i=1}^Q 2^{-(Q+1-i)} \cdot r_i, \end{aligned} \tag{4.5}$$

or, equivalently

$$z_n = \sum_{i=1}^Q 2^{-(Q+1-i)} \cdot g(d_{n-i+1}, z_{n-i}), \tag{4.6}$$

where  $g(d_n, z_{n-1}) = d_n \otimes r_Q$  (note that this is equivalent to Eq. (4.4)). If we compare this with Eq. (3.4), the only difference is that the associated register stores values of  $g(d_n, z_{n-1})$  instead of the binary data  $d_n$ . Though here we focus only on the simple case of the BSM to illustrate the potential properties of SCCCM, there is again a whole kind of chaos coded modulators based upon the same principles that could be employed in this same framework and that could be described by an equivalent trellis encoder, and thus decoded with known sequence estimation techniques [Kozic, 2006]. Note also that, since  $g(d, z)$  is equivalent to the precoder defined by the polynomial  $1 + D^{Q-1}$ , it will always be possible to include such kind of simple recursive precoder (rate 1 accumulate code) before the chaos coded modulator, as is usually done in turbo-equalization [Koetter et al., 2004; Narayanan, 2001], or in serial concatenation of convolutional encoders and TCM [Tullberg and Siegel, 2005a; Tullberg and Siegel, 2005b], in order to preserve the interleaver gain. In the context of chaos coded modulations with serially concatenated schemes, the key role of feedback on the inner encoder was made evident in [Escribano et al., 2005; Escribano et al., 2006c].

Thanks to this precoding, we can avoid the situation where a binary error event of weight  $d$  in the convolutional encoder, after being interleaved and assuming that the error bits are scattered far apart enough<sup>2</sup>, only induces  $d$  1-bit minimum distance events in the BSM CCM (see Subsection 3.2.1 in the last chapter). It is straightforward to show that, in the AWGN channel, such concatenation of error events within the CCM would induce an error event probability  $P_e(\mathbf{x} \rightarrow \mathbf{x}'|\mathbf{x})$  given by [Proakis, 2001]

$$P_e(\mathbf{x} \rightarrow \mathbf{x}'|\mathbf{x}) \propto \operatorname{erfc} \left( \sqrt{Add_{\min}^2} \right), \tag{4.7}$$

<sup>2</sup>Which could be most probable when  $d$  is small, i.e., for the most critical binary error events in the CC.

where  $A$  is a factor depending on the signal to noise ratio, the power of the CCM signal and the CC rate,  $\mathbf{x}$  is the CCM word actually sent, and  $\mathbf{x}'$  is the erroneous decoded word corresponding to the binary error event of weight  $d$ . As usual,  $d_{\min}^2$  is the minimum Euclidean distance of the BSM encoder. This probability could be very high due to the poor minimum distance properties of BSM. With the precoder  $1 + D^{Q-1}$ , the error events leading to  $d_{\min}^2$  in the BSM CCM will consist in binary events of length  $L = Q + 1$  with two bit errors,  $d_n, d_{n+1}, \dots, d_{n+Q-1}, d_{n+Q}$  vs  $d_n^*, d_{n+1}^*, \dots, d_{n+Q-1}^*, d_{n+Q}^*$ , with  $d_i \neq d_i^*$ . With a good interleaver design, i.e., a high value of  $S$  in the S-random interleaver, any binary error event of weight  $d$  in the convolutional encoder could be scattered so efficiently that this undesirable situation is not met with higher probability than in the case without precoding.

### 4.2.2. Channel

Once modulated, the samples  $x_n$  are sent to the channel in baseband. Note that  $x_n$  is a broadband signal that, contrary to the usual coded modulation schemes employed in SCTCM or BICM, does not intend to provide spectral efficiency. Though each  $x_n$  conveys information about  $Q$  bits, we send all  $x_n$  at the rate of 1 sample per bit. In the channel, shown in Fig. 4.1 as a black box,  $x_n$  is subjected at least to the effects of additive white Gaussian noise (AWGN) and possibly other sources of distortion, according to the channel models seen in Chapter 3. We will thus consider three kind of channels:

1. AWGN channel. In this simple case, the sequence arriving at the decoder side,  $\mathbf{r} = (r_1, \dots, r_N)$ , will be

$$r_n = y_n + n_n = x_n + n_n, \quad (4.8)$$

where  $y_n = x_n$ , and  $n_n$  is the usual *iid* sample of a Gaussian RV with zero mean and power  $\sigma^2$ .

2. ISI channel. It will be characterized, as in the last chapter, by means of a normalized FIR filter of length  $2M + 1$  and coefficients  $\mathbf{h} = (h_{-M}, \dots, h_M)$ .  $r_n$  will then be (see Fig. 3.3)

$$r_n = y_n + n_n = \sum_{m=-M}^M h_m x_{n+m} + n_n. \quad (4.9)$$

The filter coefficients will be those of Table 3.1.

3. Fading channel. It will follow the model of the last chapter as well, so that the chaotic signal will be affected by frequency-non selective (flat) uncorrelated fading. The fading amplitude samples  $a_n$  follow uncorrelated Rician RVs with parameter  $K$  and unit power.  $r_n$  will then be (see Fig. 3.9)

$$r_n = y_n + n_n = a_n x_n + n_n. \quad (4.10)$$

Note that, as the CC has rate  $R$ , the relationship between the power of the noise and the signal to noise ratio in terms of bit energy to noise spectral density will now be

$$\sigma^{-2} = 2 \frac{R E_b}{P N_0}, \quad (4.11)$$

where  $P$  is the power of the chaos coded modulated signal. This relationship holds for all channels mentioned.

### 4.2.3. Iterative decoder

Each block of  $N$  samples  $\mathbf{r} = (r_1, \dots, r_N)$  is finally decoded by means of an iterative decoder, shown on the right side of Fig. 4.1. This iterative decoder is based on two soft-input soft-output (SISO) modules [Benedetto et al., 1997] linked by means of a corresponding interleaver  $\Pi$  and a corresponding deinterleaver  $\Pi^{-1}$ . The SISO module for CCM is the adaptation of the usual SISO module implementing the MAP BCJR forward-backward algorithm [Bahl et al., 1974] that takes advantage of the equivalent trellis encoder description of the chaos coded modulation and the underlying symbolic dynamics [Escribano et al., 2005]. This module was described in Chapter 3. Again, since we do not perform any trellis termination in the inner encoder, the ending state of the data block for the inner SISO will be unknown, and the backward calculation of the MAP algorithm will be performed after initializing  $\beta_N(\cdot)$  equiprobably [Benedetto et al., 1997] (see Subsection 3.2.2). Future research should include the possible enhancement attainable by the use of trellis termination techniques in the inner chaos coded modulator, though a good design of the interleaver and a high value of  $N$  could be enough to make the differences between trellis termination and non-trellis termination negligible [Hokfelt et al., 2001].

The SISO module for the CC is as presented in [Benedetto et al., 1997] without any further change, and it takes into account the fact that the trellis of the convolutionally coded sequence is terminated. Each run of the iterative decoding process (see Fig. 4.1, where the parameters O and I in the log probability ratios stand for output and input, respectively) will consist first in a calculation of the output log probability ratios

$$\Lambda(d_n; \text{O}) = \log \frac{P(d_n = 1 | \mathbf{r}, \Theta)}{P(d_n = 0 | \mathbf{r}, \Theta)}, \quad (4.12)$$

as a function of the channel output  $\mathbf{r}$  and the input *a priori* log probability ratios  $\Lambda(d_n; \text{I})$ . We have included the possible knowledge of the channel parameters  $\Theta$  in Eq. (4.12) for the case where the channel state is known at the decoder and the algorithm metrics take it into account. After being deinterleaved, the log probability ratios  $\Lambda(c_n; \text{I})$  serve as input *a priori* information for the outer SISO, which gives as output the log probability ratios for the input binary message  $\Lambda(b_n; \text{O})$  and for the convolutionally coded sequence  $\Lambda(c_n; \text{O})$ . These last ones are interleaved and used as input log probability ratios  $\Lambda(d_n; \text{I})$  for the next iterative decoding run.

As presented in [Benedetto et al., 1997], the SISO modules have two possible inputs and two possible outputs, but not all of them are needed in the serially

concatenated setup. In the case of the inner SISO, the output probability estimations of the chaos coded modulated sequence  $p(x_n|\mathbf{r}, \Theta)$  are not needed. For the outer SISO, since  $b_n$  is an *iid* sequence, the input log probability ratios  $\Lambda(b_n; \mathbf{I})$  are 0 throughout all the iterative decoding process.

The AWGN channel and the ISI channel employ the same metrics, calculated as functions of  $(r_n - x_n)^2$ , and no knowledge of the channel is taken into account<sup>3</sup>, so that the extrinsic information at the output of the inner SISO can be simply expressed as

$$\Lambda(d_n; \mathbf{O}) = \log \frac{P(d_n = 1|\mathbf{r})}{P(d_n = 0|\mathbf{r})}. \quad (4.13)$$

In the fading channel without CSI, the metrics are calculated as functions of  $(r_n - \eta_a x_n)^2$  [Hall and Wilson, 1998] and so the inner SISO module provides the extrinsic information in form of log probability ratios

$$\Lambda(d_n; \mathbf{O}) = \log \frac{P(d_n = 1|\mathbf{r}, \eta_a)}{P(d_n = 0|\mathbf{r}, \eta_a)}. \quad (4.14)$$

Finally, for the fading channel with perfect CSI and metrics calculated as functions of  $(r_n - a_n x_n)^2$ , we will have

$$\Lambda(d_n; \mathbf{O}) = \log \frac{P(d_n = 1|\mathbf{r}, \mathbf{a})}{P(d_n = 0|\mathbf{r}, \mathbf{a})}, \quad (4.15)$$

where  $\mathbf{a} = (a_1, \dots, a_N)$  is the vector of fading amplitudes. After several decoding iterations through the inner and outer SISO modules, we get the estimated decoded sequence  $\hat{b}_n$  by hard deciding over the log probability ratios  $\Lambda(b_n; \mathbf{O})$  as indicated in Eq. (3.16).

### 4.3. Convergence analysis

In this section we study the convergence behaviour of the iterative decoding algorithm for the proposed channels. A powerful tool to look into the convergence of the algorithm as a function of channel distortion is the so called EXtrinsic Information Transfer (EXIT) charts [ten Brink, 2001]. They have proved to be useful as design tools not only in the context of binary turbocodes or in serial concatenation of binary codes, but also with SCTCM [Howard and Schlegel, 2006] and turbo TCM (TTCM) systems [Chen and Haimovich, 2004; Kliever et al., 2006], in the evaluation of turbo equalized systems [Otnes and Tüchler, 2002] and with BICM [Li and Ritchey, 1997]. The EXIT charts are based on the computation of the mutual information [Cover and Thomas, 2006] of the log probability ratios at the input of the SISO module *versus* the mutual information of the log probability ratios at the output of the same SISO module after each decoding step. These mutual informations  $I$  are calculated

<sup>3</sup>Recall that at least the transition probabilities of the channel require knowledge about the noise power  $\sigma^2$ , but we drop the dependency for convenience since it affects all the cases.

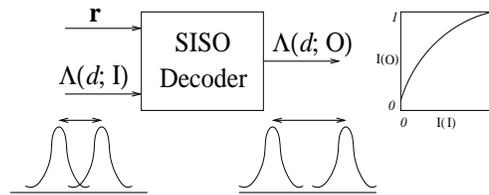
as [Proakis, 2001]

$$I = \frac{1}{2} \sum_{d=0,1} \int_{-\infty}^{\infty} p(\Lambda|d) \log_2 \left( \frac{2p(\Lambda|d)}{p(\Lambda|1) + p(\Lambda|0)} \right) d\Lambda, \quad (4.16)$$

where  $p(\Lambda|d)$  is the probability density function of the log probability ratios when the bit sent takes the value  $d = 0$  or  $d = 1$ . We have dropped the index in  $d$  for simplicity. In Fig. 4.3 we can see a scheme of the inner decoder where we have depicted the distribution of a set of input log probability ratios,  $\Lambda(d; I)$ , with two peaks corresponding to each case,  $d = 0$  and  $d = 1$ . If the decoding algorithm is working well for the level of distortion in the channel, the distribution of the output log probability ratios,  $\Lambda(d; O)$ , will have the two peaks more distinctly separated, which means a higher probability to decode without error, and also a growing value of the mutual information  $I$ , as shown in the related  $I(O)/I(I)$  curve located at the right hand side of the figure (O: output, I: input). Though Fig. 4.3 refers to the SISO module for the chaos coded modulation, the same principles are straightforwardly applicable to the SISO module for the convolutional code, just by renaming the input and output as in Fig. 4.1 and dropping the input  $\mathbf{r}$ , so that the mutual informations will be calculated over  $\Lambda(c; I)$  and  $\Lambda(c; O)$ , respectively. Note that the  $I(O)/I(I)$  transfer curve depends on the channel distortion level for the inner SISO module (it makes use of the channel output  $\mathbf{r}$  to calculate  $\Lambda(d; O)$ ), while it does not for the outer SISO module. This  $I(O)/I(I)$  curve is known as the mutual information transfer function of the decoder, and it relates the input and output mutual information values through

$$I(O) = T(I(I), \Theta), \quad (4.17)$$

where again the dummy vector  $\Theta$  emphasizes the possible dependence with the channel state.



**Figure 4.3.** Decoding behaviour of the SISO module from the mutual information point of view.

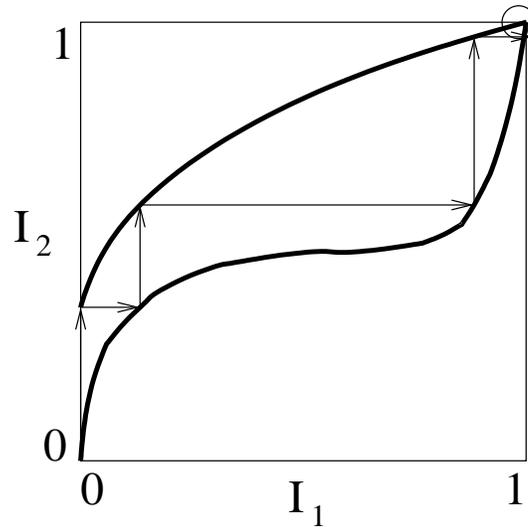
The mutual information values  $I$  defined in Eq. (4.16) will be calculated under the usual assumption of having Gaussian distributed input log probability ratios, and so we will feed into the SISO modules values of  $\Lambda(d; I)$  and  $\Lambda(c; I)$  drawn from a known Gaussian distribution and study the corresponding output values of  $\Lambda(d; O)$  and  $\Lambda(c; O)$  for the inner and outer SISO modules, respectively. These *a priori* inputs are generated as

$$\Lambda(d; I) = \mu_S d + n_S, \quad (4.18)$$

where  $n_S$  is a Gaussian RV with zero mean and variance  $\sigma_S^2$ , and  $\mu_S = \sigma_S^2/2$  [ten Brink, 2001]. Though written for  $\Lambda(d; \text{I})$ , the values of  $\Lambda(c; \text{I})$  for the output SISO are processed in the same way. Taking all this into account, the input mutual information can be finally calculated as [ten Brink, 2001]

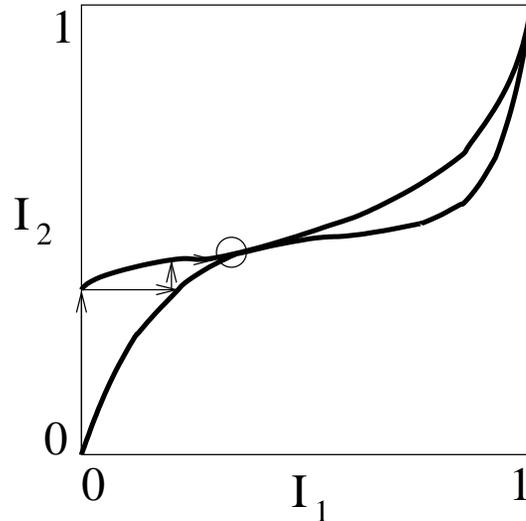
$$I(\text{I}) = 1 - \int_{-\infty}^{\infty} \frac{\exp(-(\Lambda - \sigma_S^2/2)^2/(2\sigma_S^2))}{\sqrt{2\pi}\sigma_S} \log_2(1 + \exp(-\Lambda)) d\Lambda. \quad (4.19)$$

This integral has to be calculated numerically [Abramowitz and Stegun, 1965].  $I(\text{O})$  will be calculated also through numerical integration, but after computing the histogram over the output log probability ratios  $\Lambda(d; \text{O})$  and  $\Lambda(c; \text{O})$ , and averaging over a number of runs.



**Figure 4.4.** EXIT chart and mutual information trajectory in iterative decoding. Iterations proceed as marked with arrows and reach the  $(1, 1)$  point (circled).

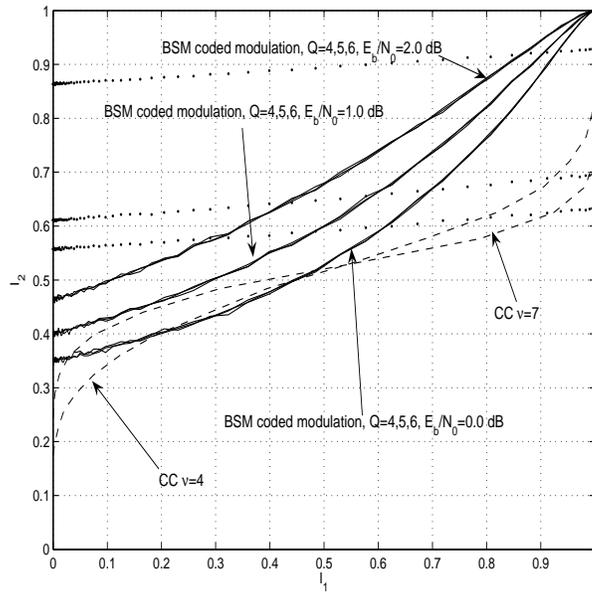
By combining the two  $I(\text{O}) = T(I(\text{I}), \Theta)$  transfer curves from the two SISO modules, we get the EXIT chart for the joint iterative decoder, as shown with boldlines in Figs. 4.4 and 4.5 in the case of two imaginary encoders. The upper curve is for the inner SISO, and the lower curve for the outer SISO.  $I_1$  is the input mutual information for the inner SISO module, and  $I_2$  is the corresponding output mutual information. At the same time, given that the output mutual information of the inner SISO module, after going through the deinterleaving stage, is the input mutual information for the outer SISO module,  $I_2$  is as well the input mutual information for the outer SISO module, and, conversely,  $I_1$  is the output mutual information for this same SISO module. The reason for this is that the deinterleaver and the interleaver do not change the input/output distributions of the log probability ratios, and their operation on the EXIT chart is a simple change in the meaning of the axis of the corresponding plot. In this way, the iterative decoding process starts over the  $(I_1 = 0, I_2 = T_{\text{inner}}(0))$  point on the upper curve (corresponding to the inner SISO),



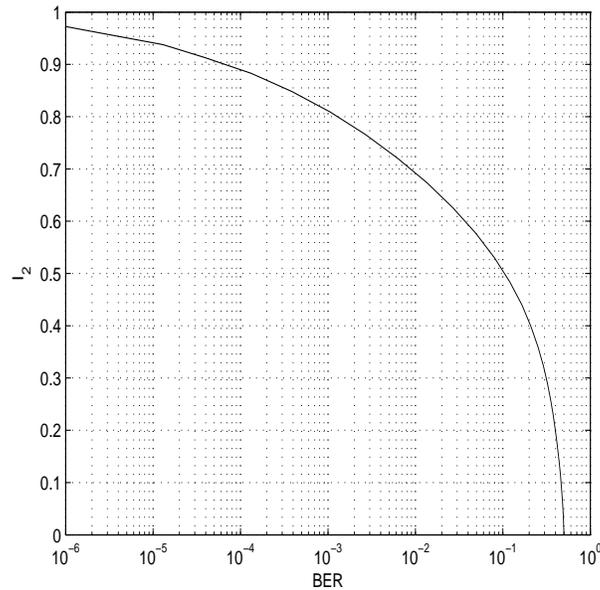
**Figure 4.5.** EXIT chart and mutual information trajectory in iterative decoding. Iterations proceed as marked with arrows and get stuck in a fixed point (circled) far from the  $(1,1)$  point.

i.e., when there is still no input mutual information. This  $I_2$  value is the input value for the lower curve, corresponding to the outer SISO, and we get at the output a corresponding value  $I_1 = T_{\text{outer}}(I_2)$  from this curve, which, in turn, will be the input value for the next step at the inner SISO and its corresponding curve. This process, illustrated with the help of arrows in Figs. 4.4 and 4.5, goes on till we reach a crossing point between the two curves where the algorithm cannot proceed further. This kind of EXIT charts and the iterative process they try to describe assume additionally that the output log probability ratios from one decoding step and the input log probability ratios for the next one are uncorrelated, which is not exactly true as we iterate, but this situation can be approached with a good interleaver design [ten Brink, 2001].

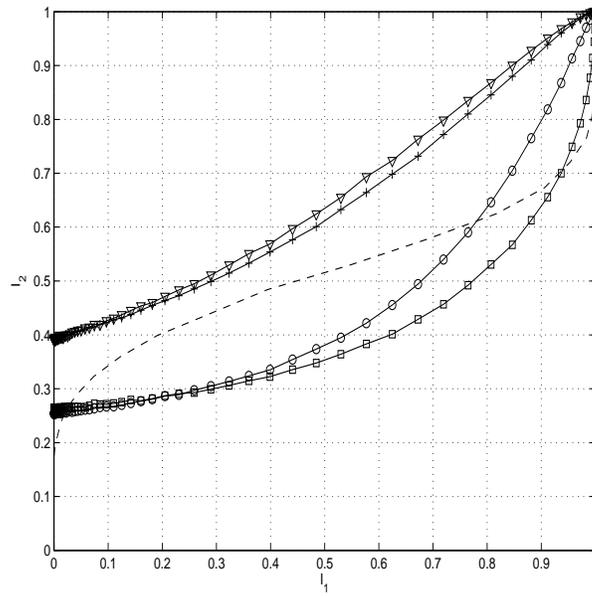
In Figs. 4.6, 4.8, 4.9 and 4.10 we have depicted several EXIT charts for the different kinds of channels considered. All the charts have been calculated by simulating data sequence blocks of 10000 samples and by averaging the results over 100 runs, both for the inner and the outer SISO. In Fig. 4.6, where we have plotted the EXIT chart for two convolutional encoders and for the BSM modulator in the AWGN channel, we see that the crossing point of the transfer curves is a function of the channel distortion level. If this final point is the  $(1,1)$  point, as is the case of the transfer curves of the BSM modulator for  $E_b/N_0 = 2.0$  dB with respect to the two CC transfer curves, the algorithm truly converges and we are in the region where the BER is dominated by the error floor, and where we need just a few number of iterations to reach this BER [ten Brink, 2001]. If the final point is not the  $(1,1)$  point (see trajectory of Fig. 4.5), as is the case of the BSM encoder transfer curves for  $E_b/N_0 = 0.0$  dB with respect to the transfer curves of both convolutional coders, the algorithm gets stuck at a fixed point leading to a higher value in the BER and a



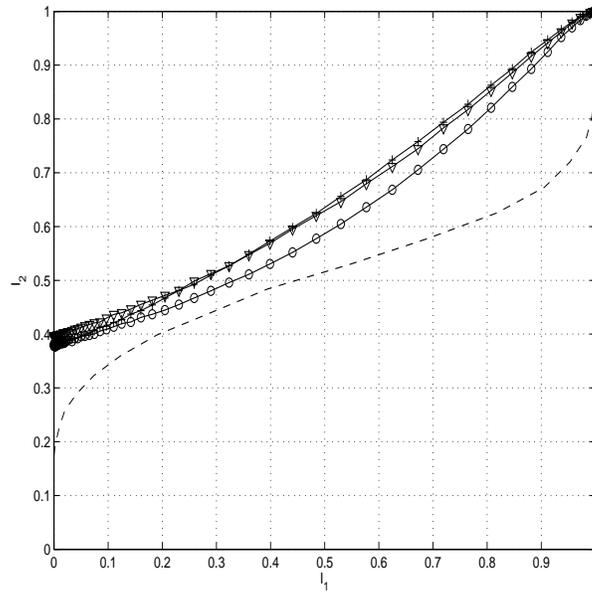
**Figure 4.6.** EXIT chart in the AWGN channel with two different convolutional encoders, and a BSM CCM with  $Q = 4, 5, 6$ . Dotted lines represents transfer curve for BSM without the  $1 + D^{Q-1}$  precoder, with  $Q = 5$  and three different signal to noise ratios:  $E_b/N_0 = 0.0$  dB (lower line),  $E_b/N_0 = 1.0$  dB (mid line) and  $E_b/N_0 = 5.0$  dB (upper line).



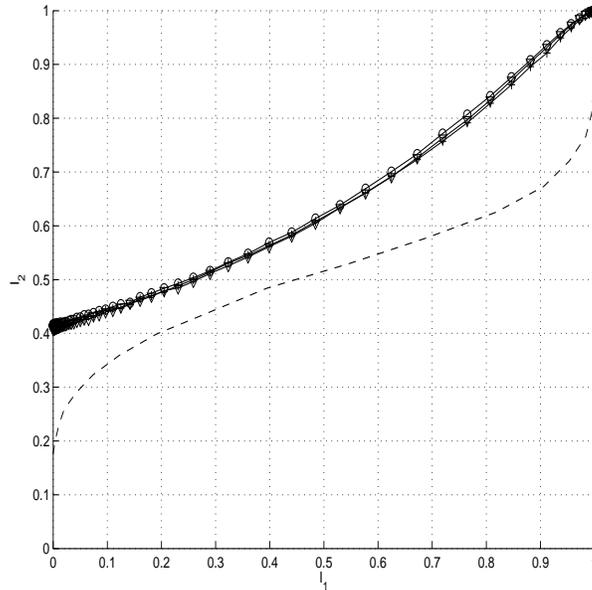
**Figure 4.7.** BER of the CC SISO decoder for the CC with  $\nu = 4$  as a function of its input mutual information  $I_2$ .



**Figure 4.8.** EXIT chart for the convolutional encoder with  $\nu = 4$  (dashed line), and BSM coded modulation with  $Q = 5$  for the ISI channel. '+' :  $E_b/N_0 = 1.0$  dB, low ISI. 'v' :  $E_b/N_0 = 1.5$  dB, moderate ISI. 'o' :  $E_b/N_0 = 1.5$  dB, high ISI. '□' :  $E_b/N_0 = 8.0$  dB, high ISI.



**Figure 4.9.** EXIT chart for the convolutional encoder with  $\nu = 4$  (dashed line), and BSM coded modulation with  $Q = 5$  for the fading channel, in the case where perfect CSI is available at the decoder. '+' :  $E_b/N_0 = 3.4$  dB,  $K = 0$ . 'v' :  $E_b/N_0 = 2.0$  dB,  $K = 5$ . 'o' :  $E_b/N_0 = 1.0$  dB,  $K = 20$ .



**Figure 4.10.** EXIT chart for the convolutional encoder with  $\nu = 4$  (dashed line), and BSM coded modulation with  $Q = 5$  for the fading channel, in the case where there is no CSI at the decoder. '+' :  $E_b/N_0 = 4.0$  dB,  $K = 0$ . '∇' :  $E_b/N_0 = 2.2$  dB,  $K = 5$ . 'o' :  $E_b/N_0 = 1.4$  dB,  $K = 20$ .

higher degree of uncertainty over the message sent, no matter the number of iterations performed. When the distortion in the channel is just low enough to allow the decoding process to snake through a bottleneck between the curves, we are in the waterfall or turbo cliff region [Schlegel and Pérez, 2004], where the algorithm starts to converge to the (1, 1) point and where there is normally an abrupt change in the BER slope.

The CC transfer curves of Fig. 4.6 have been drawn for two non-systematic non-recursive rate  $R = 1/2$  convolutional encoders of different complexity: an 8-state convolutional code with  $\nu = 4$  and generator polynomials  $1 + D^3$  and  $1 + D + D^3$  ( $d_{\text{free}} = 5$ ), and a 64-state convolutional code with  $\nu = 7$  and generator polynomials  $1 + D + D^4 + D^5 + D^6$  and  $1 + D^3 + D^4 + D^5 + D^6$  ( $d_{\text{free}} = 8$ ) [Lin and Costello, Jr., 2004]. With respect to the BSM modulator, we see that the mutual information transfer curves are the same as a function of  $E_b/N_0$  for three different quantization levels,  $Q = 4, 5, 6$ , which is a desirable property that hints to the relative independence of the performance on the  $Q$  parameter. This is good since we can just keep  $Q$ , and thus the encoding and decoding complexity, as low as possible in practice, while being able to study the system using the chaotic properties of the signal when  $Q \rightarrow \infty$  (at least whenever this can help to simplify the calculations). This also gives a hint that the results may be linked to the properties of the underlying map rather than to such ad-hoc control parameter. On the other hand, we can see that, for  $E_b/N_0 = 1.0$  dB, we reach full convergence to the (1, 1) point, at least theoretically, with the convolutional encoder of lower complexity ( $\nu = 4$ ), while we

would need additionally as much as some tenths of dB to be able to converge with the other convolutional encoder, as the BSM curve for this  $E_b/N_0$  almost intersect the convolutional encoder curve for  $\nu = 7$  at the point (0.20, 0.45). This is a known property of the serially concatenated systems with interleavers [Narayanan, 2001], and so we will prefer the convolutional encoder with lower complexity, but better convergence results.

Moreover, the recursive inner encoder leads to a worse behaviour in the first iterations, and a worse BER in the low  $E_b/N_0$  region, than a non-recursive one [Narayanan, 2001], as can be seen by comparing the transfer curves of the non-recursive BSM CCM with the recursive ones for the same  $E_b/N_0$  (see Fig. 4.6). Nevertheless, with the recursive setup, once we are immediately above the threshold and thus in the BER waterfall region, the performance is much better and we get a higher coding gain [Otnes and Tüchler, 2002]. This is related to the fact that the non-recursive BSM CCM transfer function does not tend to the (1, 1) point for the signal to noise ratios of interest. In Fig. 4.7 we have plotted the BER of the SISO for the CC with  $\nu = 4$  as a function of its input mutual information  $I_2$ . We see that, for  $E_b/N_0 = 0.0$  dB, the transfer function of the recursive BSM CCM intersects the transfer curve of the CC at the point (0.2, 0.4), thus giving a BER of about  $2 \cdot 10^{-1}$ , while the non-recursive BSM CCM can proceed till the point (0.8, 0.6), corresponding to a BER of  $3 \cdot 10^{-2}$ . The situation changes abruptly for  $E_b/N_0 = 1.0$  dB, since the recursive BSM CCM is in fact in the waterfall region, but the non-recursive BSM CCM transfer curve gets stuck still at (0.9, 0.7), with a BER of around  $10^{-2}$ . Even when  $E_b/N_0 = 5.0$  dB, the non-recursive BSM CCM does not still reach full convergence, and we get stuck at around (0.99, 0.92), with a BER of  $2 \cdot 10^{-5}$ . We will see in the section devoted to the simulation results that this BER level is reached for the recursive BSM CCM well before  $E_b/N_0 = 5.0$  dB. Though in the AWGN channel there seems to be a threshold of around 1.0 dB for the BSM CCM, the BER curves will show that with a practical and finite size interleaver, and as a consequence of not having exactly Gaussian distributed input log probability ratios, this threshold is placed in practice at a higher  $E_b/N_0$ .

In Fig. 4.8 we have depicted the EXIT charts for the CC with  $\nu = 4$  and the BSM CCM in the ISI channel. The  $E_b/N_0$  threshold for low ISI will be almost the same of the case without ISI, thus pointing to a loss of maybe a few tenths of dB. For moderate ISI, the chart exhibits at least a loss in 0.5 dB with respect to the low ISI case, while the two high ISI cases stress the fact that there will be no convergence to the waterfall region, even for  $E_b/N_0 \rightarrow \infty$ . In fact, the starting  $I_2$  value of the BSM CCM curve for high ISI grows a little from  $E_b/N_0 = 1.5$  dB to  $E_b/N_0 = 8.0$  dB, but this is not enough to avoid crossing the convolutional encoder curve at around (0.02, 0.27). Moreover, for a higher  $I_1$  input, the BSM CCM curve with  $E_b/N_0 = 8.0$  dB exhibits a clearly worse behaviour. Therefore, no matter how high the  $E_b/N_0$ , the channel output is so badly distorted that the output mutual information of the inner SISO never grows at the required rate to avoid an early crossing with the CC curve.

In Fig. 4.9 we have plotted the EXIT chart for the convolutional encoder with

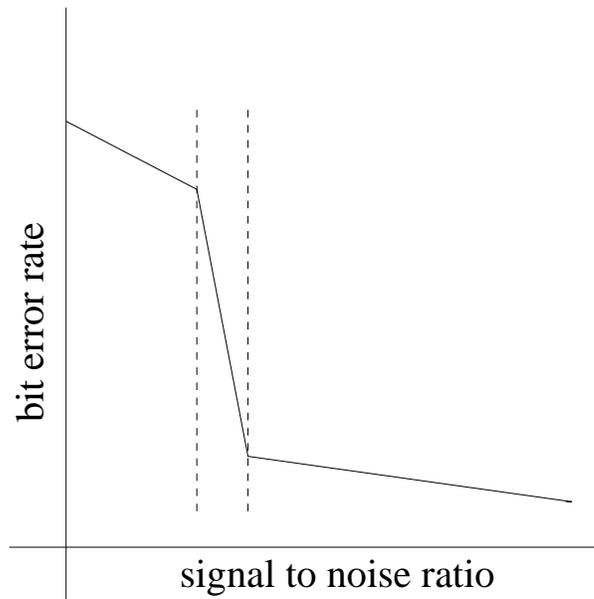
$\nu = 4$  and three different cases for the BSM chaos coded modulation in the channel with Rician fading and perfect CSI. When  $K = 20$  and the channel tends to be a pure AWGN one, we will almost have the same threshold as with the  $K \rightarrow \infty$  case,  $E_b/N_0 \approx 1.0$  dB. As expected, there is a gradual degradation as we tend to the Rayleigh channel, so that the threshold for  $K = 5$  is around 2.0 dB, and, finally, around 3.4 dB for  $K = 0$ . As already stated, we will see that the BER curves show an additional degradation in the  $E_b/N_0$  thresholds due again to the non-ideal interleaver and because the Gaussian distribution assumption is not accurately met. In any case, we can guess that, with CSI, there will be at most a loss of around 2.0 dB in signal to noise ratio, which is a positive result that confirms that the SCCCM scheme can keep the potentially good properties of SCTCM or BICM in fading channels [Biglieri et al., 1998]. In Fig. 4.10 we plot the EXIT charts for the same situation, but without CSI. Now there is an additional loss of around 1.6 dB in the Rayleigh fading non-CSI case with respect to the CSI case, and this loss tends again to vanish as we approach the pure AWGN case ( $K \rightarrow \infty$ ).

These results will prove to be more orientative than exact, because the EXIT charts shown here have been developed under the mentioned assumptions that we have Gaussian distributed log probability ratios and that the interleaver depth is high enough to make the input and output log probability ratios practically uncorrelated from one iterative decoding step to the next one. In a binary turbocode setup with convolutional encoders as constituent encoders, or in usual TTCM, SCTCM or BICM systems, the input/output log probability ratios are approximately Gaussian distributed for the AWGN channel depending on the structure of the decoder and the implementation of the related MAP algorithm within the SISO module [Avudainayagam et al., 2004; Martínez et al., 2006]. We have found out that this also approaches the case for our SCCCM system in the AWGN channel, but it is only a loose approximation in the case of the fading or ISI channels, and the Gaussian distribution assumption only becomes close to the real situation as  $K \rightarrow \infty$  in the fading channel, or when the degree of ISI is low enough. Thus, we will find some mismatch between the threshold points for the waterfall region as calculated in this section and the ones shown in the BER plots depending not only on the size and sort of the interleaver employed, which affects to the independence and uncorrelation assumption, but also on the kind of channel and the degree of distortion introduced. Note also that the uncorrelation assumption is not met in the ISI channel even with a good interleaver, because the channel itself introduces extra correlation among the chaotic samples. Nevertheless, the thresholds obtained from this EXIT charts analysis will give an approximate situation of the waterfall region within the BER plots when there is sufficient interleaving.

#### 4.4. Error probability analysis

The general behaviour of a concatenated system with interleavers, be it serial or parallel, for a purely AWGN channel can be seen in Fig. 4.11. We can distinguish three regions as a function of the signal to noise ratio [Schlegel and Pérez, 2004].

For low  $E_b/N_0$ , the error rate is high because the iterative decoder process is stuck at a fixed point in the EXIT chart far from the (1, 1) point. For mid values of  $E_b/N_0$ , after reaching the threshold for which the (1, 1) point in the EXIT chart is accessible, the error rate falls with a steep slope: this is the already mentioned turbo cliff or waterfall region. When  $E_b/N_0 \rightarrow \infty$ , the error events are dominated by the events occurring with highest probability, which are the ones corresponding to output words with lower squared Euclidean distance, and we are in the error floor region, where the slope changes substantially and the error rate decreases at a much lower rate. It is clear that we need different approaches if we want to bound the bit error probability in each of the regions of interest: the waterfall region and the error floor region.



**Figure 4.11.** *Typical BER curve for a concatenated code with interleavers after several decoding iterations.*

To provide bounds for the bit error probability, we will make use of the developments introduced in [Martínez et al., 2006] for BICM. These developments are only valid if we do not perform iterative decoding, i.e., if we perform only one decoding pass through both SISO modules. Nevertheless, with slight arrangements, they will provide us with useful bounds also for the iterative case. Following [Martínez et al., 2006], we will not focus on the standard channel interface (the one whose input is  $x_n$  and whose output is  $r_n$ , see Fig. 4.1). We will not try either to establish the union bound for ML decoding by calculating the distance spectrum of the serial concatenation of the convolutional code and the chaos coded modulation, which could be a very cumbersome task because the error event probability (EEP) calculated over the chaotic samples in the channel,  $P_e(\mathbf{x} \rightarrow \mathbf{x}'|\mathbf{x})$ , and the bit error probability are not straightforwardly related due to the nonlinear structure of the chaos coded modulation. The evaluation of the joint input-output weight enumerator coefficients,

needed to provide a ML bound as is done in [Benedetto et al., 1998] for serially concatenated systems, would be very difficult, even though our chaos coded modulation admits a description in terms of states and binary transitions between states, and it is possible to calculate the corresponding transfer function as is done with TCM [Lin and Costello, Jr., 2004] or BICM [Narayanan and Stüber, 1999]. To avoid these tasks, we will make use instead of the fact that the interface between  $c_n$  and  $\Lambda(c_n; \mathbf{I})$  (see Fig. 4.1) constitutes a binary input-output symmetric (BIOS) channel if there is sufficient bit interleaving at the convolutional encoder output, so that we can assume that the mapping between  $\mathbf{c}$  and  $\mathbf{x}$  is independent and the performance will depend on the binary error event  $\mathbf{c} \otimes \mathbf{c}'$  instead of on the individual values of  $\mathbf{c}$  and  $\mathbf{c}'$ . In our case, due to the characteristics of the chaos-based encoder, we do not need to symmetrize the channel by randomly manipulating the mapping, as is done with BICM in [Caire et al., 1998]. Therefore, we can only focus on the convolutional decoder and on the EEP over the outputs of the deinterleaver at the receiver side [Caire et al., 1998]. Fortunately, since this BIOS channel is linear, we can assume, without loss of generality, that the all-zero codeword  $\mathbf{c} = (0, \dots, 0)$  has been sent. For clarity sake, we review briefly the method of [Caire et al., 1998], where the bit error probability is closely upper bounded by the union bound

$$P_b \leq \sum_{d=d_{\text{free}}}^{\infty} B_d P_e(d|\Theta), \quad (4.20)$$

where  $B_d$  is the bit enumerator of the CC for error paths differing in a Hamming weight  $d$ ,  $P_e(d|\Theta)$  is the error event probability for an output error event with Hamming weight  $d$ , and  $d_{\text{free}}$  is the free distance of the convolutional code. We have introduced the dummy variable vector  $\Theta$  which includes the parameters that define the distortion introduced by the channel, just to recall that the EEP will depend on the state of the channel. This EEP will be a function of the random variable (or *equivocation*)

$$\Lambda = \Lambda(c; \mathbf{I}) = \log \frac{P(c = 1|\Theta)}{P(c = 0|\Theta)}, \quad (4.21)$$

where we have dropped the subindex because the sequence  $\Lambda(c_n; \mathbf{I})$  will be approximately *iid* under the assumption of sufficient interleaving. As shown in [Martínez et al., 2006], the EEP for MAP decoding (which is the same of ML decoding in the non-iterative case, since the *a priori* probabilities are the same) will be

$$P_e(d|\Theta) = P\left(\sum_{i=1}^d \Lambda_i > 0\right), \quad (4.22)$$

where the  $\Lambda_i$  are  $d$  independent realizations of the random variable  $\Lambda$  of Eq. (4.21). That is to say, Eq. (4.22) is the probability of decoding a word with a Hamming weight  $d$  instead of the all-zero codeword. The tail probability of the sum of random variables of Eq. (4.22) can be efficiently bounded by using the cumulant transform

[Martínez et al., 2006]

$$\kappa(s) = \log E [e^{s\Lambda}] = \log \left[ \int_{-\infty}^{\infty} e^{s\Lambda} f_{\Lambda}(\Lambda) d\Lambda \right], \quad (4.23)$$

where  $s \in \mathbb{R}$  and  $f_{\Lambda}(\Lambda)$  is the pdf of  $\Lambda$ .  $\kappa(s)$  is a convex function which reaches its maximum at  $\hat{s} = 1/2$ , and, following the Gaussian approximation of [Guillén i Fábregas et al., 2004], the bound for the EEP can be written finally as

$$P_e(d|\Theta) \approx \frac{1}{2} \operatorname{erfc} \left( \sqrt{-d\kappa(\hat{s})} \right) \leq \frac{1}{2} e^{d\kappa(\hat{s})}, \quad (4.24)$$

where we have made use of the inequality  $\operatorname{erfc}(\sqrt{x}) \leq e^{-x}$  [Proakis, 2001]. On the other side, knowing that the bit weight enumerator function of the convolutional code can be expressed as [Lin and Costello, Jr., 2004]

$$B(X) = \sum_{d=d_{\text{free}}}^{\infty} B_d X^d = \frac{1}{k} \frac{\partial A(X, W)}{\partial W} \Big|_{W=1}, \quad (4.25)$$

where  $A(X, W)$  is the input-output weight enumerator function and  $k = 1$  is the number of information bits per unit time,  $P_b$  can be upper bounded by

$$P_b \leq \frac{1}{2} B(X) \Big|_{X=e^{\kappa(\hat{s})}}. \quad (4.26)$$

A tighter bound can be given by using the inequality  $\operatorname{erfc}(\sqrt{x+y}) \leq \operatorname{erfc}(\sqrt{x}) e^{-y}$ ,  $x > 0$ ,  $y \geq 0$  [Lin and Costello, Jr., 2004], so that, finally

$$P_b \leq \frac{1}{2} \operatorname{erfc} \left( \sqrt{-d_{\text{free}}\kappa(\hat{s})} \right) e^{-d_{\text{free}}\kappa(\hat{s})} B(X) \Big|_{X=e^{\kappa(\hat{s})}}. \quad (4.27)$$

To calculate  $\kappa(\hat{s})$  as the logarithm of the expectation of Eq. (4.23), we can resort to simulate the system and store the resulting samples to get the histogram of  $\Lambda$ , since the theoretical calculation of the pdf  $f_{\Lambda}(\Lambda)$  with a CCM system will be difficult even in the AWGN channel, and it could depend in a complex way on the mapping between the bits and the chaos coded modulated symbols. This also depends on the particular implementation of the BCJR algorithm within the SISO module [Yeh et al., 2006]. In any case, the simulations needed to get the histogram of  $\Lambda$  with enough accuracy are less time consuming than the simulations needed to give a Monte Carlo estimation of the BER.

With respect to the iterative case, we can make use of the same principles and take the histogram over  $\Lambda$  after several iterations. The channel thus described will also be BIOS, and the decoding will now be MAP decoding, since we are making use of the *a priori* information from the other SISO module. Following the definition of  $\Lambda$ , the EEP for MAP decoding, assuming that the all-zero codeword has been sent, is again given by the equations seen, and  $\kappa(\hat{s})$  can be as well calculated by

simulation and using the corresponding histogram. Though this bound is based on probability density estimation as is done in the calculation of EXIT charts, it will be more accurate than a possible bound based on these charts [ten Brink, 2001], since it takes into account the actual interleaver structure and depth. Note that these developments are valid if  $\Lambda$  is Gaussian distributed at least in the tail [Martínez et al., 2006], and if we provide sufficient interleaving to keep  $\mathbf{c}$  and  $\mathbf{x}$  mutually independent. These are almost the same assumptions we made to calculate the EXIT charts. As was mentioned in this case, these assumptions will determine a mismatch between the theoretical results and the simulation results. This difference will be specially remarkable in the dispersive channels, or when the interleaver length is small.

In the examples of the next section, we will make use of the simple  $\nu = 4$  CC introduced in the preceding section, whose input-output weight enumerator and bit weight enumerator functions can be calculated using standard methods [Lin and Costello, Jr., 2004], as

$$\begin{aligned} A(X, W) &= \frac{WX^5 - W^2X^6 + W^2X^8}{1 - 2WX - WX^3} = \\ &= WX^5 + W^2X^6 + 2W^3X^7 + (4W^4 + 2W^2)X^8 + \dots, \end{aligned} \quad (4.28)$$

$$\begin{aligned} kB(X) &= \frac{X^5 - 2X^6 + 2X^7 + 2X^8 - X^9 - X^{11}}{1 - 4X + 4X^2 - 2X^3 + 4X^4 + X^6} = \\ &= X^5 + 2X^6 + 6X^7 + 20X^8 + 56X^9 \dots \end{aligned} \quad (4.29)$$

Note also that the bound of Eq. (4.27) for the non-iterative decoding is a union bound and will converge only for high  $E_b/N_0$  [Proakis, 2001]. In the case of iterative decoding, the bound will prove useful for the waterfall BER region, but not in the error floor region, where we cannot resort to the computation of the evolution of the log probability ratios due to the structural constraints of the SISO modules (where some clipping and normalization in the log probabilities values must always be performed in order to provide stability and avoid overflows [Benedetto et al., 1997]). In this case, we would need a more involved analysis based upon a study of the ML decoding of the joint SCCCM system without the simplification of the BIOS channel. Though still not addressed due to its inherent complexity, this task is required in order to complete the error analysis of the SCCCM, since in the error floor region the BER cannot be reached easily by simulation<sup>4</sup>.

It has been shown for non-iterative decoding of BICM [Martínez et al., 2006] that the system will behave in the AWGN channel with  $E_b/N_0 \rightarrow \infty$  like the CC alone in a channel with  $\kappa(\hat{s}) \rightarrow -\frac{d_{\min}^2}{4} SNR = -\frac{2Rd_{\min}^2}{4} \frac{E_b}{N_0}$ , where  $d_{\min}^2$  is the minimum squared Euclidean distance of the modulation. This will also hold in our case for the non-iterative case. This is a direct consequence of the metric definition,  $(r_n - x_n)^2$ , which will make that, for high signal to noise ratios, only the path with the minimum

<sup>4</sup>This is specially true in serial concatenated schemes like this one [Garello et al., 2001].

metrics will survive in the decoding process (see Eq. (3.8)), and so, following a development similar to the one presented in [Martínez et al., 2006], we would have

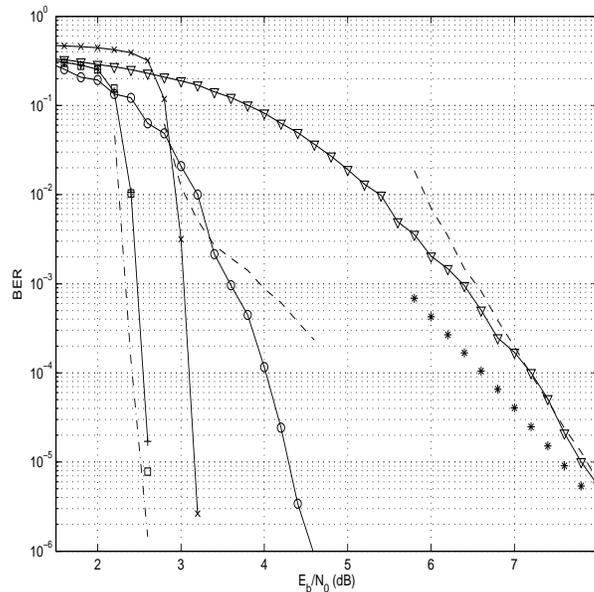
$$\kappa(\hat{s}) \rightarrow -\frac{\min_{\mathbf{x}, \mathbf{x}'} \sum_n (x_n - x'_n)^2}{4} SNR = -\frac{2Rd_{\min}^2}{4} \frac{E_b}{N_0} \quad (4.30)$$

In our case, we take  $d_{\min}^2 = 4/3$ , which is the minimum squared Euclidean distance between all the possible chaotic sequences of the BSM encoder with  $Q \rightarrow \infty$ , since it does not differ much from the minimum squared Euclidean distance for the  $Q = 5$  value of our examples [Kozic, 2006]. In this situation, the bound of Eq. (4.27) for  $E_b/N_0 \rightarrow \infty$  and non-iterative decoding can be calculated using

$$P_e(d|\Theta) = e^{-\frac{2Rd_{\min}^2}{4} \frac{E_b}{N_0}}. \quad (4.31)$$

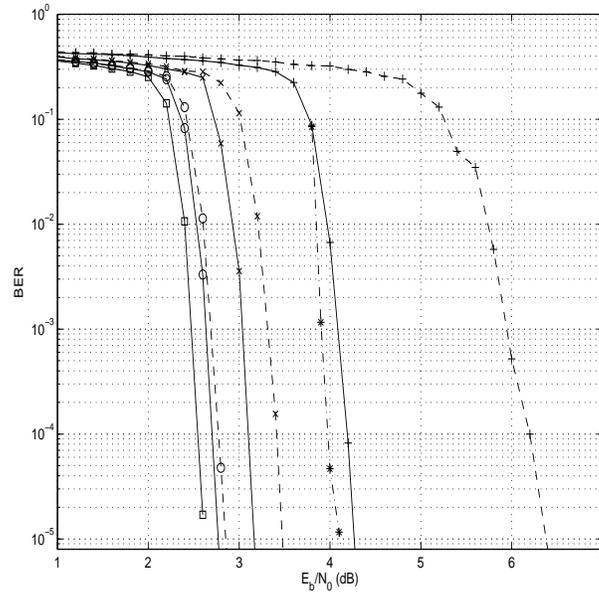
We will see that this bound tends as a lower bound to the BER without iterative decoding for high  $E_b/N_0$ .

## 4.5. Simulation results

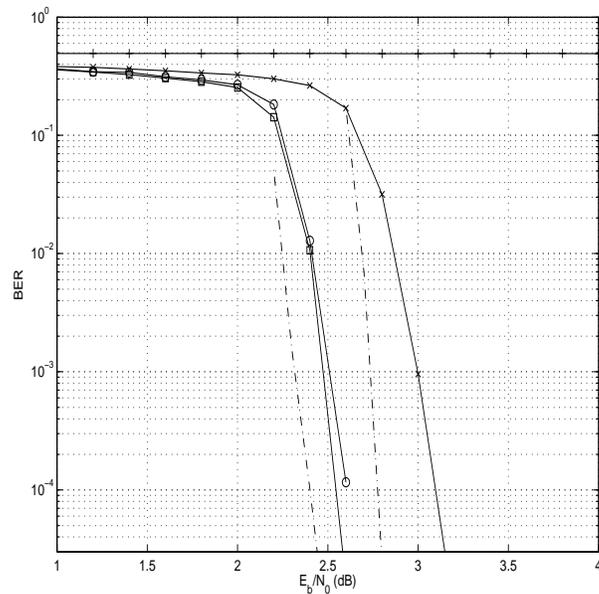


**Figure 4.12.** BER for several cases of SCCCM in the AWGN channel. '+' :  $N = 10000$ ,  $S = 23$ ,  $Q = 5$ ,  $\nu = 4$ . 'x' : same parameters, but with convolutional code of  $\nu = 7$ . 'o' :  $N = 400$ ,  $S = 11$ ,  $Q = 5$ ,  $\nu = 4$ . 'v' : same parameters as '+', but without iterative decoding. Dashed lines: from left to right, bounds for the cases with  $\nu = 4$  and  $Q = 5$ . '\*': bound with  $d_{\min}^2$  for the non-iterative case. '□': same parameters as '+', but with  $Q = 4$ .

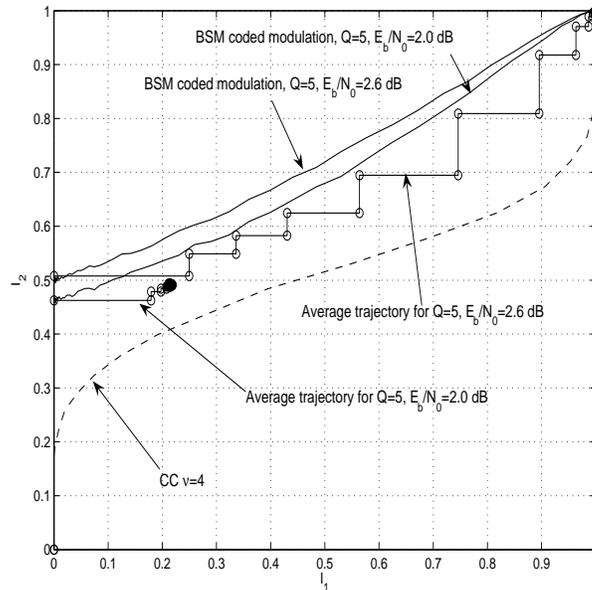
In Fig. 4.12, we show the simulation results for the SCCCM system in the AWGN channel with different sets of parameters. For the cases with iterative decoding, we



**Figure 4.13.** BER and bounds for several cases of SCCCM in the fading channel, with and without CSI. Continuous line: with CSI. Dashed line: without CSI. In all cases,  $N = 10000$ ,  $S = 23$ ,  $Q = 5$ ,  $\nu = 4$ . From left to right,  $K = 20$ ,  $K = 5$ ,  $K = 0$ . BER in the AWGN channel and same parameters is depicted with '□' for comparison. '\*' with dash-dotted line: bound for  $K = 0$  and CSI.



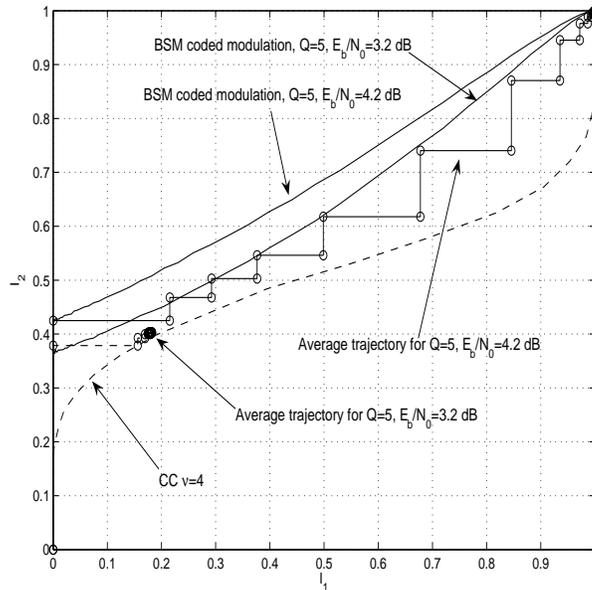
**Figure 4.14.** BER and bounds for several cases of SCCCM in the ISI channel. In all cases,  $N = 10000$ ,  $S = 23$ ,  $Q = 5$ ,  $\nu = 4$ . 'o': low ISI. 'x': moderate ISI. '+' : high ISI. BER in the AWGN channel and same parameters is depicted with '□' for comparison. Dash-dotted lines: bounds for low ISI (left) and moderate ISI (right).



**Figure 4.15.** EXIT chart and average trajectory of the mutual information interchange in the AWGN channel, for the case with  $Q = 5$ ,  $N = 10000$  and  $S = 23$ .

performed 20 decoding iterations both to get the BER and to get the described bound that uses the histogram over the log probability ratios. First of all, as predicted through the EXIT charts, the case with  $\nu = 7$  reaches the waterfall region with an  $E_b/N_0$  threshold higher than the case with  $\nu = 4$ , for the same size and kind of interleaver and for the same quantization level in the BSM modulator. We also show the BER for this same same sets of parameters with  $\nu = 4$ , but without decoding iteratively (i.e., only one decoding pass is performed). We see that the decoding gain is dramatically linked to the iterative decoding of the SCCCM, just as is was shown for other serially concatenated systems as SCTCM [Divsalar and Pollara, 1997] or BICM [Li and Ritchey, 1997].

The results make evident the importance of a good interleaver design, since, though the performance of the case with  $N = 400$  and  $S = 11$  without iterative decoding (not shown) is the same as the performance for the case with  $N = 10000$  and  $S = 23$  without iterative decoding, when decoding iteratively there is a remarkable difference in the behaviour of the BER: with  $N = 400$ , we have a much shallower BER slope. This means that a high interleaver depth is necessary to get good results, because a better interleaver design can provide a higher degree of uncorrelation between coded bits and chaotic samples and, as a consequence, more accurate *a priori* information at the beginning of each decoding pass. We have also depicted in Fig. 4.12 the bounds for the cases with  $\nu = 4$ . As expected, when  $E_b/N_0 \rightarrow \infty$ , the bound of Eq. (4.27) for the non-iterative case converges as a union upper bound to the final performance, and the bound that uses the EEP of Eq. (4.31) converges as a lower bound. On the contrary, the bound for the  $N = 400$  case and iterative decoding is not very helpful. This is due to the fact that the BIOS assumption

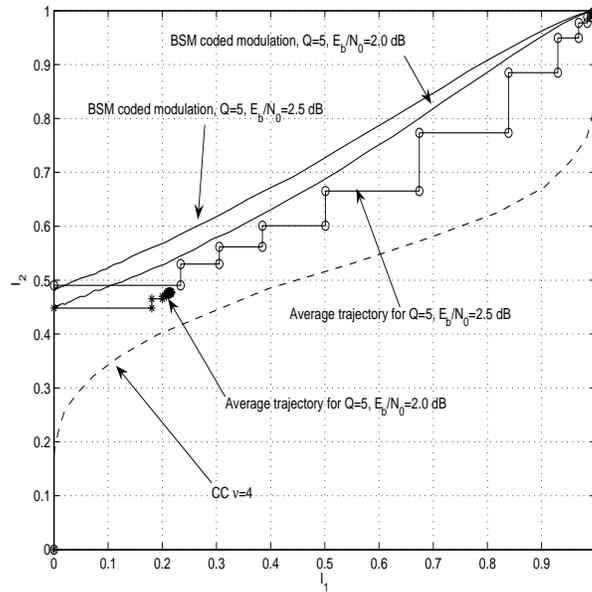


**Figure 4.16.** EXIT chart and average trajectory of the mutual information interchange in the fading channel, for the case with  $Q = 5$ ,  $N = 10000$ ,  $S = 23$ ,  $K = 0$  and perfect CSI at the decoder.

only holds for sufficient interleaving, and a low value of  $N$  does not guarantee the needed symmetry. On the other hand, the bound for the  $N = 10000$  case and iterative decoding, though not extremely tight, at least gives accurate information about the slope in the waterfall region. Note also that what was said about the  $E_b/N_0$  thresholds for the waterfall region through the EXIT charts is true: in the best case ( $N = 10000$ ), there is a difference between the expected value and the actual value of around 1.0 dB, since the EXIT chart predicted a threshold of about 1.0 dB for the convolutional code of  $\nu = 4$  and of about 2.0 dB for the convolutional code of  $\nu = 7$ . We can as well verify that the influence of  $Q$  is small, provided that it has not a very low value (for example, for  $Q = 1$  we would be in the BPSK case and no chaotic dynamics would be involved): the BER for  $Q = 5$  and  $Q = 4$  is virtually the same for the same set of parameters (see Fig. 4.12). The best results we have for the AWGN case are similar to the BER results of BICM with iterative decoding presented in [Narayanan and Stüber, 1999], which were drawn for a 4-state rate  $R = 1/2$  convolutional code and coded modulations of 2 and 4 states. Though our system is more complex<sup>5</sup>, it will show real advantages in dispersive channels.

In Fig. 4.13 we have depicted the results for  $N = 10000$ ,  $S = 23$ ,  $\nu = 4$  in the fading channel with different degrees of fading, both when there is CSI at the decoder and when there is no CSI. All the results have been taken after 20

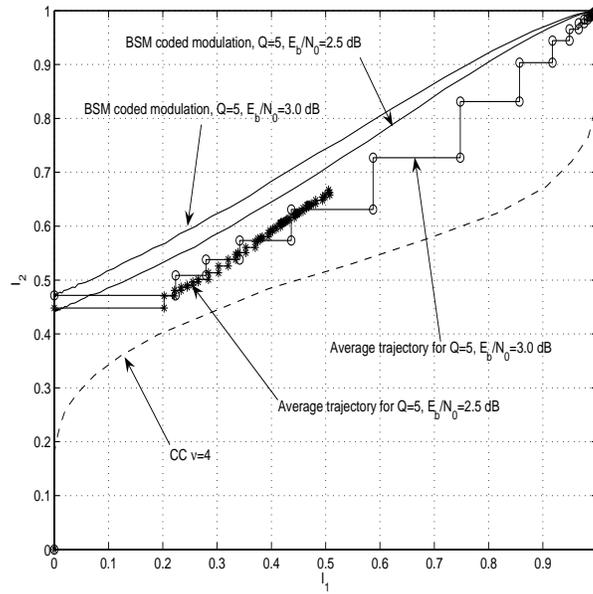
<sup>5</sup>We have an 8-state convolutional code and a CCM with  $2^5 = 32$  states ( $Q = 5$  in most of the examples). The SISO module for CCM is also more complex than the SISO module used for the inner modulation in iterative decoded BICM [Li and Ritchey, 1997], because it requires the forward-backward MAP sequence decoding algorithm [Bahl et al., 1974].



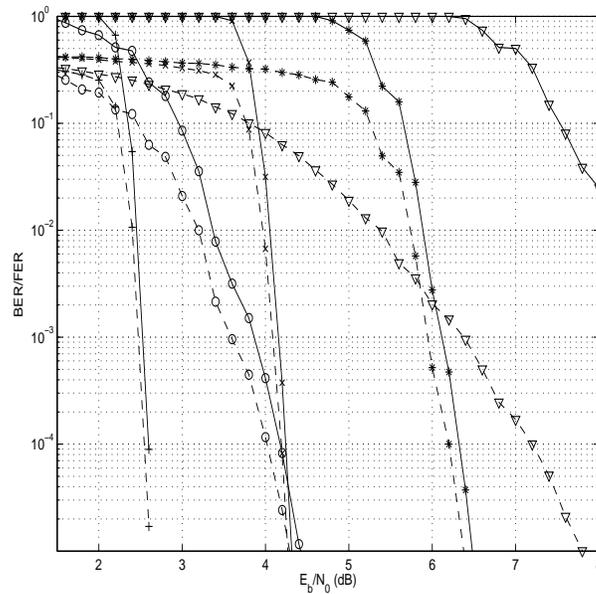
**Figure 4.17.** EXIT chart and average trajectory of the mutual information interchange in the ISI channel with low ISI, for the case with  $Q = 5$ ,  $N = 10000$ ,  $S = 23$ .

iterations. Again, there is a mismatch between the  $E_b/N_0$  threshold given by the simulations with respect to the information given by the EXIT charts. While for the Rayleigh fading case the expected thresholds were around 3.0 dB with CSI and 4.0 dB without CSI, the BER results show the thresholds at about 4.0 dB and 5.5 dB, respectively. Since the Rayleigh fading is the worst case of Rician fading, it is not surprising to see that it yields the worst results and the largest difference between the CSI and non-CSI cases. On the contrary, the results tend to the BER in the AWGN channel as  $K$  grows, both with CSI and without CSI. Even the worst case (Rayleigh without CSI) still keeps the good properties in point of BER slope once reached the waterfall region, though with a loss of about 3.0 dB with respect to the non-dispersive AWGN case. For the Rayleigh fading and perfect CSI, the loss is around 1.5 dB. These results are consistent with what was shown in [Yuan et al., 2002] for serial concatenation of convolutional codes, and improves the results of [Caire et al., 1998; Li and Ritchey, 1997], where the differences between the  $E_b/N_0$  needed to reach a given BER for the AWGN case and for the Rayleigh fading with CSI case were higher in BICM systems whose complexity was not much lower than the complexity of our SCCCM examples. We have also included the bound with iterative decoding for the Rayleigh channel with CSI, and, as seen before for the AWGN channel, when the interleaver depth is enough, the results can approximate accurately the slope in the waterfall region. Note that the bound with fading is not as tight as the bound with AWGN alone. This is due to the fact that  $\Lambda$  in the fading channel is no longer well matched with a Gaussian density.

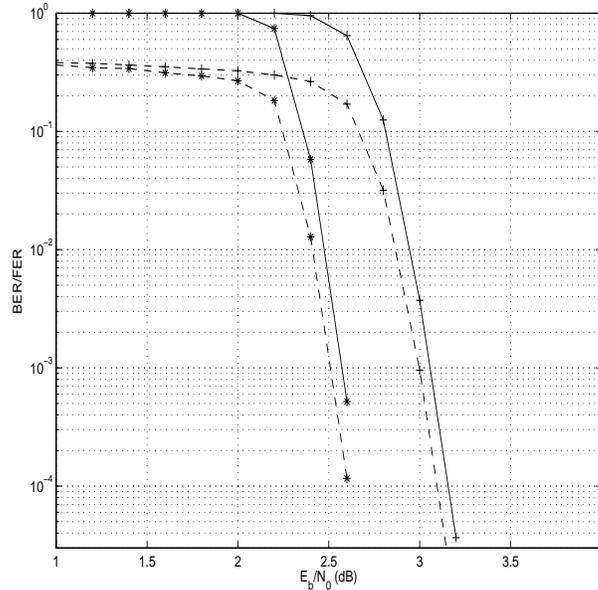
In Fig 4.14 we show the results for  $N = 10000$ ,  $S = 23$ ,  $\nu = 4$  after 20 iterations



**Figure 4.18.** EXIT chart and average trajectory of the mutual information interchange in the ISI channel with moderate ISI, for the case with  $Q = 5$ ,  $N = 10000$ ,  $S = 23$ .



**Figure 4.19.** FER and BER for several cases of SCCCM with the convolutional code with  $\nu = 4$  and BSM with  $Q = 5$ . Continuous line: FER. Dashed line: BER. '+' :  $N = 10000$ ,  $S = 23$ , AWGN channel. 'o' :  $N = 400$ ,  $S = 11$ , AWGN channel. '∇' : same as '+', but without iterative decoding. 'x' : same as '+', but with fading,  $K = 0$  and CSI. '\*' : same as 'x', but without CSI.



**Figure 4.20.** FER and BER in the ISI channel for two cases of SCCCM with the convolutional code with  $\nu = 4$  and BSM with  $Q = 5$ ,  $N = 10000$ ,  $S = 23$ . Continuous line: FER. Dashed line: BER. '\*' : low ISI. '+' : moderate ISI.

in the case of the ISI channel. The channel with low ISI exhibits a very low loss with respect to the non-dispersive case, while in the case of moderate ISI we lose about 0.5 dB. These values agree with what was seen through the EXIT charts, though the exact values of the thresholds are again 1.5 dB above the results of these charts. In the case of high ISI, the BER does not fall never below  $4 \cdot 10^{-1}$ . This matches very well with what was said during the convergence analysis. We have also depicted the corresponding bounds for low and moderate ISI. For low ISI, the bound is as tight as it was for the pure AWGN channel, and it gives reason again of the slope in the waterfall region. For moderate ISI, as the received samples have a larger degree of correlation due to the lower cutoff frequency of the filter, the bound is not as tight and its slope differs remarkably from the results. Recall that this was also the case for the fading channel. Apart from the examples mentioned, it is easy to verify that the BER results of SCCCM in the channels under consideration are at least comparable to the results of other related serially concatenated systems [Divsalar and Pollara, 1997; Narayanan and Stüber, 1999; Li and Mow, 1999; Tullberg and Siegel, 2001; Altunbas and Narayanan, 2001; Yuan et al., 2002; Pfister and Siegel, 2003; Tullberg and Siegel, 2005a; Tullberg and Siegel, 2005b; Schlegel and Pérez, 2004; Howard and Schlegel, 2006].

To shed further light on the behaviour of the iterative decoding process, we have depicted in Figs. 4.15, 4.16, 4.17 and 4.18 several EXIT charts where we can compare the predicted behaviour with the real evolution of the mutual information. This evolution was calculated by averaging over the log probability ratios after each iteration during the simulations performed to plot the BER results. In this way,

we get the so called average trajectory of the mutual information [ten Brink, 2001]. As hinted when comparing the  $E_b/N_0$  thresholds of the waterfall region, there is a mismatch between the theoretical curves and the real trajectory. We can see in all the graphs that only the first step corresponds with the expected value, since this is the case when we have no feedback from the other decoder. In the rest of steps, there is always a lower value of the mutual information with respect to the one calculated with the assumption of uncorrelated Gaussian distributed input log probability ratios, and so the iterative decoding process does not achieve convergence at the first  $E_b/N_0$  values where the calculated transfer curves do not intersect before the (1,1) point. Apart from the Gaussian density assumption, which only holds approximately, there are other possible sources of mismatch, like an inappropriate definition of mutual information [Cover and Thomas, 2006], or like an interleaver design not providing enough uncorrelation between channel samples and encoded bits. This could be overcome to some extent by increasing the interleaver size  $N$  at the expense of a higher complexity and a longer processing delay, but in a channel with ISI the correlation will subsist unless some equalization method is implemented. Nevertheless, the losses in the  $E_b/N_0$  thresholds are almost the same for the AWGN channel and for the dispersive channels considered, and this points towards the SISO and interleaver structures as reasons for such mismatch, rather than towards the differences in the channel distortion.

Finally, we have depicted in Figs. 4.19 and 4.20 the frame error rate (FER) and the BER for several of the cases already commented. When there is no iterative decoding, the FER is very high, and starts decreasing when the main error events are of the  $d_{\text{free}}$  kind<sup>6</sup>, so that each frame on error contains only 1 error bit and then  $\text{BER} \approx (1/D) \text{FER}$ , with  $D = R \cdot N - \nu = 4996$ . On the other hand, when decoding iteratively, this situation is highly improved, since the FER for the interleaver with  $N = 10000$  in the AWGN channel follows the behaviour of the BER and exhibits the same steep slope in the waterfall region (see Fig. 4.19). This means that there are less frames on error, but, as a contrast, each frame on error contains around 1000 erroneous bits in the waterfall region, where thus  $\text{BER} \approx (1/5) \text{FER}$ . For the case with  $N = 400$ , the situation is the same, but now each frame on error contains around 60 erroneous bits, and, as  $D = 196$ ,  $\text{BER} \approx (1/3) \text{FER}$ . So the gain in BER and FER is made at the expense of having large bursts of errors, instead of having the dominant low Hamming weight error events of the non-iterative case. The situation for the fading channel is the same (Fig. 4.19), as shown through the worst case, i.e., Rayleigh fading. In the waterfall region, the FER falls down with the same slope as the BER and we have still that each frame on error contains around 1000 erroneous bits, so that, again,  $\text{BER} \approx (1/5) \text{FER}$ , both for the CSI and non-CSI cases. The same holds for the ISI channel with low and moderate ISI (see Fig. 4.20). Therefore, the iterative decoder behaves, at least in the waterfall region, in the same way with independence of the kind of channel. The channel only seems to affect the  $E_b/N_0$  threshold for this region, and this shows that SCCCM keeps

<sup>6</sup>This  $d_{\text{free}}$  is determined in the case of our non-recursive CC by binary input error events  $\mathbf{e}$  with Hamming weight  $w(\mathbf{e}) = 1$ .

the good properties of other serially concatenated systems in AWGN and dispersive channels, uncoupling the channel state and the error correcting code capabilities once we are above some threshold of channel distortion. It would be interesting to study what happens on the error floor region, but, as we cannot reach it easily through simulation, this will require developing much more complex performance bounds based upon ML analysis.

## 4.6. Concluding remarks

In this chapter we have pushed further the analogy between traditional coded modulated systems and CCM, according with the principle that the applicability of chaos-based communications systems would arise from a close interaction between well established digital communications theory and Chaos Theory. In this case, we have extended the concept of serially concatenated trellis coded modulations by replacing the usual bandwidth-efficient phase or phase-amplitude inner TCM with a CCM system. Thanks to the principles shown in the previous chapters respecting the chaotic encoder and the MAP SISO decoder, this new development has been carried out pretty straightforwardly, with only the slight arrangement of allowing feedback in the CCM encoder as required for any inner encoder in a concatenated system [Narayanan and Stüber, 1999].

The analogy with SCTCM and BICM has allowed us to adapt other standard tools, such as the EXIT charts in order to evaluate the iterative decoding process, or such as the concept of BIOS channel in order to give a bound for the bit error probability. The results have shown that, with a moderate mismatch due to the fact that CCM does not meet exactly the assumptions taken from the TCM case, the predicted behaviour is close enough to the simulations. Therefore, we are provided with tools well suited to the design and evaluation of CCM's other than the BSM one and which may do still better. We have also shown in our analysis that the chaotic properties of the modulator are the determining factor for the final behaviour against the channel distortion with independence of the quantization level of the CCM, at least down to a minimum value of the same. This establishes a new bridge between digital communications theory and Chaos Theory, since the systems can be implemented with limited complexity, while their properties and performance can be potentially studied for the unquantized case.

Moreover, the BER results obtained point out that SCCCM keeps the good behaviour of SCTCM and BICM in dispersive environments even with the simplest BSM CCM, and that the robustness against low or moderate ISI and fading without CSI are greatly improved with respect to the results of BSM CCM alone as shown in Chapter 3. Besides, we have verified that other facts known from SCTCM and BICM seem to be valid for SCCCM, such as the extremely low BER error floor and the better behaviour of low constraint length CC's working as outer encoders. Nevertheless, the resulting SCCCM system is more complex to decode than BICM because the SISO module for the chaos coded modulation requires more computations due to the forward-backward MAP algorithm. On the other hand, SCCCM

has the advantage that the chaos coded modulations are easy to generate and produce noise-like broadband signals suitable, among others, for multiuser environments [Rovatti et al., 2001] where phase or phase-amplitude TCM based systems could be severely distorted.

## Chapter 5

# Parallel Concatenated Chaos Coded Modulations

### 5.1. Introduction

In Section 4.1 we considered the possibility of enhancing the CCM based systems performance by using concatenated systems with interleavers, so that the dimensionality and redundancy of the systems were increased. We have explored accordingly in Chapter 4 the extension of the paradigm of serial concatenation to systems including chaos-based encoders and we have shown that the CCM encoder can work in any scheme admitting a coded modulation with binary inputs. Other way of managing the dimensionality of a system with CCM's is the possibility of parallel concatenation. If SCTCM was the traditional counterpart of the system analyzed in Chapter 4, the parallel concatenation of two or more CCM's linked by means of a corresponding number of bit interleavers is the natural extension to chaotic modulations of the turbo TCM (TTCM) systems [Robertson and Wörz, 1995; Divsalar and Pollara, 1997; Robertson and Wörz, 1998; Firmanto et al., 2002; Robertson and Wörz, 2002; Schlegel and Pérez, 2004].

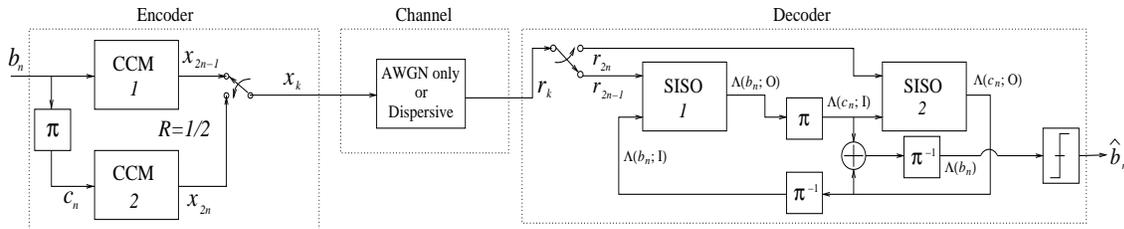
Parallel concatenated coding of any kind relies mainly on the results of the leading work on parallel concatenated binary channel codes [Berrou et al., 1993], also known as turbocodes, where the iterative decoding of the resulting code was shown to achieve outstanding performance very close to Shannon's limit. Turbo TCM arised as a normal development in environments where binary modulation is not desirable, and so we can think of performing another step forward and verify if parallel concatenated CCM's can be as good as their non-chaotic counterparts. In fact, since the TTCM systems combining the bandwidth efficient TCM systems and the philosophy of turbocoding [Robertson and Wörz, 1998] have shown good results in AWGN and radio channels, we expect that new chaos-based systems built under the same principles could lead at least to comparable results.

As a consequence, we address in this chapter the design of encoding and decoding systems including two CCM's parallel concatenated by means of a bit interleaver, and show again that standard tools from digital communications theory can help to look into the properties of such systems and evaluate their performance. According to all this, the chapter is structured as follows. Section 5.2 is devoted to the design

of the concatenated encoder and to the particularities of the individual chaotic encoders, together with a revision of the channel models and of the iterative decoder. Section 5.3 introduces the convergence analysis of the decoding algorithm by means of the already exploited extrinsic information transfer charts device. Section 5.4 looks into the squared Euclidean distance properties of the resulting concatenated coded modulation and proposes a bound for the bit error probability. Section 5.5 is devoted to the simulation results and to the validation of the predictions of Sections 5.3 and 5.4. Finally, in Section 5.6 we give our concluding remarks.

## 5.2. System model

As mentioned, we will study here the possibility to get good chaos coded modulated systems by performing parallel concatenation of individual CCM's linked by means of a bit interleaver. According to this, we will call this system parallel concatenated chaos coded modulations (PCCCM). The scheme of the complete system can be seen in Fig. 5.1, and we will describe it with some detail in the following subsections.



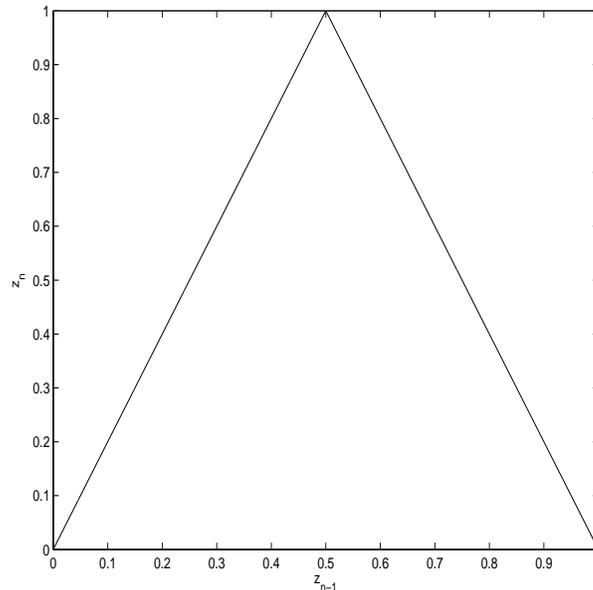
**Figure 5.1.** Block diagram of the parallel concatenated encoder, the channel and the iterative decoder.

### 5.2.1. Parallel concatenated encoder

The concatenated encoder is shown on the left side of Fig. 5.1. It reproduces the well known structure of the so called turbo codes [Berrou et al., 1993], which consist originally on the parallel concatenation of two channel encoders fed with the same input binary sequence  $b_n$ , with the difference that the sequence is scrambled before the second encoder by means of an interleaver  $\Pi$ . The encoding with CCM's instead of binary channel codes can also be seen as an instance of turbo trellis coded modulation (TTCM) [Schlegel and Pérez, 2004; Robertson and Wörz, 1998]. As in the case of SCCCM, the interleaver operation is performed over bit blocks of size  $N$ . This means that now we will deal with an input alphabet of binary words of size  $N$ ,  $\mathbf{b} = (b_1, \dots, b_N)$ . The output words will have size  $2N$ ,  $\mathbf{x} = (x_1, \dots, x_{2N})$ , with  $x_k$ ,  $k = 1, \dots, 2N$ , taking values alternatively from each of the encoders: when  $k$  is odd ( $k = 2n - 1$ ,  $n = 1, \dots, N$ ), the sample corresponds to the first encoder; when  $k$  is even ( $k = 2n$ ,  $n = 1, \dots, N$ ), the sample corresponds to the second encoder (see Fig. 5.1).

As stated, the main difference with respect to a common turboencoder is the presence of the chaos coded modulators on each branch, denoted as CCM 1 and CCM 2 in Fig. 5.1. From Chapter 3 and so far, we have limited ourselves to the study of chaos coded modulations based on a chaotic map driven by small perturbations with a setup which allowed the representation of the encoding process in terms of a related trellis encoder. Moreover, the specific examples have been given for the Bernoulli shift map, which has served us to illustrate with the simplest example the potential coding gain of CCM when used alone in dispersive environments, or when concatenated with a binary channel encoder and a bit interleaver.

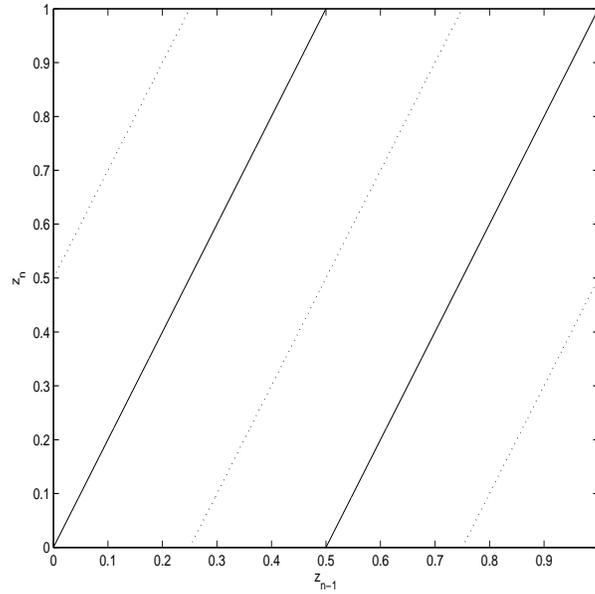
Nevertheless, as we will verify throughout this chapter, the BSM does not offer good properties for parallel concatenation, and so we propose and evaluate here other CCM's as well. We will use chaos-based encoders of two types. The first sort is the encoder already seen and consisting on a chaotic map controlled by small perturbations<sup>1</sup> [Kozic et al., 2003a; Kozic, 2006], and the second is a chaos shift keying (CSK) encoder [Lau and Tse, 2003], but combined with a small perturbations setup. These encoders joining both switching map control and small perturbations control provide better behaviour than pure CSK encoders or pure small perturbations controlled one-dimensional maps [Kozic et al., 2003a]. That is the reason to propose them here, together with the fact that such encoders provide a more complex structure than the BSM based CCM and therefore potentially better results in a concatenated scheme. The small perturbations control or the switching between maps are determined by the binary input for each encoder,  $b_n$  and  $c_n$  as shown in Fig. 5.1.



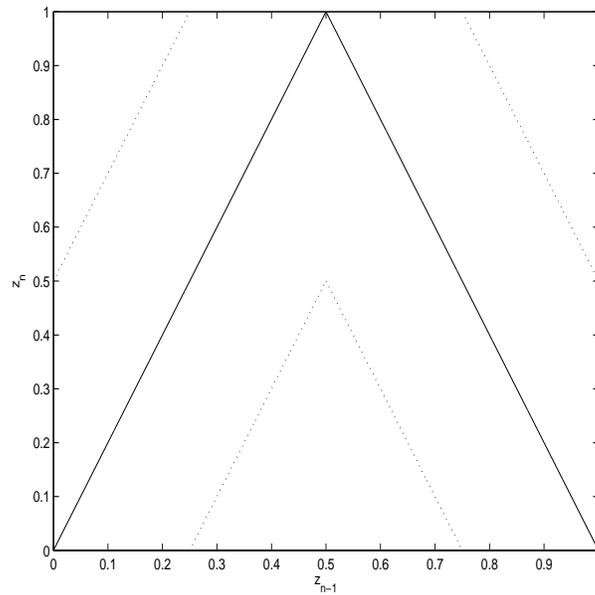
**Figure 5.2.** *The figure shows the tent map.*

---

<sup>1</sup>See Section 3.2.1 in Chapter 3.



**Figure 5.3.** The figure shows the multi-Bernoulli shift map. The continuous line corresponds to  $b_n = 0$ ; the dotted line, to  $b_n = 1$ .



**Figure 5.4.** The figure shows the multi-tent map. The continuous line corresponds to  $b_n = 0$ ; the dotted line, to  $b_n = 1$ .

Both kinds of chaotic encoding systems can be described, for  $n = 1, \dots, N$ , by a general recursion in the form:

$$z_n = f(z_{n-1}, b_n) + g(b_n, z_{n-1}) \cdot 2^{-Q}, \quad (5.1)$$

$$x_n = 2z_n - 1, \quad (5.2)$$

where  $f(\cdot, 0) = f_0(\cdot)$  and  $f(\cdot, 1) = f_1(\cdot)$  are chaotic maps that leave the interval  $[0, 1]$  invariant. In addition, and according with which was said in Chapter 3, they are piecewise linear maps with slope  $\pm 2$  wherever it is defined. Recall that the natural number  $Q$  indicates the number of bits to represent  $x_n$ , and  $g(b_n, z_{n-1}) \in \{0, 1\}$  is the small perturbations term. As denoted,  $g(\cdot, \cdot)$  is a function which depends on the input bit  $b_n$  (or  $c_n$  for the second encoder) and on the previous chaotic sample  $z_{n-1}$ ; its expression and meaning were seen in Chapter 4<sup>2</sup>. With these definitions, it is easy to see again that the recursion of Eq. (5.1) leaves the finite set  $S_Q = \{i \cdot 2^{-Q} | i = 0, \dots, 2^Q - 1\}$  invariant and, therefore, we can restrict Eq. (5.1) to  $S_Q$ . When  $Q \rightarrow \infty$ , Eq. (5.1) becomes simply the recursion by the chaotic maps  $f(\cdot, 0)$  or  $f(\cdot, 1)$ , depending on the value of  $b_n$ , but without small perturbations control. The Eqs. (5.1) and (5.2) apply also for the non-CSK case: it is enough to take the same map for  $b_n = 0$  and  $b_n = 1$ , i.e.,  $f_0(\cdot) = f_1(\cdot)$ . This case of CCM with small perturbations control alone is the same as described at the beginning of Chapter 3.

Throughout this chapter, we shall consider the following pairs of maps  $f_0(\cdot)$  and  $f_1(\cdot)$ :

1. Bernoulli shift map (BSM). Recall that, in this case

$$f_0(z) = f_1(z) = 2z \bmod 1. \quad (5.3)$$

This leads to the encoder already used in the examples of Chapters 3 and 4.

2. Tent map (TM), corresponding to equations

$$f_0(z) = f_1(z) = \begin{cases} 2z & 0 \leq z < \frac{1}{2} \\ 2 - 2z & \frac{1}{2} \leq z \leq 1 \end{cases}. \quad (5.4)$$

See Fig. 5.2.

3. The BSM and a shifted version of the same, following

$$f_0(z) = 2z \bmod 1, \quad (5.5)$$

$$f_1(z) = \begin{cases} 2z + \frac{1}{2} & 0 \leq z < \frac{1}{4} \\ 2z - \frac{1}{2} & \frac{1}{4} \leq z < \frac{1}{2} \\ 2z - \frac{3}{2} & \frac{3}{4} \leq z \leq 1 \end{cases}. \quad (5.6)$$

We will call the resulting system multi-Bernoulli shift map (mBSM). The corresponding maps are depicted in Fig. 5.3.

4. The tent map and a shifted version of the same, following (see Fig. 5.4)

$$f_0(z) = \begin{cases} 2z & 0 \leq z < \frac{1}{2} \\ 2 - 2z & \frac{1}{2} \leq z \leq 1 \end{cases}, \quad (5.7)$$

---

<sup>2</sup>See Eq. (4.4) in Section 4.2.1 of said chapter.

$$f_1(z) = \begin{cases} 2z & 0 \leq z < \frac{1}{4} \\ 2z - \frac{1}{2} & \frac{1}{4} \leq z < \frac{1}{2} \\ \frac{3}{2} - 2z & \frac{1}{2} \leq z < \frac{3}{4} \\ \frac{5}{2} - 2z & \frac{3}{4} \leq z \leq 1 \end{cases} \quad (5.8)$$

According to what was said for the mBSM example, we will call this system multi-tent map (mTM).

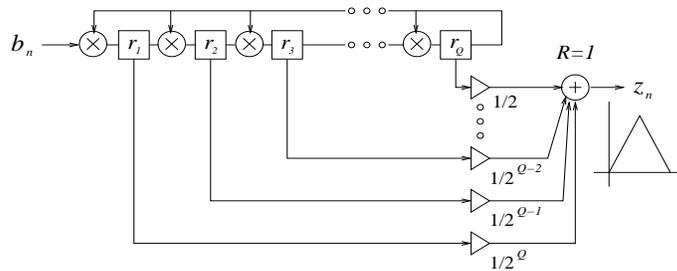


Figure 5.5. Encoding structure for the tent map CCM.

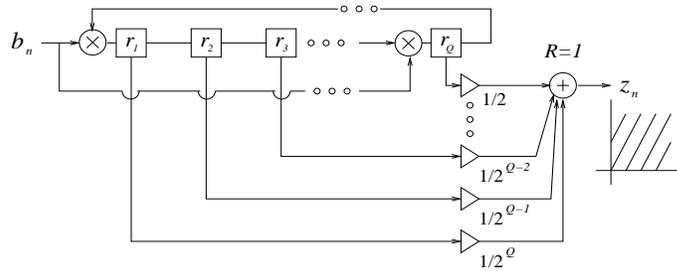


Figure 5.6. Encoding structure for multi-Bernoulli shift map CCM.

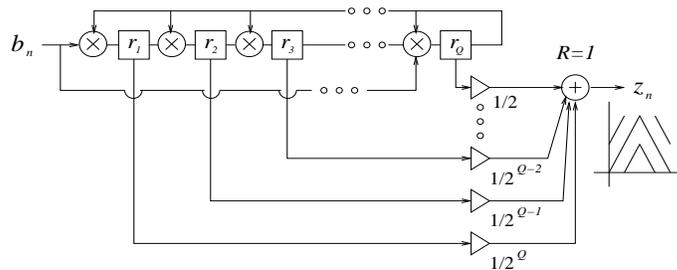


Figure 5.7. Encoding structure for the multi-tent map CCM.

In Chapter 3, we have seen how the map description of the BSM CCM as given by Eqs. (5.1) and (5.2), when restricted to  $S_Q$ , could be associated, under a trellis encoder view, with a related trellis coded modulation, since the encoding process resulted in a finite state machine. The system could be thus seen as consisting on a finite size shift register, a number of multipliers and an adder (see Fig. 3.1). We

also introduced in Chapter 4 the recursion given by  $g(b, z)$  (see Fig. 4.2), needed to get a good interleaver gain in a serially concatenated encoder, and showed it could just be considered a case of recursive precoding. In parallel concatenated coding, the importance of this feedback connection has also been established as a needed device to assure good performance [Benedetto and Montorsi, 1996], and so we keep it throughout the developments of this chapter. The TM encoder, which can also be restricted to  $S_Q$  by choosing a suitable initial condition (e.g.  $z_0 = 0$ ), has an associated finite state machine description, following

$$\begin{aligned} r_i &= r_{i-1} \otimes r_Q \quad i = Q, \dots, 2, \\ r_1 &= g(b_n, z_{n-1}), \\ z_n &= \sum_{i=1}^Q 2^{-(Q+1-i)} \cdot r_i, \end{aligned} \tag{5.9}$$

where  $g(b_n, z_{n-1}) = b_n \otimes r_Q$  (see Section 4.2.1). The Eq. (5.9), corresponding to the encoder structure shown in Fig. 5.5, is a quantized version of the binary expansion of the related real-valued outputs when using the TM with symbolic dynamics, just as the recursion of Eq. (4.5) gave also quantized samples from the binary expansion of the real-valued outputs corresponding to the BSM with symbolic dynamics [Schweizer and Schimming, 2001a; Schweizer and Schimming, 2001b]. The only particularity is, again, that restricting the output values to  $S_Q$  allows us to consider this *chaotic map driven by small perturbations* based encoder as a case of TCM. With respect to the instances of CSK + small perturbations, what we call here multi-map cases, the trellis encoder view keeps the same structure as the encoders of the corresponding basic map, and only the value for  $r_Q$  requires a change, so that the recursion follows the maps  $f_0(\cdot)$  or  $f_1(\cdot)$  according to the value of  $b_n$ . If  $b_n = 0$ , we just apply Eq. (4.5) for the mBSM, and Eq. (5.9) for the mTM. But, if  $b_n = 1$ , we have to follow the dynamics given by the shifted maps. It is easy to see that  $f_1(z) = (f_0(z) + 1/2) \bmod 1$ , and so we can apply the recursions of the  $b_n = 0$  case, but memory position  $r_Q$  has to take the complementary value, i.e.,  $r_Q^{b_n=1} = r_Q^{b_n=0} \otimes 1$ . That is to say, the recursions of Eqs. (4.5) and (5.9) are still valid for the multi-map cases, with the exception that  $r_Q = r_{Q-1} \otimes b_n$  for mBSM, and  $r_Q = (r_Q \otimes r_{Q-1}) \otimes b_n$  for mTM. The related trellis encoder structures can be seen in Figs. 5.6 and 5.7.

Though in all these explanations we have taken the sequence  $b_n$  as the binary input for the encoder, thus implicitly pointing towards CCM 1, the same applies to CCM 2, just by changing  $b_n$  by the interleaved sequence  $c_n$ . Note that the four encoders described have rate  $R = 1$  sample per bit, even the switched map ones (CSK type), and this is a difference with other standard CSK encoders, where several samples are produced for each input bit by performing a number of recursions through the corresponding map [Lau and Tse, 2003; Larson et al., 2006]. In this way, the overall rate for a parallel concatenation of two CCM of the kind described will be  $R = 1/2$  samples per bit. It is also evident that we can concatenate two different encoders and evaluate the potential advantages of using systems with different en-

coding structures. The CCM's described produce data  $x_n$  uniformly distributed in  $[-1, 1]$  for  $Q \rightarrow \infty$ , so that the output sequences will have power  $P \approx 1/3$  [Spratt, 2003].

With respect to the interleaver which performs the scrambling between the sequences  $b_n$  and  $c_n$ , it will be a bit interleaver with size  $N$ , as mentioned at the beginning of this section. Specifically, it will consist on an S-random interleaver [Divsalar and Pollara, 1995], which has already been described in Chapter 3 for SCCCM<sup>3</sup>. As we do not perform trellis termination for either of the encoders, the importance of a good interleaver design is even higher than in the case of SCCCM, where the outer convolutional encoder added at least some degree of structure to the input data for the CCM. If we have a simple  $N = N_1 \times N_2$  block interleaver, it is easy to see that the minimum squared Euclidean distance for all the possible CCM encoders whose values are limited to  $S_Q$  will be associated with input words  $\mathbf{b} = (b_1, \dots, b_{N-1}, 0)$  and  $\mathbf{b}' = (b_1, \dots, b_{N-1}, 1)$ , since  $\pi(N) = N$ , and these sequences are interleaved as  $\mathbf{c} = (c_1, \dots, c_{N-1}, 0)$  and  $\mathbf{c}' = (c_1, \dots, c_{N-1}, 1)$ . Therefore, the output words produced will have only two different values at the end of the sequence,  $x_{2N-1}$ ,  $x_{2N}$  and  $x'_{2N-1}$ ,  $x'_{2N}$ . For two CSK + small perturbations encoders, the differences between  $x_{2N-1}$  and  $x'_{2N-1}$  (first CCM) and  $x_{2N}$  and  $x'_{2N}$  (second CCM) will be located only in the values of register positions  $r_1$  and  $r_Q$ , affected by weights  $1/2^Q$  and  $1/2$  respectively. In this way, the squared Euclidean distance between such output sequences  $\mathbf{x}$  and  $\mathbf{x}'$  would be

$$d^2(\mathbf{x}, \mathbf{x}') = \sum_{k=1}^{2N} (x_k - x'_k)^2 = 4 \left( 2\frac{1}{2} + 2\frac{1}{2^Q} \right)^2 = \left( 2 + \frac{1}{2^{Q-2}} \right)^2. \quad (5.10)$$

For encoders based only in small perturbations, the differences in the last values of the output words will be associated to the value of  $r_1$ , so that the squared Euclidean distance between such words would be

$$d^2(\mathbf{x}, \mathbf{x}') = 4 \left( 2\frac{1}{2^Q} \right)^2 = \left( \frac{1}{2^{Q-2}} \right)^2. \quad (5.11)$$

See how in this case the concatenation of two non-CSK CCM's lead to a very poor minimum squared Euclidean distance, which vanishes to 0 when  $Q \rightarrow \infty$ . Such minimum squared Euclidean distance error events associated with binary error events of Hamming weight 1 can be easily avoided either by performing trellis termination or by changing such simple interleaver design [Hokfelt et al., 2001]. We choose here the second possibility.

Therefore, with an S-random interleaver with  $S > 1$  (i.e., when the interleaver is not a purely random interleaver), we avoid the possibility of having Hamming weight 1 binary error events of the type described leading to a low squared Euclidean distance between the output words of the resulting code. Moreover, due to the feedback connection given by  $g(b, z)$ , the minimum squared Euclidean distance error

<sup>3</sup>See Section 4.2.1.

events for each of the individual CCM's seen cannot take place for an input binary error event with Hamming weight 1. Therefore, their concatenation with an S-random interleaver will determine binary error events leading to low output squared Euclidean distances only for higher input Hamming weights. These facts have been established for binary turbocodes and also hold here [Benedetto and Montorsi, 1996]. We will analyze the consequences of this in Section 5.4.

Recall that the mentioned encoders resemble Ungerboeck's trellis coded modulation (TCM) [Ungerboeck, 1982], and so the turbo system thus built could be thought as an instance of turbo trellis coded modulation (TTCM) [Robertson and Wörz, 1998]. Nonetheless, there are some important differences, as the system proposed here has as scope to have broadband chaotic (or at least chaotic-like) samples in the channel, not to provide a bandwidth-efficient modulation. Therefore, as was seen in Chapter 4 for SCCCM, the constituent encoders work at a rate of 1 bit per symbol with amplitude modulation, instead of using phase shift keying (PSK) or other more complex modulations. One advantage over TTCM is that we do not require  $m$ -wise interleavers ( $m$  being the bits per symbol in the TTCM setup) or symbol interleavers [Wu and Ogiwara, 2004], and we can work with simpler bitwise permutations.

### 5.2.2. Channel

The channels considered will be the same introduced in Chapter 3. They were also employed in Chapter 4 to assess the performance of SCCCM in the presence of different kinds of distortion. We review them here briefly for clarity's sake. As mentioned before, the samples  $x_k$  are sent to the channel in baseband. In the channel, shown in Fig. 5.1 as a black box,  $x_k$  is subjected at least to the effects of additive white Gaussian noise (AWGN) and possibly other sources of distortion. We will have three kinds of channels with the following outputs, for  $k = 1, \dots, 2N$ :

1. AWGN channel. The sequence arriving at the decoder side,  $\mathbf{r} = (r_1, \dots, r_{2N})$ , will be

$$r_k = y_k + n_k = x_k + n_k, \quad (5.12)$$

where  $y_k = x_k$ , and  $n_k$  are *iid* samples of a Gaussian RV with zero mean and power  $\sigma^2$ . Recall that  $x_{2n-1}$ ,  $n = 1, \dots, N$  are the chaotic symbols from the first chaos coded modulator, and  $x_{2n}$ ,  $n = 1, \dots, N$  are the chaotic symbols from the second chaos coded modulator.

2. ISI channel. It is described by a normalized FIR filter of length  $2M + 1$ , with impulse response  $\mathbf{h} = (h_{-M}, \dots, h_M)$ .  $r_k$  can be thus written as

$$r_k = y_k + n_k = \sum_{m=-M}^M h_m x_{k+m} + n_k. \quad (5.13)$$

The filter coefficients will be again those of Table 3.1. It is worth noting that this ISI channel, according with the alternating order in which the samples from each encoder are placed in the transmitted sequence, correlates the

chaotic samples from both constituent encoders. This will be of consequence in the final performance.

3. Fading channel. In this case, the chaotic signal will be affected by frequency-non selective (flat) uncorrelated fading. The fading amplitude samples  $a_k$  follow uncorrelated Rician RVs with parameter  $K$  and unit power.  $r_k$  can be written as

$$r_k = y_k + n_k = a_k x_k + n_k. \quad (5.14)$$

As the joint PCCCM system proposed has rate  $R = 1/2$  for two CCM of the kind described, then the relationship between the power of the noise and the signal to noise ratio in terms of bit energy to noise spectral density will be

$$\sigma^{-2} = 2 \frac{R E_b}{P N_0} = \frac{1 E_b}{P N_0}, \quad (5.15)$$

where  $P$  is the power of the chaos coded modulated signal and  $R = 1/2$ .

### 5.2.3. Iterative decoder

The decoder, shown on the right hand side of Fig. 5.1, consists on two SISO decoding blocks working iteratively over each set of received samples (the ones coming from the first encoder,  $r_{2n-1}$ , and the other ones coming from the second one,  $r_{2n}$ ,  $n = 1, \dots, N$ ). These SISO modules that perform sequence MAP decoding over the chaos coded modulated signal correspond to the modules already described in Chapter 3, Section 3.2.2. The soft outputs from the first decoder, in the form of log probability ratios,  $\Lambda(b_n; O)$ , conveniently interleaved, act as *a priori* inputs for the second decoder,  $\Lambda(c_n; I)$ , whose *a posteriori* outputs  $\Lambda(c_n; O)$  act in turn, when deinterleaved, as *a priori* values for the first one,  $\Lambda(b_n; I)$ . After a certain number of iterations, a decision may be drawn over the sum of soft outputs coming from both decoders (see Fig. 5.1) [Benedetto et al., 1996]

$$\Lambda(b_n) = \Lambda(b_n; O) + \Pi^{-1}[\Lambda(c_n; O)], \quad (5.16)$$

where  $\Pi^{-1}$  denotes the operation of the deinterleaver. We obtain thus the estimated received sequence  $\hat{b}_n$ ,  $n = 1, \dots, N$ . As usual, I stands for input and O, for output. This process is analogous to the iterative decoding process of SCCCM, with the exception that here both concatenated encoders are inner encoders and so both SISO decoders work on the output channel samples  $r_k$ . The output log probability ratios can be thus expressed as

$$\Lambda(b_n; O) = \log \frac{p(b_n = 1 | \mathbf{r}^b, \Theta)}{p(b_n = 0 | \mathbf{r}^b, \Theta)}, \quad (5.17)$$

$$\Lambda(c_n; O) = \log \frac{p(c_n = 1 | \mathbf{r}^c, \Theta)}{p(c_n = 0 | \mathbf{r}^c, \Theta)}, \quad (5.18)$$

for  $n = 1, \dots, N$ . Vectors  $\mathbf{r}^b$  and  $\mathbf{r}^c$  contain the channel samples corresponding to each CCM encoder / SISO decoder pair, i.e.,  $\mathbf{r}^b = (r_1, r_3, \dots, r_{2N-1})$  and  $\mathbf{r}^c = (r_2, r_4, \dots, r_{2N})$ . As usual, the dummy vector  $\Theta$  denotes the channel parameters known to the decoders. Note that both decoders work on non-terminated trellises, and, as usual, this has to be taken into account when initializing the SISO decoding algorithm in the backward pass.

We shortly review the particularities for each kind of channel. The AWGN channel and the ISI channel employ the same metrics, calculated as functions of  $(r_k - x_k)^2$ , and so the extrinsic information at the output of the first SISO is simply written as

$$\Lambda(b_n; \text{O}) = \log \frac{P(b_n = 1 | \mathbf{r}^b)}{P(b_n = 0 | \mathbf{r}^b)}. \quad (5.19)$$

In the fading channel without CSI, the metrics are functions of  $(r_k - \eta_a x_k)^2$  [Hall and Wilson, 1998] and so the first SISO module produces the log probability ratios

$$\Lambda(b_n; \text{O}) = \log \frac{P(b_n = 1 | \mathbf{r}^b, \eta_a)}{P(b_n = 0 | \mathbf{r}^b, \eta_a)}, \quad (5.20)$$

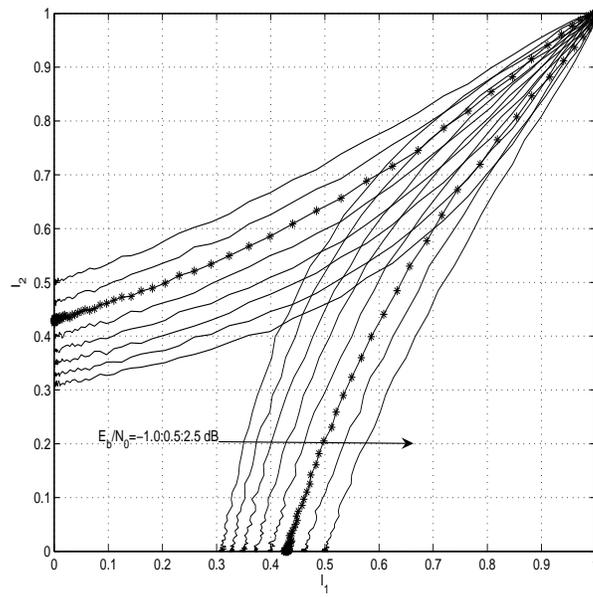
Finally, for the fading channel with perfect CSI, the metrics are functions of  $(r_k - a_k x_k)^2$ , so that the extrinsic information is expressed as

$$\Lambda(b_n; \text{O}) = \log \frac{P(b_n = 1 | \mathbf{r}^b, \mathbf{a}^b)}{P(b_n = 0 | \mathbf{r}^b, \mathbf{a}^b)}, \quad (5.21)$$

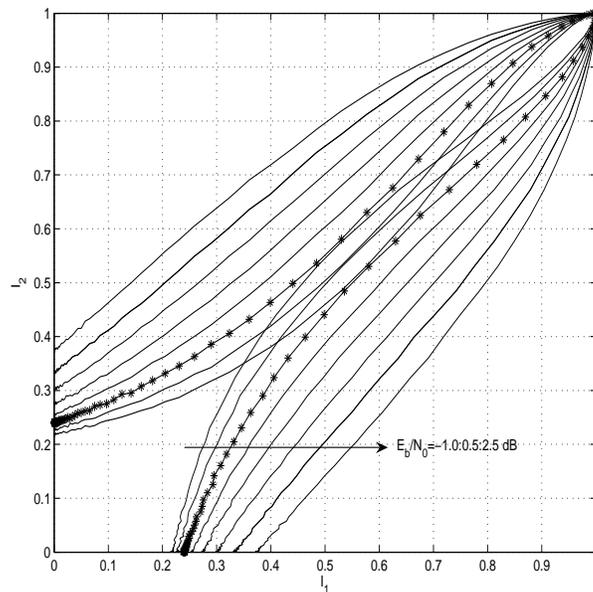
where  $\mathbf{a}^b = (a_1, a_3, \dots, a_{2N-1})$  is the vector of fading amplitudes affecting the samples from the first CCM. The same expressions hold for the second CCM, but with  $c_n$  instead of  $b_n$ ,  $\mathbf{r}^c$  instead of  $\mathbf{r}^b$ , and  $\mathbf{a}^c = (a_2, a_4, \dots, a_{2N})$  instead of  $\mathbf{a}^b$ .

### 5.3. Convergence analysis

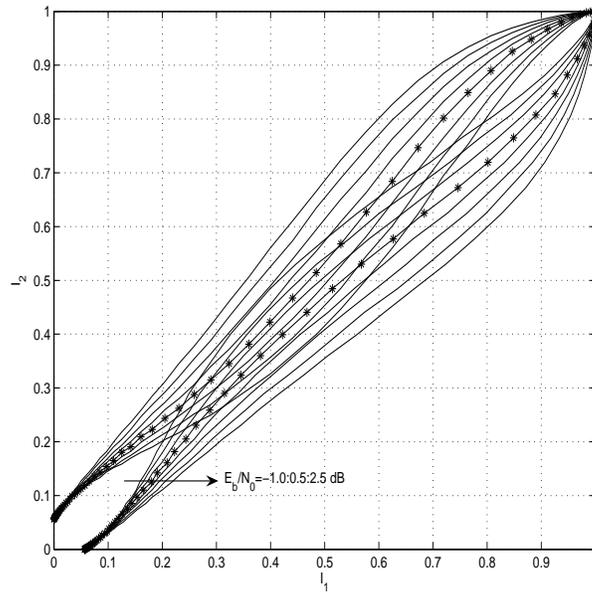
We saw in Section 4.3 that the EXtrinsic Information Transfer (EXIT) charts were useful to predict the general behaviour of the iterative decoding process of a SCCCM system. The EXIT charts have also been successfully employed TTCM systems [Chen and Haimovich, 2004; Schlegel and Pérez, 2004; Kliewer et al., 2006], and therefore we expect that they will also provide us here with information about the possibilities of the parallel combination of CCM encoders under iterative decoding. The main difference with respect to what was seen in Section 4.3 is that here both encoders send samples to the channel, and so both transfer curves will be affected by the channel distortion level. The procedure to get these transfer curves is the same as seen in the mentioned section: for each  $E_b/N_0$  in the channel and each additional impairment process (fading with or without CSI, or ISI, or just nothing else in the case of simple AWGN), we have run 100 simulations with blocks of  $N = 10000$  input bits. We have fed again into the SISO modules Gaussian distributed log probability ratios following Eq. (4.18), and we have calculated numerically the integral of Eq. (4.19) to give the input value  $I(\text{I})$ . The output values  $I(\text{O})$  have been calculated by



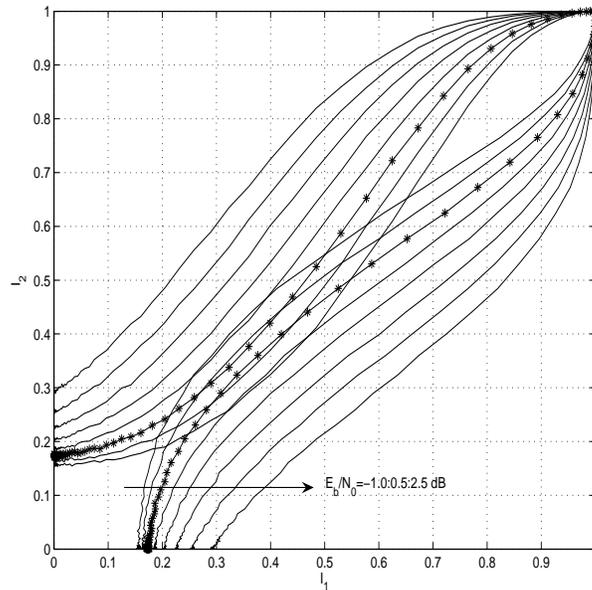
**Figure 5.8.** EXIT chart for two BSM CCM with  $Q = 5$  in the AWGN channel. The  $E_b/N_0$  goes from left to right (lower curves) and from bottom to top (upper curves) from  $-1.0$  dB to  $2.5$  dB in  $0.5$  dB steps. Marked with '\*':  $E_b/N_0$  threshold at about  $1.5$  dB.



**Figure 5.9.** EXIT chart for two mBSM CCM with  $Q = 5$  in the AWGN channel. The  $E_b/N_0$  goes from left to right (lower curves) and from bottom to top (upper curves) from  $-1.0$  dB to  $2.5$  dB in  $0.5$  dB steps. Marked with '\*':  $E_b/N_0$  threshold at about  $0.0$  dB.



**Figure 5.10.** EXIT chart for two TM CCM with  $Q = 5$  in the AWGN channel. The  $E_b/N_0$  goes from left to right (lower curves) and from bottom to top (upper curves) from  $-1.0$  dB to  $2.5$  dB in  $0.5$  dB steps. Marked with '\*':  $E_b/N_0$  threshold at about  $0.5$  dB.



**Figure 5.11.** EXIT chart for two mTM CCM for  $Q = 5$  in the AWGN channel. The  $E_b/N_0$  goes from left to right (lower curves) and from bottom to top (upper curves) from  $-1.0$  dB to  $2.5$  dB in  $0.5$  dB steps. Marked with '\*':  $E_b/N_0$  threshold at about  $0.0$  dB.

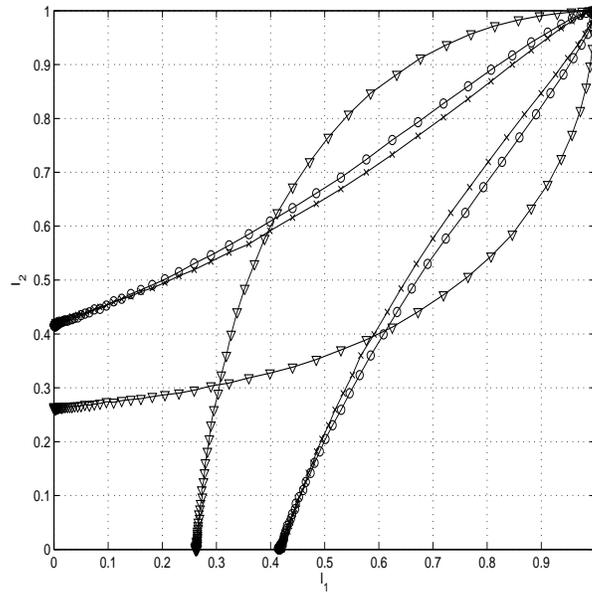
integrating Eq. (4.16) numerically after computing the histogram of  $\Lambda$  at the output of each SISO. Finally we have averaged over the ensemble of runs.

In Figs. 5.8, 5.9, 5.10 and 5.11, we can see the EXIT charts in the AWGN channel for a combination of two equal CCM's of the kind described in Subsection 5.2.1. The lower curves correspond to the second CCM and the upper ones to the first CCM (see right hand side of Fig. 5.1), so that axis  $I_1$  is the input mutual information and axis  $I_2$  is the output mutual information for the first SISO (and viceversa for the second SISO). The iterative decoding process works as seen in Section 4.3, with a series of hops between the upper and lower curves for a given  $E_b/N_0$ , starting from the point where the upper curve intersects the vertical axis  $I_1 = 0$ . In said figures, we have also highlighted the curves representing the approximate  $E_b/N_0$  threshold where the decoding process can theoretically converge to the (1, 1) point. Apart from the fact that the thresholds suggest differences in the final BER performance, the shape of the curves and the evolution of the fixed point as the  $E_b/N_0$  evolves from  $-1.0$  dB to  $2.5$  dB will provide us with valuable information about the general behaviour of the iterative decoding process.

Following Fig. 5.8, we can see that the fixed point where the lower and upper curves intersect for a given  $E_b/N_0$  tends gradually to the (1, 1) point, so that the threshold highlighted at  $1.5$  dB is only a visual approximation and we could be in fact very near it, but still not at (1, 1). As a consequence, we can guess that there will be no distinct waterfall region with a steep slope in the BER curve for this concatenation of two BSM CCM's. On the contrary, we can think of the BER tending gradually with a smoother slope to the final error floor. The simulations will illustrate the accurateness of this prediction. With respect to Figs. 5.9, 5.10 and 5.11, we can see in all the cases that the fixed point changes abruptly from a point near (0.5, 0.5) to the (1, 1) point, so that most probably we will have a waterfall region in the BER curves with a steep slope where the error rate changes fast from a high value to a value near the final error floor.

There are other differences to take into account. Though Figs. 5.9 and 5.11 have almost the same  $E_b/N_0$  threshold, we can see that the value chosen for two concatenated mBSM CCM's is more conservative, so that it is to expect a slightly better behaviour from this concatenated encoder with respect to the two mTM CCM's case. Regarding the two TM CCM's case (see Fig. 5.10), we can also guess that, once reached the thresholds  $E_b/N_0$ , this concatenated system would require more decoding iterations to get a given BER than the rest of concatenated systems considered. The initial decoding gain grows almost nothing with growing  $E_b/N_0$  and we would always need at least 4 or 5 iterations just to reach the rightmost upper quadrant delimited by  $I_1 = 0.5$  and  $I_2 = 0.5$ .

In Figs. 5.12, 5.13, 5.14 and 5.15, we have plotted the EXIT charts for the combination of two equal CCM's in the ISI channel. We may expect some deviation from the behaviour shown here with respect to the simulations, since the mutual information transfer functions have been drawn with only the samples from one encoder on the channel, while in the system described in Section 5.2 every other sample comes from the other decoder. This means, by one side, that the samples

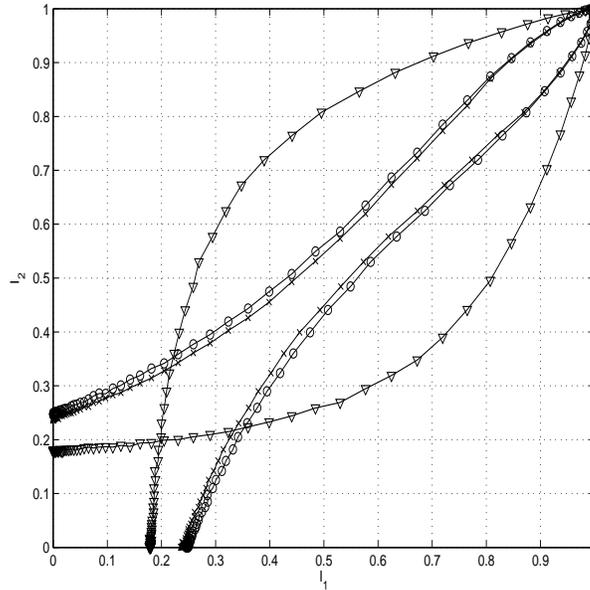


**Figure 5.12.** EXIT chart for two BSM CCM with  $Q = 5$  in the ISI channel with different levels of distortion. 'x': low ISI,  $E_b/N_0 = 1.5$  dB. 'o': moderate ISI,  $E_b/N_0 = 2.0$  dB. '▽': high ISI,  $E_b/N_0 = 7.5$  dB.

$r_k$  simulated to get the EXIT charts are affected by a higher degree of correlation from the own chaotic samples of the testing encoder, but, on the other side, they are not correlated with the samples from the other decoder. Since this is not the real situation in the ISI channel, we foresee further mismatches between the predictions and actual values for the thresholds in this case.

First of all, it is clear that none of the parallel concatenated systems will cope with the high ISI case regardless of the noise power in the channel, since the mutual information transfer functions exhibit an early crossing point and the lower transfer function  $I_1 = T_2(I_2, \Theta)$  (second SISO) is never confined below the  $I_2 = I_1$  line, nor the upper transfer function  $I_2 = T_1(I_1, \Theta)$  (first SISO) over this same line. The information available from the channel is so severely distorted that there is no possibility of convergence.

On the other hand, the low and moderate ISI cases show a slight degradation in the convergence thresholds. The TM and the mTM cases have a loss of about 0.5 dB for low ISI and about 1.0 dB for moderate ISI. In the mBSM case, there seems to be no loss for low ISI and only 0.5 dB for moderate ISI. Nevertheless, the window between the mutual information transfer functions for mBSM in AWGN was taken slightly more conservative (bigger) than the other two cases mentioned. Therefore, it is to expect that, since the convergence of the concatenation of two mBSM CCM in AWGN would be a bit below the threshold for two mTM CCM (they are the same following Figs. 5.9 and 5.11), there will also be similar losses for low and moderate ISI. In the case of the concatenation of two BSM CCM's, there seems to be the same situation as for the concatenation of two mBSM CCM's.



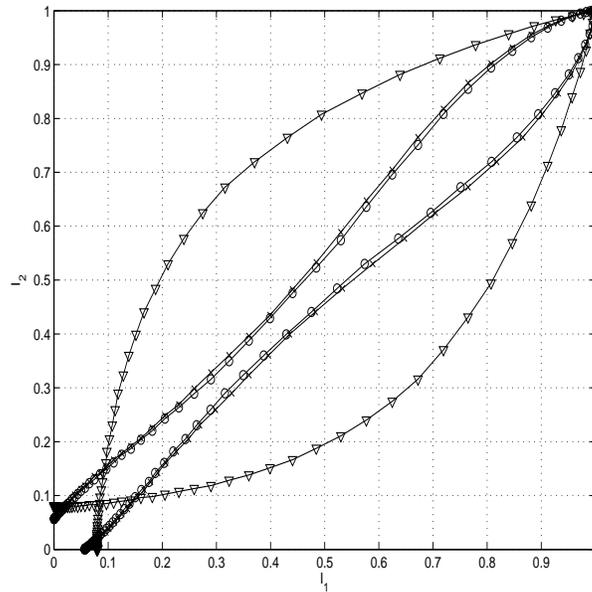
**Figure 5.13.** EXIT chart for two *mBSM CCM* with  $Q = 5$  in the ISI channel with different levels of distortion. 'x': low ISI,  $E_b/N_0 = 0.0$  dB. 'o': moderate ISI,  $E_b/N_0 = 0.5$  dB. '∇': high ISI,  $E_b/N_0 = 7.5$  dB.

Nevertheless, the convergence characteristics of the BSM, with the crossing point of the transfer functions evolving smoothly to the  $(1, 1)$  point both for the AWGN and the ISI channels (excepting the high ISI case), will not determine a distinct BER waterfall region and very little we can say about the  $E_b/N_0$  for which the error floor is reached.

In Figs. 5.16, 5.17, 5.18, 5.19, 5.20 and 5.21, we show the EXIT charts for the concatenation of two identical CCM's in the fading channel, both with CSI and without CSI. We do not consider the BSM CCM, since the information that we can draw from its EXIT charts is very limited. In the Rayleigh fading channel ( $K = 0$ ), there will be for all the cases a loss of about 1.0 dB when there is side information at the decoder (CSI case), and around 2.0 dB when said side information is not available (case without CSI). As already seen with the SCCCM system, the degradation in the threshold  $E_b/N_0$  tends to vanish as we approach the  $K \rightarrow \infty$  case (AWGN channel) and the differences respecting the CSI and the non-CSI case also tend to collapse to the values in AWGN alone (see Figs. 5.17, 5.19 and 5.21). Specifically, when  $K = 5$ , the case of two *mTM CCM*'s shows that the losses are reduced to 0.5 dB (CSI) and to 1.0 dB (without CSI).

Though not shown, we have verified that the EXIT charts seem to be unaffected when changing the quantization level<sup>4</sup>  $Q$ , and so we have taken the value  $Q = 5$  for all cases. This value is not too low to lose the chaotic properties of the signal  $x_n$  in the channel, and not too high to make decoding unfeasible. Note also that the EXIT

<sup>4</sup>Recall what was seen in Chapter 4.

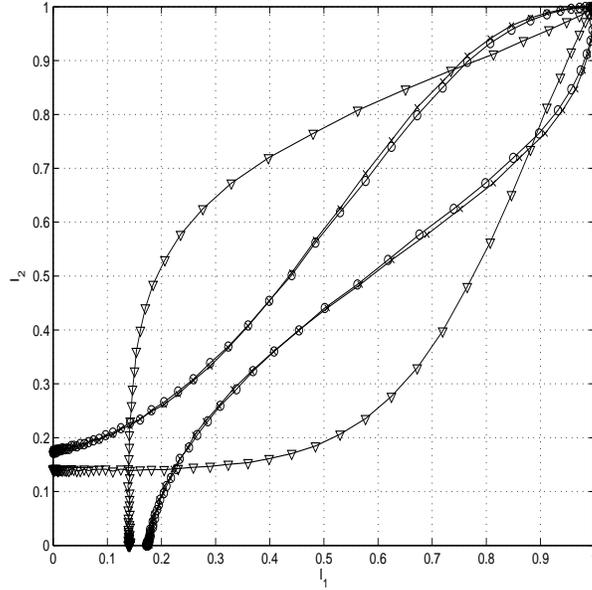


**Figure 5.14.** EXIT chart for two TM CCM with  $Q = 5$  in the ISI channel with different levels of distortion. 'x': low ISI,  $E_b/N_0 = 1.0$  dB. 'o': moderate ISI,  $E_b/N_0 = 1.5$  dB. 'v': high ISI,  $E_b/N_0 = 7.5$  dB.

charts for the distorting channels have been drawn again under the assumption of Gaussian distributed input log probability ratios, and therefore we expect to get some mismatch between the predicted thresholds and the actual ones got by simulation. We have to take into account as well that these EXIT charts assume perfectly uncorrelated input and output mutual information in each SISO decoder, and therefore the use of a real and finite size interleaver will introduce the usual degradation in the  $E_b/N_0$  thresholds. We will verify this in Section 5.5 with plots of the actual average trajectory of the mutual information.

## 5.4. Minimum squared Euclidean distance analysis

In Chapter 4 we saw that, for SCCCM, it was easier to give an estimation of the bit error probability in the waterfall region thanks to the reasonable assumption of a BIOS channel, but the ML analysis for the  $E_b/N_0 \rightarrow \infty$  case needed to give a bound for the bit error probability in the error floor region was hindered by the nonlinear structure of the CCM. This same happens with PCCCM, since we cannot rely either on the ML analysis and use a transfer function bound approach as is usually made with turbocodes [Divsalar et al., 1995; Babich et al., 1998]. Nevertheless, we are going to see that it is easier to provide a bound for the error floor region, even though we are working with two nonlinear CCM's. We are also specially interested in the error floor with parallel concatenation, since it is a known fact that the parallel concatenated systems, while having lower  $E_b/N_0$  thresholds for the BER waterfall region in comparison to analogous serially concatenated systems, are affected by a

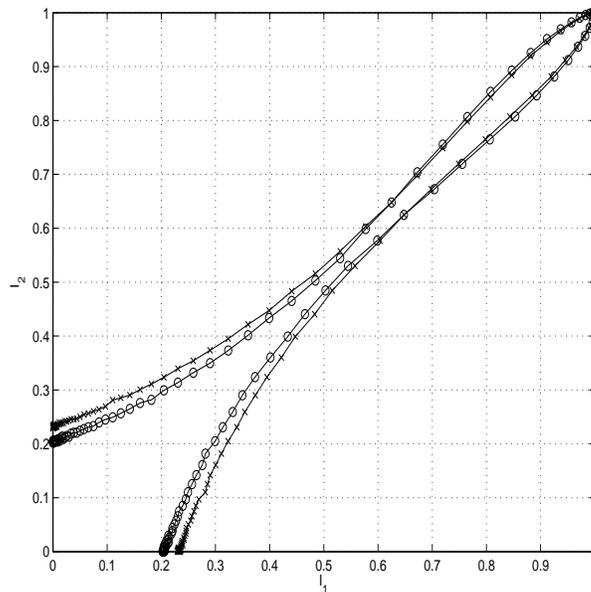


**Figure 5.15.** EXIT chart for two  $m\text{TM}$  CCM with  $Q = 5$  in the ISI channel with different levels of distortion. 'x': low ISI,  $E_b/N_0 = 0.5$  dB. 'o': moderate ISI,  $E_b/N_0 = 1.0$  dB. '∇': high ISI,  $E_b/N_0 = 7.5$  dB.

higher BER in the error floor region [Garello et al., 2001]. Thus, estimating this error floor and its nature can help us to get a good understanding of the system and the way to design better alternatives.

Let us denote as  $\mathbf{b}$  and  $\mathbf{b}'$  two binary input words of size  $N$ , and  $\mathbf{x}$  and  $\mathbf{x}'$  the resulting PCCCM encoded codewords of size  $2N$ . The difference between  $\mathbf{b}$  and  $\mathbf{b}'$  is an input binary error event  $\mathbf{e} = \mathbf{b} \otimes \mathbf{b}'$  with Hamming weight  $w(\mathbf{e})$  and length  $L$ , and the resulting difference between  $\mathbf{x}$  and  $\mathbf{x}'$  is a pairwise error event with squared Euclidean distance  $d_E^2 = \sum_{n=1}^{2N} (x_n - x'_n)^2$ . Note that  $\mathbf{b}$  and  $\mathbf{b}'$  only differ in  $L$  bits, and that  $\mathbf{x}$  and  $\mathbf{x}'$  may or may not differ in all the  $2N$  possible samples. Let us denote as  $d_H(\mathbf{x}, \mathbf{x}')$  the Hamming distance between codewords, defined as the number of positions where  $x_n \neq x'_n$ .

The good results of the turbocode structure resulting from the parallel concatenation through a bit interleaver rely in the fact that, with a good interleaver design, the input binary error events with low Hamming weight susceptible to produce on any of the individual encoders a pairwise error event with low squared Euclidean distance are conveniently broken, so that the possibility of having the same low Hamming weight error events simultaneously on both sequences never arises. Nevertheless, good candidates for the minimum squared Euclidean distance events are normally to be found among some combination of the low Hamming weight binary error events, which are difficult to spread efficiently in both binary input words  $\mathbf{b}$  and  $\mathbf{c}$  [Schlegel and Pérez, 2004]. Due to this, just before proceeding with the minimum squared Euclidean distance analysis for PCCCM, we will characterize the error events of all the individual CCM's considered in Subsection 5.2.1. While the

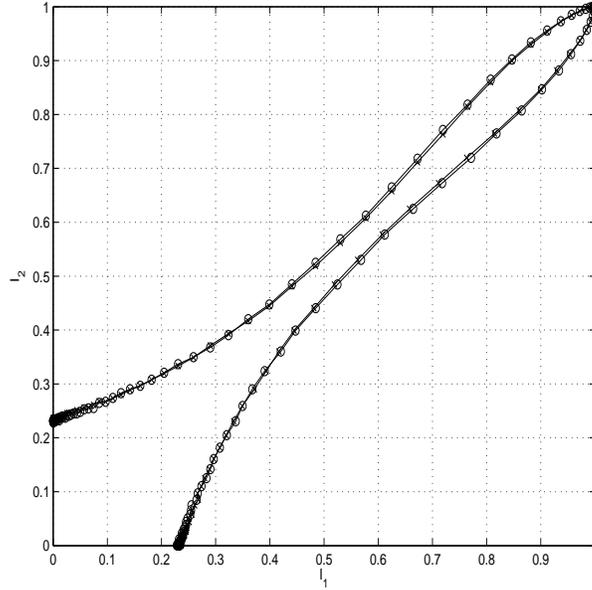


**Figure 5.16.** EXIT chart for two *mBSM CCM* with  $Q = 5$  in the fading channel for  $K = 0$  (Rayleigh fading). 'x': with CSI,  $E_b/N_0 = 1.0$  dB. 'o': without CSI,  $E_b/N_0 = 2.0$  dB.

searching of the minimum squared Euclidean distance for a CCM could be a tough task [Kozic, 2006], and the recursive connection does not allow us to apply here the approach of [Kozic et al., 2003b] based upon the dynamics of the maps, it will result easier to rely on the examination of the error events of lowest input Hamming weight  $w(\mathbf{e})$  and lowest length  $L$ . Though no formal proof is given for the statements made in the following regarding the distances and error events, the results are easily verified by considering the finite state machine nature of these CCM's.

1. In the case of the BSM CCM with the recursive input  $g(z, b)$ , we saw in Subsection 4.2.1 of Chapter 4 that the lowest Hamming weight binary input error event was given by  $\cdots b_{n-1}, b_n, b_{n+1} \cdots, b_{n+Q-1}, b_{n+Q}, b_{n+Q+1}, \cdots$  vs  $\cdots b_{n-1}, b'_n, b_{n+1} \cdots, b_{n+Q-1}, b'_{n+Q}, b_{n+Q+1}, \cdots$ , corresponding to a binary error event  $\mathbf{e} = \mathbf{b} \otimes \mathbf{b}' = (0, \cdots, 0, 1, (Q-1 \text{ 0's}), 1, 0, \cdots, 0)$  of length  $L = Q + 1$  and Hamming weight  $w(\mathbf{e}) = 2$ . This error event produces a codeword error event with Hamming distance  $d_H(\mathbf{x}, \mathbf{x}') = Q$ . The BSM CCM is a special case of CCM due to its almost linear structure, since it complies with the uniform error property (UEP) [Biglieri and McLane, 1991], and the resulting squared Euclidean distance between the associated codewords  $\mathbf{x}$  and  $\mathbf{x}'$ , besides being the minimum squared distance of the code, does not depend on the values of  $\mathbf{b}$  and  $\mathbf{b}'$ , or on the previous value<sup>5</sup> of  $x_{n-1} = x'_{n-1}$ , but only on the binary error event  $\mathbf{e}$  itself. Recall that this minimum squared Euclidean distance could be

<sup>5</sup>Which is equivalent to the starting state of the error event in the finite state machine description of the CCM.



**Figure 5.17.** EXIT chart for two mBSM CCM with  $Q = 5$  in the fading channel for  $K = 20$ . 'x': with CSI,  $E_b/N_0 = 0.0$  dB. 'o': without CSI,  $E_b/N_0 = 0.0$  dB.

calculated as

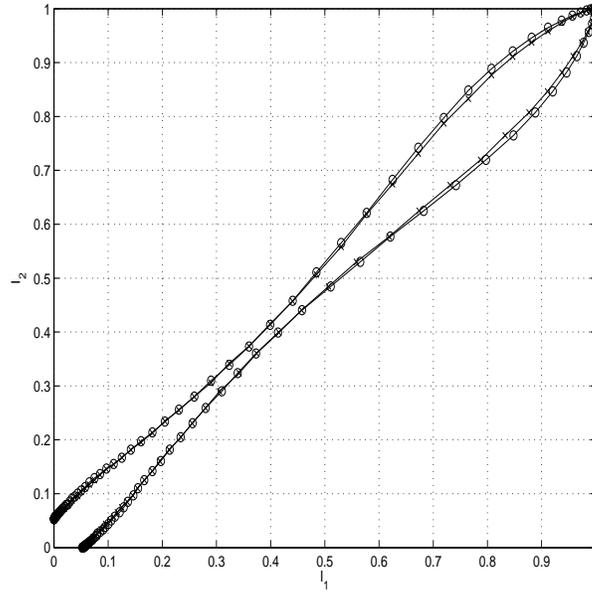
$$d_{\min}^2 = \sum_{n=m}^{L+m-1} (x_n - x'_n)^2 = 4 \sum_{i=1}^Q \frac{1}{4^i} = \frac{4}{3} \left(1 - \frac{1}{4^Q}\right), \quad (5.22)$$

and that we could approximate it without great error by the value corresponding to  $Q \rightarrow \infty$ ,  $d_{\min}^2 = 4/3$ , when  $Q \geq 4$ . There are also other possible error events with Hamming weight 2 and length  $L_p = p(L - 1) + 1 = pQ + 1$ , with  $p > 1$ ,  $p \in \mathbb{N}$ , but their associated squared Euclidean distance would be around  $p$  times the distance of Eq. (5.22).

2. For the TM CCM, the situation is slightly different. In this case, we have a first binary error event  $\mathbf{e} = (0, \dots, 0, 1, (Q - 1 \text{ 1's}), 1, 0, \dots, 0)$  with Hamming weight  $w(\mathbf{e}) = Q + 1$  and Hamming distance  $d_H(\mathbf{x}, \mathbf{x}') = Q$ . The exact value of the squared Euclidean distance between codewords depends on the previous value of  $x_{n-1} = x'_{n-1}$  and on the exact binary patterns in  $\mathbf{b}$  and  $\mathbf{b}'$ , but it has an absolute minimum value of

$$d_{\min}^2 = \sum_{n=m}^{L+m-1} (x_n - x'_n)^2 = 4 \sum_{i=1}^Q \left(\frac{1}{2^{Q-1}}\right)^2 = Q \cdot \frac{1}{4^{Q-1}}, \quad (5.23)$$

which tends to zero as  $Q \rightarrow \infty$ . This is one of the reasons why the TM CCM works so poorly when used alone [Kozic et al., 2003b], and so led us to consider only antisymmetric maps in Chapter 2. Nevertheless, this error event



**Figure 5.18.** EXIT chart for two TM CCM with  $Q = 5$  in the fading channel for  $K = 0$  (Rayleigh fading). 'x': with CSI,  $E_b/N_0 = 2.0$  dB. 'o': without CSI,  $E_b/N_0 = 3.0$  dB.

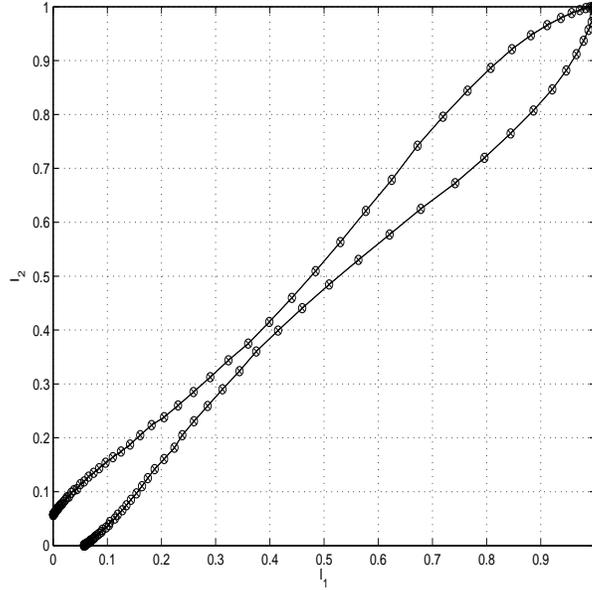
has Hamming weight  $w(\mathbf{e}) = Q$  and so, for values of  $Q \geq 4$ , it will not be a good candidate for the most probable BER error events in the error floor region of the PCCCM system.

There is another error event with Hamming weight  $w(\mathbf{e}) = 2$ , length  $L = Q + 2$  and Hamming distance  $d_H(\mathbf{x}, \mathbf{x}') = Q + 1$ , with structure  $\mathbf{e} = \mathbf{b} \otimes \mathbf{b}' = (0, \dots, 0, 1, (Q \text{ 0's}), 1, 0, \dots, 0)$ . In this case the UEP does not hold either and the squared Euclidean distance depends as well on  $x_{n-1}$  and on  $\mathbf{b}, \mathbf{b}'$ . The minimum squared Euclidean distance for this set of events is

$$d_{\min}^2 = \sum_{n=m}^{L+m-1} (x_n - x'_n)^2 = \frac{1}{4^{Q-1}} + \sum_{i=1}^Q \frac{1}{4^{i-1}} = \frac{1}{4^{Q-1}} + \frac{4}{3} \left(1 - \frac{1}{4^Q}\right), \quad (5.24)$$

which tends to  $4/3$  as  $Q \rightarrow \infty$ .

If these are the error events we have to consider in our parallel concatenated system, we see that we get a big improvement with respect to the use of this TM CCM alone, where the dominant error events would be the ones described previously with Hamming weight  $w(\mathbf{e}) = Q$  and very low squared Euclidean distance. Note that this TM used alone is a catastrophic encoder, since two input sequences differing virtually in an infinite number of bits for  $Q \rightarrow \infty$  can generate two output sequences with an squared Euclidean distance tending to 0 [Andersson, 1998]. In any case, though the UEP does not hold for these error events with Hamming weight 2, it is true that the set of all possible squared Euclidean distances for different  $x_{n-1}$ ,  $\mathbf{b}$  and  $\mathbf{b}'$  is limited to  $2^{Q-1}$  values. This



**Figure 5.19.** EXIT chart for two TM CCM with  $Q = 5$  in the fading channel for  $K = 50$ . 'x': with CSI,  $E_b/N_0 = 0.5$  dB. 'o': without CSI,  $E_b/N_0 = 0.5$  dB.

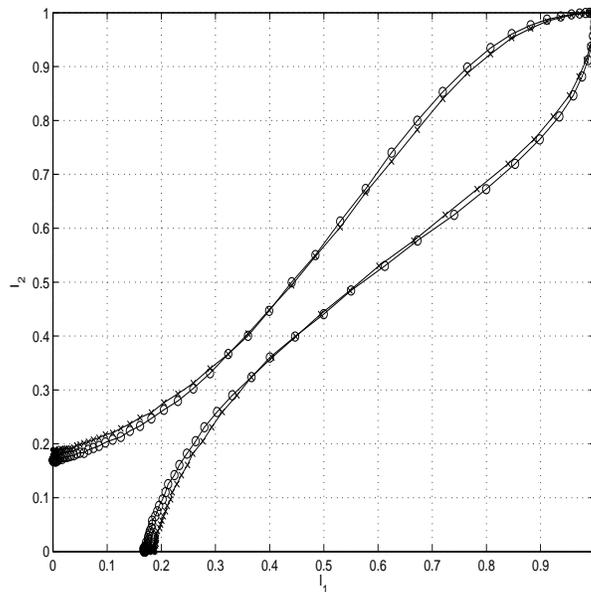
can be verified in the histogram of Fig. 5.22, taken after generating all the  $2^Q \cdot 2^{Q+1}$  different possibilities<sup>6</sup>. Let us denote as  $D_2^L$  the set of all the possible squared Euclidean distances produced by the weight 2 error event with length  $L$ , that is to say

$$D_2^L = \{d_E^2 = d^2(\mathbf{x}, \mathbf{x}') \mid \mathbf{x} \leftrightarrow \mathbf{b}, \mathbf{x}' \leftrightarrow \mathbf{b}', b_m \neq b'_m, b_{m+L-1} \neq b'_{m+L-1}\}. \quad (5.25)$$

Note also that each of the  $2^{Q-1}$  patterns would appear equiprobably for an *iid* binary input sequence under the presence of the related binary error event, since  $x_{n-1}$  and  $\mathbf{b}$  will have equiprobable values. As seen with the BSM CCM, the TM CCM will also have in general binary error events of Hamming weight 2 with length  $L_p = p(L - 1) + 1 = p(Q + 1) + 1$ ,  $p > 1$ ,  $p \in \mathbb{N}$ , but with increasing squared Euclidean distances.

3. The mBSM CCM also exhibits a minimum length error loop with  $L = Q$ ,  $w(\mathbf{e}) = Q$  and  $d_H(\mathbf{x}, \mathbf{x}') = Q - 1$ . As in the previous case, the exact  $d_E^2$  depends on the initial state of the encoder when the loop starts (i.e., on the value of  $x_{n-1}$ ) and on the specific values of  $\mathbf{b}$  and  $\mathbf{b}'$ . For this error event, the minimum squared Euclidean distance is higher than 2. The next loop in length, with  $L = Q + 1$ , is the one that interests us, since it has  $w(\mathbf{e}) = 2$ . It also exhibits the known structure  $\mathbf{e} = (0, \dots, 0, 1, (Q - 1 \text{ 0's}), 1, 0, \dots, 0)$ , and a Hamming distance  $d_H(\mathbf{x}, \mathbf{x}') = Q$ . Again the exact value of  $d_E^2$  depends

<sup>6</sup>There are  $2^Q$  different starting states and  $2^{Q+1}$  different possibilities for  $\mathbf{b}$  and  $\mathbf{b}'$  such that  $\mathbf{e} = \mathbf{b} \otimes \mathbf{b}'$  without repeating the patterns of length  $L = Q + 2$ .



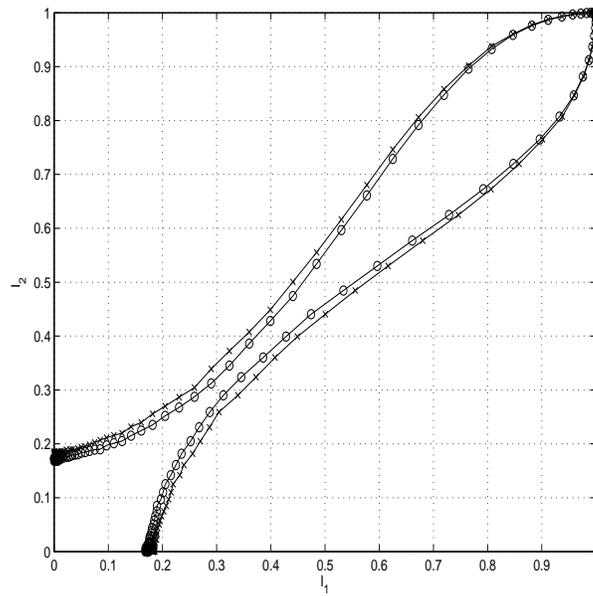
**Figure 5.20.** EXIT chart for two mTM CCM with  $Q = 5$  in the fading channel for  $K = 0$  (Rayleigh fading). 'x': with CSI,  $E_b/N_0 = 1.5$  dB. 'o': without CSI,  $E_b/N_0 = 2.5$  dB.

on  $x_{n-1}$  and on the binary sequences, and the minimum squared Euclidean distance is

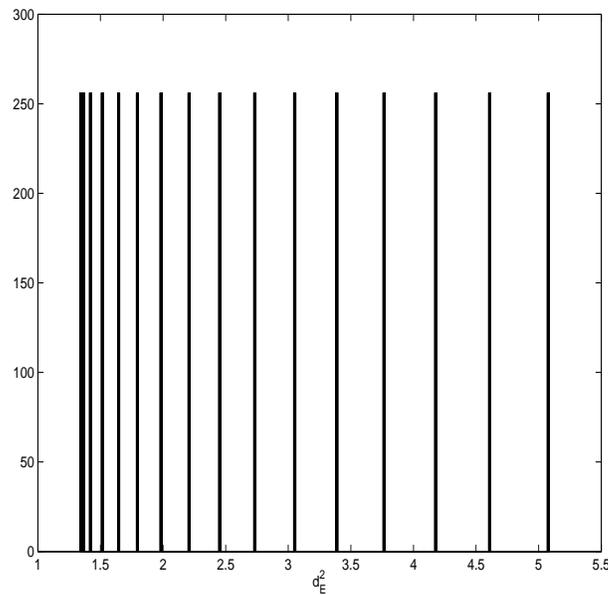
$$\begin{aligned} d_{\min}^2 &= \sum_{n=m}^{L+m-1} (x_n - x'_n)^2 = \sum_{i=1}^{Q-1} \frac{1}{4^i} + \left( \sum_{i=1}^{Q-1} \frac{1}{2^i} \right)^2 \\ &= \frac{1}{3} \left( 1 - \frac{1}{4^{Q-1}} \right) + \left( 1 - \frac{1}{2^{Q-1}} \right)^2, \end{aligned} \quad (5.26)$$

which tends to  $4/3$  as  $Q \rightarrow \infty$ . Nevertheless, as in the case of the TM CCM, the number of possible squared Euclidean distances in  $D_2^L$  is far less than  $2^Q \cdot 2^Q$ , as shown in Fig. 5.23, where we can see that there are only 8 possible different values when  $Q = 5$ . Error events of weight 2 and lengths  $L_p = p(L-1) + 1 = pQ + 1$ ,  $p > 1$ ,  $p \in \mathbb{N}$ , are also possible, but they have higher distances and offer a higher cardinality for  $D_2^{L_p}$ .

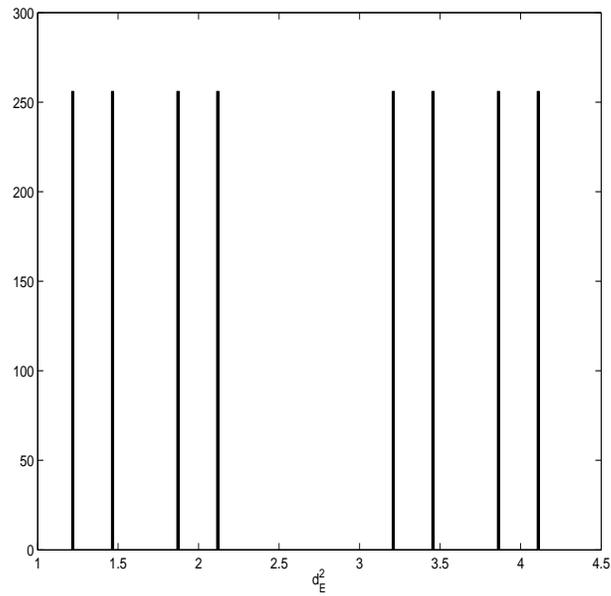
4. In the case of the mTM encoder, the analysis of the error events corresponding to low loop lengths  $L$  is more involved, since there are different possibilities depending on whether  $Q$  is odd or even, and we could have in some cases minimal loops with  $L = Q - 1$  or  $L = Q$ , but always with  $w(\mathbf{e}) > 2$ . We have in any case the error event with  $L = Q + 1$ ,  $w(\mathbf{e}) = Q + 1$  and  $d_H(\mathbf{x}, \mathbf{x}') = Q$ , but this time it does not exhibit a minimum squared Euclidean distance tending to 0 when  $Q \rightarrow \infty$  as in the TM case. For our purposes, the interesting error event is the one with  $L = Q + 2$ ,  $w(\mathbf{e}) = 2$  and  $d_H = Q + 1$ ,  $\mathbf{e}$  consisting again



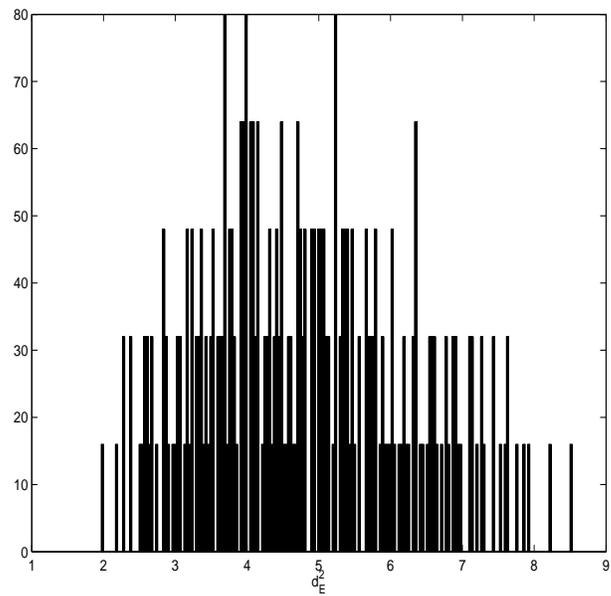
**Figure 5.21.** EXIT chart for two *mTM* CCM with  $Q = 5$  in the fading channel for  $K = 5$ . 'x': with CSI,  $E_b/N_0 = 0.5$  dB. 'o': without CSI,  $E_b/N_0 = 1.0$  dB.



**Figure 5.22.** Histogram of the possible output squared Euclidean distances for the Hamming weight 2 binary input error events in the case of a *TM* encoder with  $Q = 5$  for different starting states in the encoder and different binary input sequences corresponding to said input error event.



**Figure 5.23.** Histogram of the possible output squared Euclidean distances for the Hamming weight 2 binary input error events in the case of a mBSM encoder with  $Q = 5$  for different starting states in the encoder and different binary input sequences corresponding to said input error event.



**Figure 5.24.** Histogram of the possible output squared Euclidean distances for the Hamming weight 2 binary input error events in the case of a mTM encoder with  $Q = 5$  for different starting states in the encoder and different binary input sequences corresponding to said input error event.

in two ones separated by  $Q$  zeroes. The minimum squared distance for this error events is lower than, but very close to, 2, and tends to it when  $Q \rightarrow \infty$ . As we can see in the histogram of Fig. 5.24, the  $d_E^2$  values depend very much on the different possibilities of  $x_{n-1}$ ,  $\mathbf{b}$  and  $\mathbf{b}'$  and the cardinality of  $D_2^L$  is very high. While this distance spectrum structure will complicate the error floor analysis, it will provide us with an explanation of the good BER properties of the resulting turbo system. We can finally consider also the error events of Hamming weight 2 and lengths  $L_p = p(L - 1) + 1 = p(Q + 1) + 1$ ,  $p > 1$ ,  $p \in \mathbb{N}$ , but their importance for the following developments is very limited due to their associated increasing squared Euclidean distance.

In the context of binary turbocodes and assuming ML decoding, whenever the minimum squared Euclidean distance of the code can be found, and taking into account the principle of rare error events [Proakis, 2001], the limit BER in the error floor region could be approximated in the case of the AWGN channel as [Benedetto et al., 1998]

$$P_{b_{\text{floor}}} \approx \frac{N_{\min} \bar{w}_{\min}}{2N} \operatorname{erfc} \left( \sqrt{\frac{d_{\min}^2}{4P} R \frac{E_b}{N_0}} \right), \quad (5.27)$$

where  $N_{\min}$  is the number of binary input error events which lead to output words whose distance is the minimum squared Euclidean distance,  $\bar{w}_{\min}$  is the mean value of the Hamming weight of the input binary error events leading to these output words,  $N$  is the size of the interleaver,  $d_{\min}^2$  is the minimum squared Euclidean distance,  $P$  is the power of the signal in the channel, and  $R$  is the overall code rate. For the binary turbocodes with antipodal signalling,  $d_{\min}^2 = 4d_{\text{free}}^2$  and  $P = 1$ , and so the bound of Eq. (5.27) is a function of the free distance of the parallel concatenated binary code. Recalling what was seen in Chapter 3 with respect to the bounding of the bit error probability, in the case of the PCCCM system the Eq. (5.27) would take the general form

$$P_{b_{\text{floor}}} \approx \beta_{\min} \sum_{\mathbf{x}, \mathbf{x}' | d_{\min}^2} P_e(\mathbf{x} \rightarrow \mathbf{x}' | \mathbf{x}), \quad (5.28)$$

where  $P_e(\mathbf{x} \rightarrow \mathbf{x}' | \mathbf{x})$  is the error event probability corresponding to error events with minimum squared Euclidean distance  $d_{\min}^2$  and which, in general, would depend on  $\mathbf{x}$  and  $\mathbf{x}'$ . The parameter  $\beta_{\min}$  is the bit enumerator factor and will be related to the same parameters as seen in the binary turbocode case. Note that the expression in Eq. (5.28) is general and could be applied to channels other than the AWGN one, just by redefining the meaning of minimum squared Euclidean distance (see Chapter 3), and applying the corresponding error event probability as a function of the channel distortion. In the AWGN channel, the error event probability takes the known form

$$P_e(\mathbf{x} \rightarrow \mathbf{x}' | \mathbf{x}) = \frac{1}{2} \operatorname{erfc} \left( \sqrt{\frac{d_E^2}{4P} R \frac{E_b}{N_0}} \right). \quad (5.29)$$

We have seen that the error event probability could be given as Eq. (3.29) in the ISI channel, as Eq. (3.49) for the fading channel with CSI, and as Eq. (3.66) for the fading channel without CSI. A more accurate bound than the one of Eq. (5.28) could be established using the union bound techniques, as is done in [Tullberg and Siegel, 2005a] for TCM, but this would require to compute the transfer functions of these encoders [Lin and Costello, Jr., 2004], which is almost an unfeasible task in general due to the fact that the UEP is not applicable. For the purposes of calculating the error floor, Eq. (5.28) will be enough.

There exist a number of methods to search for the minimum squared Euclidean distance of a parallel concatenated binary or non-binary code, mainly based upon the uniform interleaver assumption which makes the evaluation of the bit enumerator factor feasible regardless of the size of the interleaver  $N$  [Perez et al., 1996; Garelo et al., 2001; Schlegel and Pérez, 2004]. These methods also rely on the uniform error property, so that for them it is enough to search for the binary error events  $\mathbf{e}$  leading to the free (or minimum) distance. In our case, with the constituent CCM's not accomplishing the UEP (with the exception of BSM), the analysis could be very cumbersome, even assuming the uniform interleaver device to avoid considering the specific structure of the interleaver.

Nevertheless, taking into account what was said before about the error events with Hamming weight 2 for the constituent CCM's, there are some hints that we can employ to give a possible bit error probability bound in the error floor region based on the structure of the interleaver itself. We assume the S-random interleaver as mentioned in Subsection 5.2.1, so that we avoid the possible error events with Hamming weight 1 of the simple  $N_1 \times N_2$  block interleaver when the trellis is not terminated in neither of the constituent encoders [Hokfelt et al., 2001]. Even with trellis termination, though the  $N_1 \times N_2$  block interleaver could have a large spreading factor leading to a high minimum squared Euclidean distance, the multiplicities of the error events can be potentially very high due to its regularity [Schlegel and Pérez, 2004]. Regularity itself is one of the reasons we will see for the poor behaviour of the PCCCM system with BSM CCM's as constituent encoders.

With the S-random interleaver, it is clear that the error events with Hamming weight 2 that can occur simultaneously in both constituent encoders have to be those meeting the restrictions of the S-random algorithm (see Subsection 4.2.1). Let us denote as  $L_1$  the minimum loop length for which one of such error events can take place in the CCM. When  $S \leq L_1$ , there exists the possibility of having the minimal loop error event in both encoders, and so the absolute minimum squared Euclidean distance could be

$$d_{\min}^2 = 2 \cdot \min_{d_E^2} D_2^{L_1}. \quad (5.30)$$

To avoid this, we will always try to choose for  $S$  a value higher than  $L_1$ , because a weight 2 error event of length  $L_1 < S$  could be mapped at most as a weight 2 error event for the second encoder with length  $L_p > S$ ,  $p > 1$ , and so we can avoid the kind of undesirable minimum squared Euclidean distance events of Eq. (5.30). Note that the maximum possible value for  $S$  is limited by the size of the interleaver

$N$ , since  $S < \sqrt{N/2}$ . This also puts a limit on the value of  $Q$  once we have a pair  $S$  and  $N$ , because  $L_1 = Q + a$ ,  $a \in \mathbb{N}$ , and we require that  $L_1 < S$  to avoid low squared Euclidean distance weight 2 error events in the concatenated code.

In Section 5.5 we will use in many of the simulations S-random interleavers with  $S = 23$  for CCM's with  $Q = 5$ , so that, regardless of the kind of CCM,  $S > L_3$ , and so a Hamming weight 2 binary error event would have a possible minimum squared Euclidean distance higher than the sum of the minimum squared Euclidean distances for loops with  $L = L_1$  and loops with  $L = L_3$ , and therefore

$$d_{\min}^2 > 4 \cdot \min_{d_E^2} D_2^{L_1}. \quad (5.31)$$

In this case, we will verify that the minimum squared Euclidean distance is given by the association of two binary error events with Hamming weight 2, since no restriction is made in the S-random interleaver respecting the concatenation of error loops. If the structure of the permutation chosen semi-randomly allows it, we could have two loops of minimal length  $L_1$  with Hamming weight 2 starting at  $i$  and  $j$ , and ending respectively in  $i + L_1 - 1$  and  $j + L_1 - 1$ , mapped into indexes  $\pi(i)$ ,  $\pi(j)$ ,  $\pi(i + L_1 - 1)$  and  $\pi(j + L_1 - 1)$  and leading to two minimal length loops with Hamming weight 2 given by, for example

$$|\pi(i) - \pi(j + L_1 - 1)| = L_1 \quad |\pi(j) - \pi(i + L_1 - 1)| = L_1, \quad (5.32)$$

and any other possibility not violating the restrictions of the S-random algorithm. In case this possibility exists, the absolute minimum for the squared Euclidean distance will be

$$d_{\min}^2 = 4 \cdot \min_{d_E^2} D_2^{L_1}, \quad (5.33)$$

and the corresponding BER error floor could be approximated in the AWGN channel by

$$P_{b_{\text{floor}}} \approx \frac{w_{\min} N_4}{2N} \operatorname{erfc} \left( \sqrt{\frac{4 \cdot \min_{d_E^2} D_2^{L_1}}{4P} R \frac{E_b}{N_0}} \right), \quad (5.34)$$

where  $w_{\min} = 4$  is the Hamming weight of the input binary error event and  $N_4$  is the loop multiplicity, i.e., the total number of combinations of the kind of Eq. (5.32) allowed by the interleaver structure. This expression is specially useful when the UEP holds, as in the case of BSM, since  $D_2^{L_1}$  has only one element and the error events of the kind described with Hamming weight 4 will always have the same squared Euclidean distance. Conversely, in the case of the parallel concatenation of mBSM, TM or mTM CCM's, there could be a very low probability of having simultaneously 4 Hamming weight 2 binary errors with a squared Euclidean distance taking the absolute minimum value in  $D_2^{L_1}$ , specially for TM and mTM (see histograms 5.23, 5.22 and 5.24). This is a consequence of their non-UEP structure, and so the squared Euclidean distances depend on the specific value of the input words  $\mathbf{b}$  and  $\mathbf{c}$ . For this situation, a better approximation to the BER error floor can be given

following a combinatorial approach if we average over all the possible combinations with repetition of the elements of  $D_2^{L_1}$  taken 4 by 4. Note that this is only valid for mBSM and TM CCM's, since for these encoders the frequencies of the individual Hamming weight 2 binary error events for each of the elements in  $D_2^{L_1}$  are the same (see Figs. 5.22 and 5.23). If we denote the number of elements in  $D_2^{L_1}$  as  $t$ , the number of combinations to take into account is

$$\binom{t+4-1}{4} = \frac{(t+3)!}{(t-1)!4!} = \frac{(t+3)(t+2)(t+1)t}{24}. \quad (5.35)$$

With these definitions, the average bound for mBSM and TM in the AWGN channel could be given as

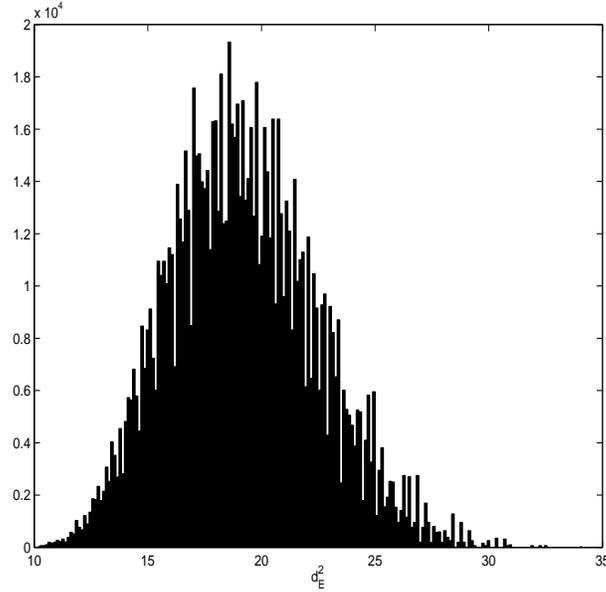
$$P_{b_{\text{floor}}} \approx \frac{w_{\min} N_4}{2N} \frac{24}{(t+3)(t+2)(t+1)t} \sum_{\substack{d_E^2 = d_1^2 + d_2^2 + d_3^2 + d_4^2 \\ d_i^2 \in D_2^{L_1}}} \text{erfc} \left( \sqrt{\frac{d_E^2}{4P} R \frac{E_b}{N_0}} \right), \quad (5.36)$$

where the distances  $d_i^2$  take values from  $D_2^{L_1}$  as combinations with repetition regardless of the order.  $w_{\min}$  and  $N_4$  have the meaning seen in Eq. (5.34). We see that the multiplicity of the absolute minimum of  $d_E^2$  is now decreased by  $24/((t+3)(t+2)(t+1)t)$  with respect to the concatenation of two BSM CCM's, and so it is to expect a better behaviour of the systems with mBSM or TM CCM's in the error floor region. This average with equal multiplicity for each combination of error events with weight 2 and minimal length is a good approximation if we can suppose that all possibilities for  $\mathbf{b}$  and  $\mathbf{c}$  are equiprobable at the input of each encoder, and this condition is met reasonably when the binary input sequence is *iid* and the interleaver is large enough to sufficiently uncouple  $\mathbf{b}$  and  $\mathbf{c}$ .

In the case of two mTM CCM's, the spectrum of squared Euclidean distances for the Hamming weight 2 binary error events prevents us from applying the same combinatorial approach. A possible tight average bound for the BER error floor could be given by taking the histogram of the distribution  $p(d_E^2)$  over  $d_E^2 = d_1^2 + d_2^2 + d_3^2 + d_4^2$ ,  $d_i^2 \in D_2^{L_1}$ , and numerically integrating [Abramowitz and Stegun, 1965]

$$P_{b_{\text{floor}}} \approx \frac{w_{\min} N_4}{2N} \int_{d_{E_{\min}}^2}^{d_{E_{\max}}^2} p(v) \cdot \text{erfc} \left( \sqrt{\frac{v}{4P} R \frac{E_b}{N_0}} \right) dv, \quad (5.37)$$

where we have replaced  $d_E^2$  by the dummy variable  $v$ . In Fig. 5.25, we can see the corresponding histogram of  $d_E^2$  for an mTM CCM with  $Q = 5$  and for the binary error events considered. Only for low values of  $Q$  we can compute easily such histogram. See that the histogram is approximately Gaussian and that the minimum and the mean values for  $d_E^2$  are high. This expanded and richer structure of the lowest Hamming weight input binary error events allows us to foresee a lower error floor with mTM CCM's than with the mBSM or TM CCM's.



**Figure 5.25.** Histogram of the possible output squared Euclidean distances when the binary error events consist on 2 individual binary input error events with Hamming weight 2 in the case of a concatenation of two mTM encoders with  $Q = 5$ .

These bounds also allow us to estimate the frame error probability in the error floor region  $P_{e_{\text{floor}}}$  through the relationship

$$P_{b_{\text{floor}}} \approx \frac{w_{\text{min}}}{N} P_{e_{\text{floor}}}, \quad (5.38)$$

where it has been taken into account that, for each frame on error, there will be  $w_{\text{min}}$  bit errors over a total of  $N$  information bits. This  $P_{e_{\text{floor}}}$  can also be seen as the average pairwise error probability  $E[P_e(\mathbf{x} \rightarrow \mathbf{x}'|\mathbf{x})]$  in the error floor region.

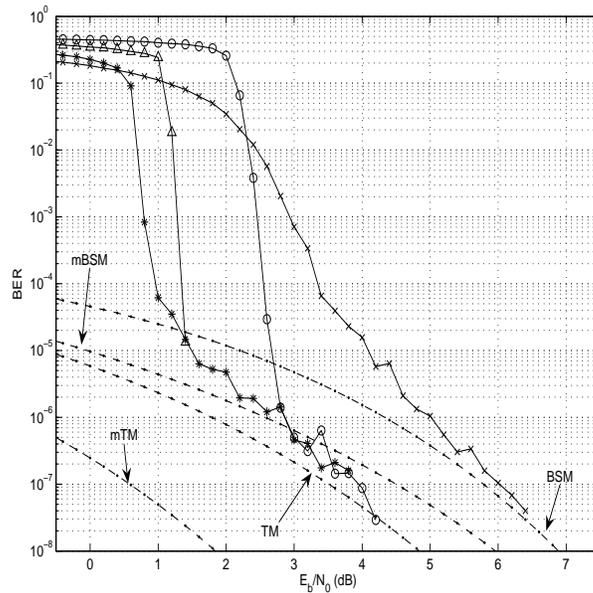
In the ISI channel, a possible bound based upon these principles would be extremely cumbersome to calculate, as the developments seen in Chapter 3 to calculate the error event probability of Eq. (3.26) are based on a combinatorial approach whose complexity would grow a lot when considering the PCCCM and the combination of several input binary error loops, apart from the fact that the task to find the events with minimum equivalent Euclidean distance for the new encoders is not itself an easy task (see Section 3.3). Recall also that now the samples from both constituent encoders are mingled in each  $r_k$ .

On the other hand, the evaluation of bounds in the fading channel, both with CSI and without CSI, is not so complicated given the error event probabilities of Eqs. (3.49) and (3.66), and if we assume that the kind of error events in PCCCM occurring in the error floor region in these channels is the same as in AWGN. The only change would be the need to calculate the set of possible sequences of  $(x_n - x'_n)^2$  or  $x_n^2(x_n - x'_n)^2$  for the error events, the corresponding multiplicities and then to average in the same way with the specific expressions of the error event probability.

We will not perform these calculations here and will limit ourselves to show the validity of the proposed error floor bounds in the AWGN channel.

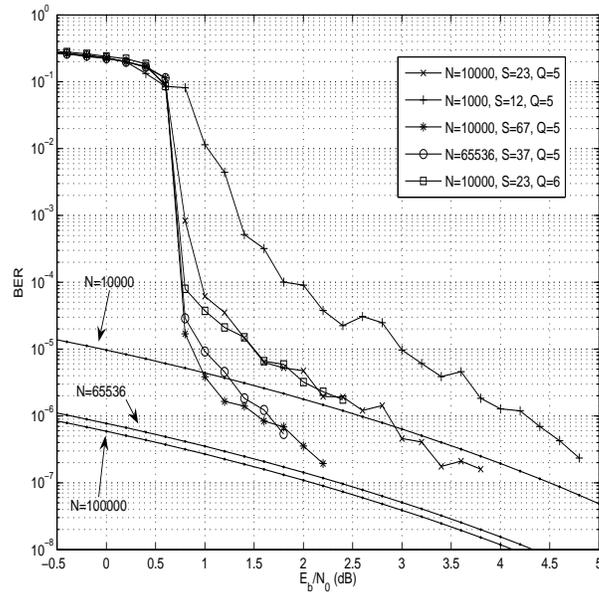
In the next section, we will introduce the simulation results for several PCCCM and we will see to what extent the predictions made through the EXIT chart analysis and the minimum squared distance analysis are close to the actual behaviour of the parallel concatenation of equal chaos coded modulation systems with a bit S-random interleaver.

## 5.5. Simulation results

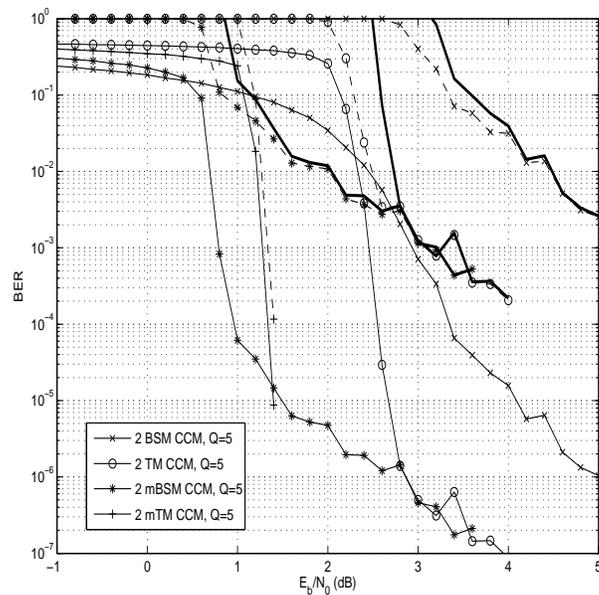


**Figure 5.26.** BER and corresponding error floor bounds for different parallel concatenation of two equal CCM's in the AWGN channel,  $Q = 5$ ,  $N = 10000$  and  $S = 23$ . Bounds are depicted with dash-dotted lines. 'x': PCCCM with two BSM CCM's. 'o': PCCCM with two TM CCM's. '\*': PCCCM with two mBSM CCM's. '△': PCCCM with two mTM CCM's.

In this section we present some simulation results which will be interpreted on the light of the previous analysis. In all the simulations, 20 decoding iterations were performed and the BER results were recorded after finding 20 frames on error. In Fig. 5.26, we have depicted the BER results for the parallel concatenation of two equal CCM's with S-random bit interleavers of length  $N = 10000$  and  $S = 23$ . All the CCM's have a quantization factor of  $Q = 5$ . The general behaviour for each of the encoders is as predicted through the EXIT charts, but with a shift in the  $E_b/N_0$  threshold for the BER waterfall region. For example, in the mBSM case the turbo cliff is not at  $E_b/N_0 = 0.0$  dB as suggested by Fig. 5.9, but well after  $E_b/N_0 = 0.5$  dB. The systems based on the TM, for which the thresholds were taken with  $E_b/N_0$  allowing narrower windows for the progression of the mutual information trajectory,



**Figure 5.27.** BER and corresponding error floor bounds for the parallel concatenation of two equal *mBSM CCM*'s in the AWGN channel, with different *Q* and *N* parameters. Bounds are depicted with dash-dotted lines.

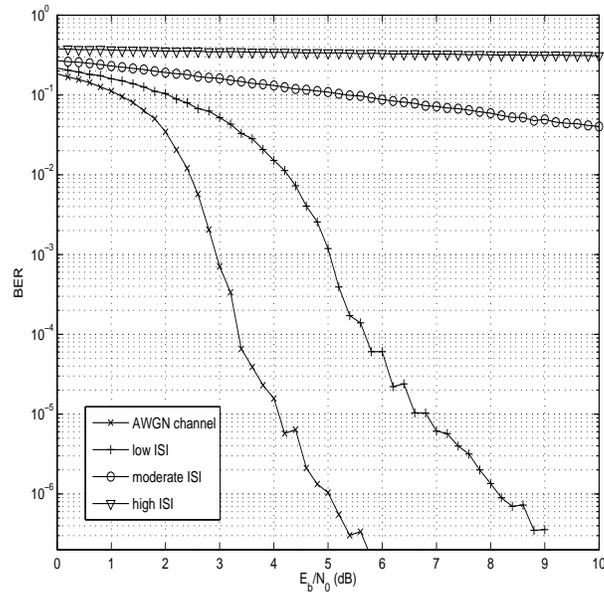


**Figure 5.28.** BER and FER for different parallel concatenation of two equal CCM's in the AWGN channel,  $Q = 5$ ,  $N = 10000$  and  $S = 23$ . BER: continuous line. FER: dashed line. Upper thick continuous lines correspond to the relation  $FER = (N/w_{\min})BER$  (not depicted for the *mTM* case).

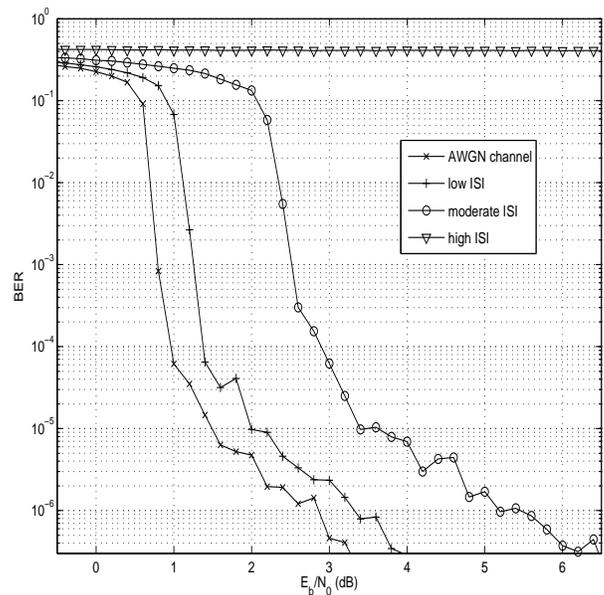
show a degradation of more than 1.0 dB with respect to the expected value. Note that what we foresaw about the concatenation of BSM CCM's is true: there is no real turbo cliff and the BER proceeds smoothly from the high BER region to the BER error floor.

In Fig. 5.26 we also show the bounds for the BER at the error floor region calculated according to the principles and expressions seen in Section 5.4. Though they only take into account the kind of error events mentioned there, they are reasonably tight. In the case of BSM and mBSM, the interleavers employed allowed a total of  $N_4 = 5$  possible composition of pairs of Hamming weight 2 binary error events with  $L = L_1 = Q + 1$ , and the interleavers of the TM and mTM cases allowed a total of  $N_4 = 4$  of such compound events. It is remarkable the fact that the theoretical error floor, together with the error floor shown by the simulations, decreases as the regularity in the spectrum of the squared Euclidean distances associated to the individual Hamming weight 2 binary error events decreases. In fact, the mTM case, which has the most complex squared Euclidean distance spectrum as seen in Fig. 5.24, exhibits a behaviour very close to the serial concatenated case, and even the simulations have not been able to reach the error floor region and validate the proposed value. The rest of cases agree well with the principles of parallel concatenation [Schlegel and Pérez, 2004]: a general good behaviour for low  $E_b/N_0$ , but a relatively high BER error floor for  $E_b/N_0 \rightarrow \infty$ .

In Fig. 5.27 we show the BER and the error floor bounds for the mBSM PCCCM with different values of  $N$ ,  $S$  and  $Q$ . The difference between  $Q = 5$  and  $Q = 6$  is small, and only seems to affect the turbo cliff, determining a slightly steeper slope for the  $Q = 6$  case. Nevertheless, the error floor is the same: though for  $Q = 6$  there are some important changes in the binary error events (the minimal loop length is  $L_1 = 7$  and the corresponding Hamming distance is  $d_H(\mathbf{x}, \mathbf{x}') = 6$ ), the values of the squared Euclidean distances are practically the same, since the changes in such values are small as  $Q \rightarrow \infty$  when  $Q > 4$ . This makes it clear that the dynamics of the map, which determines the spectrum of the squared Euclidean distances between pairs of chaotic sequences, is a more determining factor in the error floor region than the ad-hoc quantization level  $Q$ . On the other hand, when we change the value of  $N$ , we can appreciate in Fig. 5.27 that the interleaver gain in the error floor region changes as  $N^{-1}$  as expected for any parallel concatenated system with interleavers [Schlegel and Pérez, 2004]. There is not even a noticeable improvement in the  $E_b/N_0$  threshold for the waterfall region with growing  $N$  with respect to the case  $N = 10000$ . Only when  $N < 10000$  (see the  $N = 1000$  case), the situation is clearly worse and the behaviour starts resembling the pure BSM case. In fact, with  $N = 1000$  the interleaver requires a lower value for  $S$ , and, since it takes now the value  $S = 12$ , the dominant error events in the error floor region are those with overall Hamming weight 2 (instead of 4) and given by one individual binary error loop with  $L = L_1$  in one encoder and  $L = 3(L_1 - 1) + 1 = 16 > S$  on the other. These error events have associated squared Euclidean distances lower than the squared Euclidean distances for the compound Hamming weight 4 error events of the cases with  $S > 3(L_1 - 1) + 1$ .

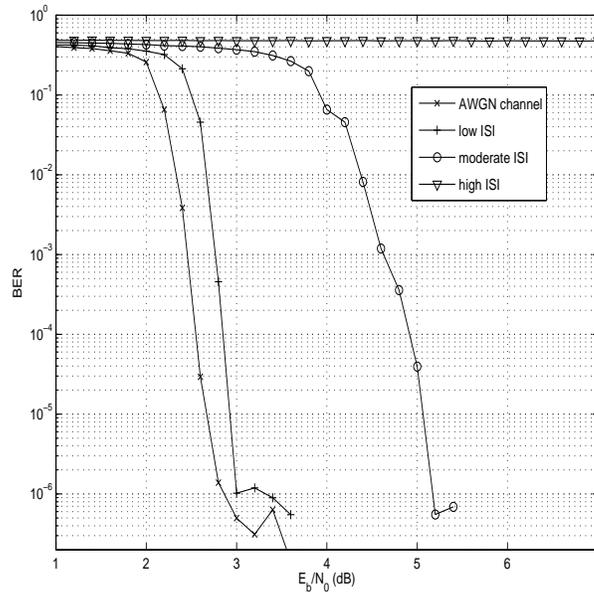


**Figure 5.29.** BER for two BSM CCM's in the ISI channel,  $Q = 5$ ,  $N = 10000$  and  $S = 23$ .

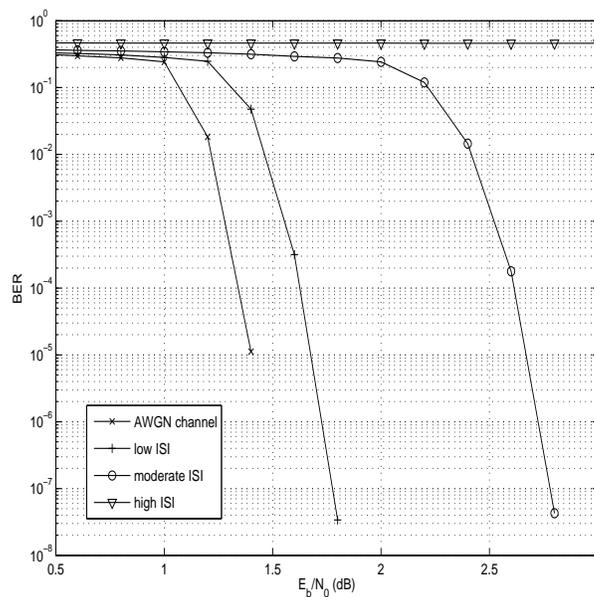


**Figure 5.30.** BER for two mBSM CCM's in the ISI channel,  $Q = 5$ ,  $N = 10000$  and  $S = 23$ .

The oscillation seen in the tails of the BER for the mBSM and TM cases of the Figs. 5.26 and 5.27 is due to the fact that 20 frames on error are not enough to accurately estimate the BER in the transition region between the turbo cliff and the error floor, since, for the values of  $E_b/N_0$  involved, the SISO decoders are on a somewhat unstable point where the erroneous frames have either only 4 bit errors



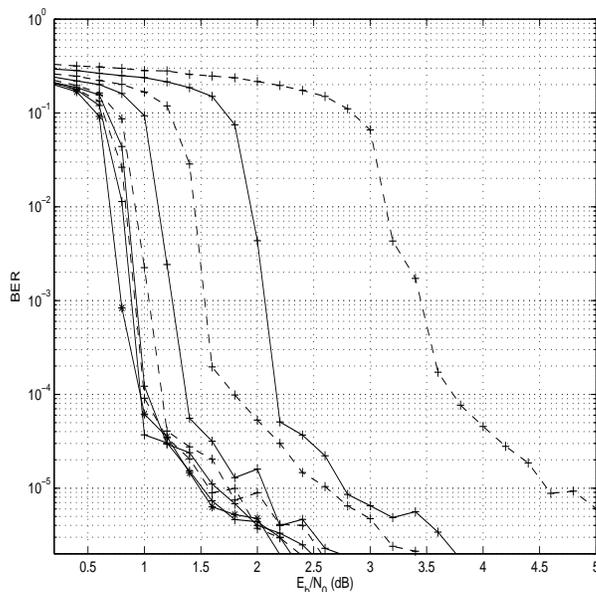
**Figure 5.31.** BER for two TM CCM's in the ISI channel,  $Q = 5$ ,  $N = 10000$  and  $S = 23$ .



**Figure 5.32.** BER for two mTM CCM's in the ISI channel,  $Q = 5$ ,  $N = 10000$  and  $S = 23$ .

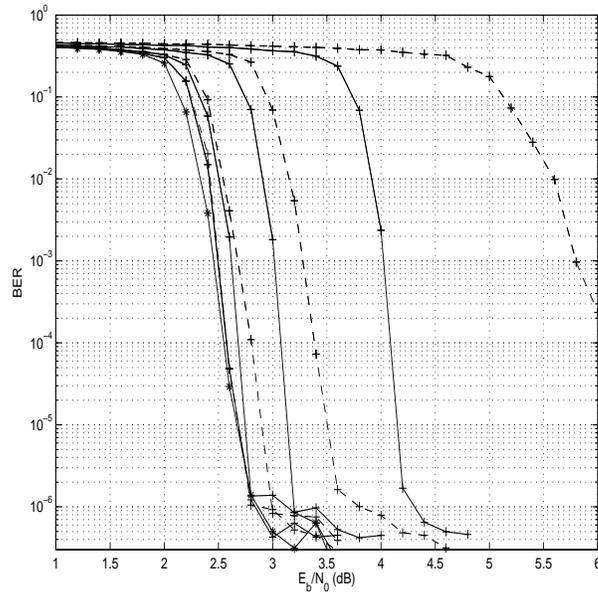
(the kind of error events seen), either error events with different number of bit errors and squared Euclidean distances not so near the minimum values. Therefore, the correct error rate could only be well estimated with a larger number of frames with errors. This is related to the fact that some of the  $w_{\min} = 4$  error events of Section 5.4, depending on the values of  $x_{n-1}$  and  $b_n, \dots, b_{n+L-1}$  for each error loop

with  $L = L_1$ , can sometimes have larger distances than other error events with, for example,  $w_{\min} = 2$  and loop lengths  $L < S$  for one decoder and  $L > S$  on the other. Nevertheless, as we can see in Fig. 5.28, the BER and the FER in the error floor region are well related through  $\text{BER} = (w_{\min}/N)\text{FER}$ , with  $w_{\min} = 4$  and  $N = 10000$ . For the mTM case, unfortunately, we do not have any information about the error floor region and we cannot compare, though we can foresee a very low value for the BER in such region.

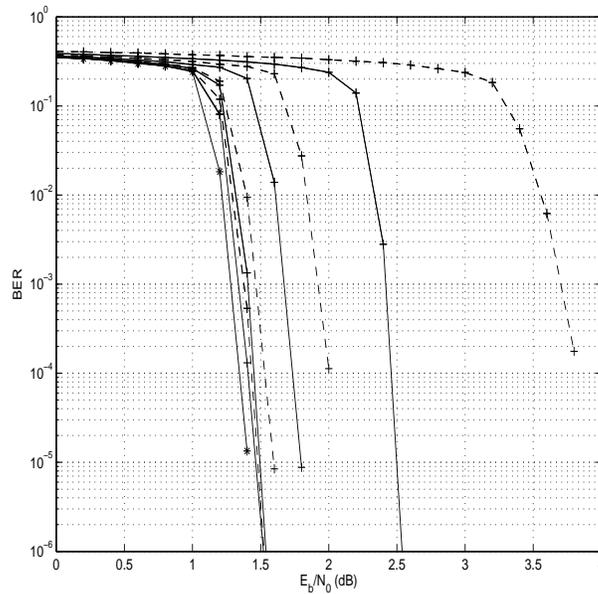


**Figure 5.33.** BER for two mBSM CCM's in the fading channel,  $Q = 5$ ,  $N = 10000$  and  $S = 23$ . Line with '\*': same system in the AWGN channel. Continuous lines with '+': cases with CSI. Dashed lines with '+': cases without CSI. From right to left:  $K = 0, 5, 20, 50$ .

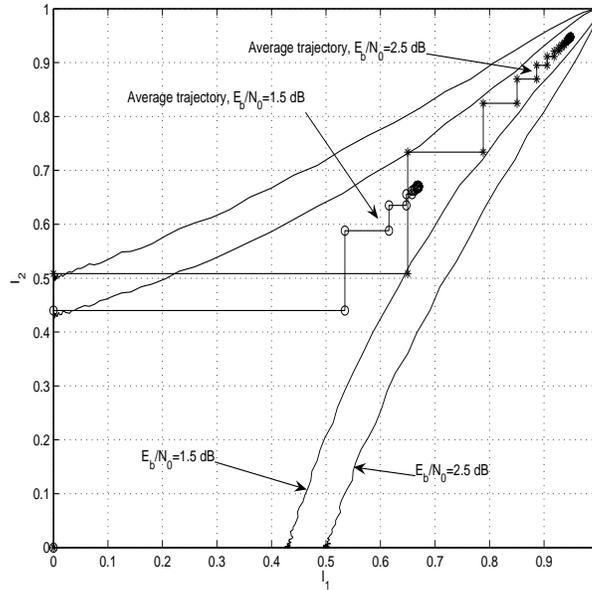
Following what was done in previous chapters, we have also evaluated the BER in dispersive channels. The results in the case of the ISI channel are shown in Figs. 5.29, 5.30, 5.31 and 5.32. We saw in Fig. 5.26 that the BSM PCCCM was a poor performing parallel concatenated system, and in the ISI channel it does not do better. In the case of low ISI, there is a loss of about 3.0 dB, and the behaviour with moderate ISI is almost as catastrophic as with high ISI. This could seem surprising, since, according to the results of Figs. 3.7 and 3.8 seen in Chapter 3 for an individual BSM CCM, the losses with respect to the AWGN case were around 1.0 dB with low ISI and 4.0 dB with moderate ISI for a BER of  $10^{-7}$ , and the overall results offered some degree of protection against these levels of ISI as compared with the uncoded BPSK case. Moreover, the simulations of PCCCM show that the dominant error events for  $E_b/N_0 \rightarrow \infty$  in the case of low ISI are the same as for BSM in the pure AWGN channel, but the system behaves worse because the equivalent squared Euclidean distance spectrum related to these error events (and so to the error event probabilities of Eq. (3.26)) is dramatically changed. For moderate ISI, the filtering



**Figure 5.34.** BER for two TM CCM's in the fading channel,  $Q = 5$ ,  $N = 10000$  and  $S = 23$ . Line with '\*': same system in the AWGN channel. Continuous lines with '+': cases with CSI. Dashed lines with '+': cases without CSI. From right to left:  $K = 0, 5, 20, 50$ .



**Figure 5.35.** BER for two mTM CCM's in the fading channel,  $Q = 5$ ,  $N = 10000$  and  $S = 23$ . Line with '\*': same system in the AWGN channel. Continuous lines with '+': cases with CSI. Dashed lines with '+': cases without CSI. From right to left:  $K = 0, 5, 20, 50$ .

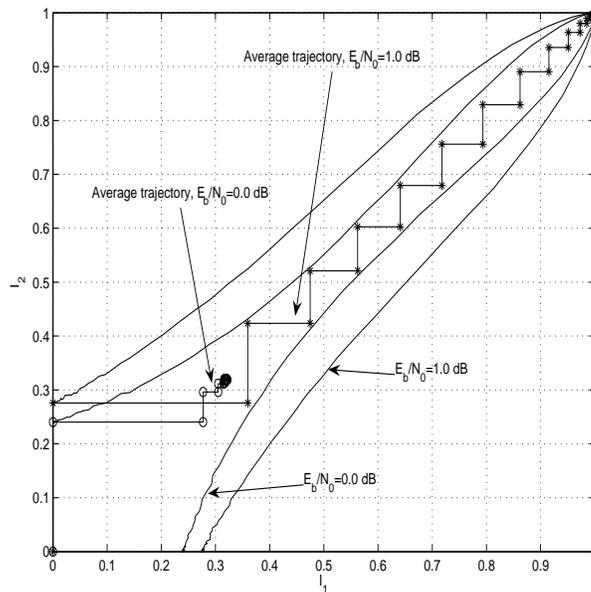


**Figure 5.36.** EXIT chart for the parallel concatenation of two BSM CCM's with  $Q = 5$  and average trajectories for an  $N = 10000$ ,  $S = 23$  S-random interleaver.

of the data establishes such a level of correlation among the samples from one encoder and the samples from the other that the turbo principle fails to work, attained in one of its main pillars: the relative independence between the samples of both encoders. This is the reason why these results have nothing to do with what could be deduced from the EXIT chart of Fig. 5.12 (i.e., no degradation with low ISI and 0.5 dB degradation with moderate ISI), since in this chart we did not take into account the extra level of correlation introduced by the FIR filter.

According to all this, the rest of the cases, though not so badly affected by ISI (excepting the high ISI case, which always will require equalization), show mismatches with respect to the predictions of the EXIT charts more important than those simply due to the failure of the Gaussian assumption for density evolution, or to an unperfect relative uncorrelation given by the finite size interleaver. This penalty will be far more remarkable for the moderate ISI case. In fact, the EXIT charts of Figs. 5.13, 5.14 and 5.15 pointed towards a loss of less than 0.5 dB for low ISI, and less than 1.0 dB for moderate ISI. According to Figs. 5.30, 5.31 and 5.32, with low ISI the predictions approximately hold because the distortion is kept very low<sup>7</sup>. In the moderate ISI case, with the exception of the mTM PCCCM, the increasing in the  $E_b/N_0$  threshold for the turbo cliff are far higher than 1.0 dB. The better behaviour of the mTM case could be related to its non-regularity, which has already proved useful in the error floor BER for the AWGN channel. The lacking of uniformity between the mapping of binary messages into encoded chaotic codewords seems to provide an extra level of immunity against frequency selective channel im-

<sup>7</sup>Note nevertheless how the low ISI channel left almost unaffected the BSM SCCCM system of Chapter 4 (see Fig. 4.14).

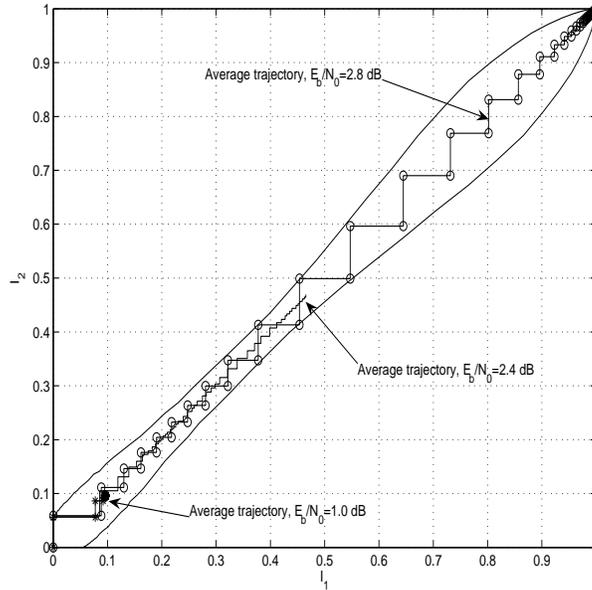


**Figure 5.37.** EXIT chart for the parallel concatenation of two *mBSM* CCM's with  $Q = 5$  and average trajectories for an  $N = 10000$ ,  $S = 23$  *S*-random interleaver.

pairment. This is also coherent with the fact that the PCCCM with constituent BSM CCM's, being almost linear, leads to the worst behaviour<sup>8</sup>.

Proceeding with the dispersive channels, we represent in Figs. 5.33, 5.34 and 5.35 the behaviour of the proposed systems under Rician and Rayleigh frequency-non selective fading, with and without CSI. We have dropped the BSM case because, though it has served us to gain more insight into the BER results, it adds no relevant additional information now. The BER results follow more closely now the predictions of the EXIT charts in Figs. 5.16, 5.17, 5.18, 5.19, 5.20 and 5.21, with the exception of the usual differences in the  $E_b/N_0$  thresholds due to the unaccuracy of the Gaussian assumption and of the limited capacities of the interleaver to uncorrelate the binary data and the chaotic samples of the different CCM's. We see that, as expected, the differences between the CSI and the non-CSI cases tend to vanish fast as  $K$  grows, and it can be appreciated how the cases with  $K = 20$  and  $K = 50$  show marginal losses with respect to the BER in AWGN. For the worst and most interesting case, i.e., Rayleigh fading ( $K = 0$ ), the differences in the  $E_b/N_0$  threshold for the waterfall region is 1.0 dB for Rayleigh fading with CSI in the BSM and *mTM* PCCCM cases, and 1.5 dB for the *TM* PCCCM one. This is in good agreement with Figs. 5.16, 5.18 and 5.20. In the Rayleigh case without CSI, the BER takes additionally some tenths of dB to reach the turbo cliff with respect to what can be seen in the EXIT charts. Again, the best result is given by

<sup>8</sup>According to this, in the case of SCCCM with a BSM CCM, the reason for the better behaviour under ISI could be related to the fact that the convolutional code together with the interleaver adds the required level of non-uniformity.

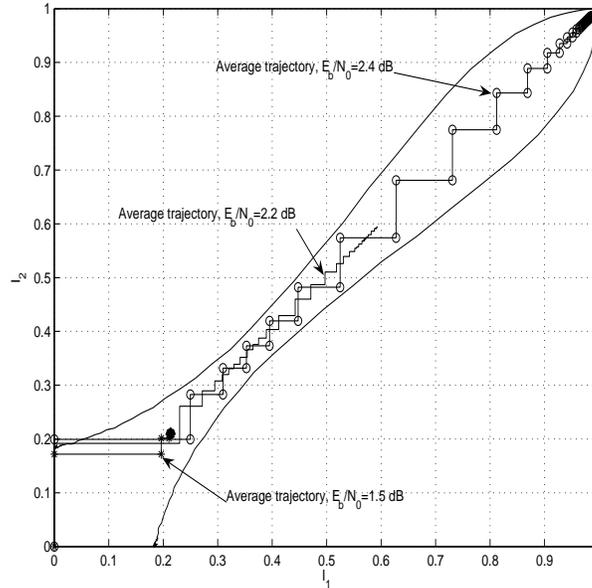


**Figure 5.38.** EXIT chart for the parallel concatenation of two TM CCM's with  $Q = 5$  when  $E_b/N_0 = 1.0$  dB in the low ISI channel, and three average trajectories for an  $N = 10000$ ,  $S = 23$  S-random interleaver.

the mTM PCCCM, where the overall loss in the worst situation (Rayleigh fading without CSI) is in practice limited to a bit more than 2.0 dB. We can compare this with our SCCCM example, where the loss was a little higher than 3.0 dB (see Fig. 4.13).

Note that the previous figures illustrate that we can get BER results on any of the channels considered that are comparable with the results attainable with turbocodes or TTCM systems of not very different complexity and rate [Benedetto and Montorsi, 1996; Divsalar and Pollara, 1997; Schlegel and Pérez, 2004].

We finally compare the trajectories of the mutual information during the iterative decoding process when using a finite length specific interleaver with respect to the theoretical transfer functions of Section 5.3. In Fig. 5.36, we can see how it is true that the mutual information, and so the BER, proceeds smoothly to the error floor limit with the BSM CCM's. Even when the road seems wide open to the (1, 1) point for  $E_b/N_0 = 2.5$  dB, 1.0 dB above the proposed theoretical threshold, the average mutual information trajectory gets stuck at a point where the BER is still above  $10^{-3}$  (see Fig. 5.26). For mBSM (Fig. 5.37), the situation is better and at 1.0 dB above the theoretical threshold we can consider to have reached the error floor region. In Figs. 5.38 and 5.39 we have plotted the EXIT charts and the average trajectories for two specific examples of PCCCM in the dispersive channels. Fig. 5.38 shows that, for low ISI, the convergence is in fact reached almost 2.0 dB over the expected value. A similar result can be observed in Fig. 5.39 for the fading channel with  $K = 0$  and perfect CSI, though here the difference with the expected value is only 1.0 dB. This stresses the fact that the mismatch will always be higher



**Figure 5.39.** EXIT chart for the parallel concatenation of two *mTM CCM*'s with  $Q = 5$  when  $E_b/N_0 = 1.5$  dB in the fading channel with CSI and  $K = 0$ , and three average trajectories for an  $N = 10000$ ,  $S = 23$  *S*-random interleaver.

in the ISI channel due to the problems already mentioned.

Note how in all the figures the only value of the average trajectory approaching the theoretical one corresponds to the first decoding step through the first SISO module, i.e., when there is no input mutual information from the other decoder. In the rest of steps, it is evident that the EXIT chart assumptions for such input mutual information only hold approximately and we deviate progressively from what could be expected. However, we can say that the EXIT charts themselves, taken with the needed reserve, can be usefully managed to analyze the PCCCM system.

## 5.6. Concluding remarks

We have continued here with the building of concatenated systems based on CCM's with the aim of improving their performance by increasing the system dimensionality, but without going out from standard digital communications theory. Therefore, we have extended the concept of turbo coding and proposed the parallel concatenation of two CCM's linked by means of a bit interleaver. While for the purposes of illuminating the possibilities of the described systems, the BSM CCM has served us well in Chapters 3 and 4, we have proposed here other CCM encoders based on the same principles, because the PCCCM built with BSM CCM's was a firm candidate for bad performance. Apart from adapting the tent map to the CCM system based on small perturbations, we have incorporated the class of CCM systems with switched maps and small perturbations control [Kozic et al., 2003a].

Again, the EXIT charts device has helped us to get useful hints about the per-

formance of the different kinds of PCCCM designed with two equal CCM's, though for the ISI channel the results are not very reliable since the assumption of uncorrelation between the data from both encoders is far from met. Besides, an analysis of the error events of the constituent CCM's has allowed us to draw a bound for the BER in the error floor region. This has improved our knowledge respecting how we can arrange the interleaver structure for lowering the error floor.

Most important, apart from verifying again the relative independence of the results from the chaotic signal quantization level, we have seen that the best performance, at least in the error floor region, is to be achieved by the CCM's that are furthest from meeting the uniform error property. In PCCCM, the regularity in the CCM behaviour seems to be a severe drawback, and this explains the uselessness of the BSM CCM. Other surprise has been the finding that the TM CCM, which individually yields a very poor BER, can improve greatly its performance when parallel concatenated. This was well explained again by the lack of regularity in the mapping between the most probable input binary error events and the corresponding output squared Euclidean distances, thanks to the complex interaction between the CCM's and the bit interleaver.

On the other hand, the behaviour of PCCCM, while comparable with TTCM in the AWGN channel, has shown a high degree of robustness in the flat fading channel. However, the BER in the ISI channel claims for the need of express equalization even with low-moderate ISI, since PCCCM does not tolerate very well the correlation between the samples of both encoders arising from frequency selective distortion.

## Chapter 6

# Conclusions

We enumerate in what follows the general conclusions of this PhD thesis.

1. We have verified that, when encoding initial conditions, the design of a communications system based on a one-dimensional piecewise nonlinear map using as design criterion the invariant density of the chaotic data leads to poor results. In fact, the invariant density can give reason of the general behaviour of the bit error rate, but it cannot explain the details, so that the map dynamics remain the main factor to be taken into account. Moreover, we have seen that in the AWGN channel, piecewise nonlinear maps are worse performing than their piecewise linear counterpart for the kind of symmetry studied.
2. When the decoding of the chaotic sequence relies on initial condition estimation, we have seen that the performance is dominated by the reference samples that lie outside the definition interval after the addition of the noise. The best way to overcome this is to perform sequence estimation, with MAP or ML algorithms, or by using recursively the decoding based on initial condition estimation. The reason for this was that, with such methods, the redundancy implicitly present in the whole chaotic sequence is conveniently propagated and can thus be fully exploited. It is a known fact that decoding *locally* leads to the appearance of a threshold effect.
3. The poor results of the encoding and decoding systems based on both piecewise linear and piecewise nonlinear maps lead us to consider the kind of chaos coded modulations that, based on piecewise linear maps driven by small perturbations, had produced already more efficient chaos-based communications systems. Nevertheless, we limited ourselves initially to a bad performing case in the AWGN channel in order to show that the coded properties of the chaos-based system can be usefully exploited in dispersive channels. In fact, after proposing a MAP decoding module suited to the small perturbations setup, we have shown that the effects of frequency selective impairment (intersymbol interference) and frequency non-selective time-varying impairment (flat fading) can be attenuated with such a poor performing chaos coded modulation.

4. To build good performing encoders in non-dispersive (AWGN) or dispersive environments, one can resort to the increasing of the redundancy provided when the system rate is  $R < 1$ . To do this, we have exploited the analogy that chaos coded modulated systems provide with convolutional encoders or trellis coded modulations, and considered the possibility of building serially and parallel concatenated systems. In both cases, the objective was to get a more complex mapping between the input binary message and the output chaotic samples as a way to effectively increase the robustness and the available redundancy in the system.
5. The serial concatenation with a convolutional encoder as outer encoder, a binary interleaver and a chaos coded modulation as inner encoder resulted in a system analogous to a serially concatenated trellis coded modulation system. We have shown that practical concatenated encoders and iterative decoders can be employed with the chaos-based setup, and that the results can be analyzed with standard tools borrowed from the digital communications field. The convergence and bit error analysis, together with the simulation results, have shown that the final performance is greatly enhanced and that these new chaos-based systems can compete with standard serially concatenated systems. In fact, the results were specially promising in channels with intersymbol interference.
6. The parallel concatenation of two chaos coded modulations with a binary interleaver resulted in a system analogous to a turbo trellis coded modulation system. We described again the possibility to get practical concatenated encoders and iterative decoders, and showed that the analysis of the system is possible with known tools taken from digital communications as well. The results show once again that we can design chaos-based systems competitive with their non-chaotic counterparts. Moreover, the parallel concatenation of chaos coded modulations has provided us with very good results in flat fading channels.
7. The analysis of the parallel concatenated systems has shown that their best results are associated with encoding setups that are far from meeting the uniform error property. These systems exhibit a complicated mapping between input binary error events and output squared Euclidean distance. This can be seen as an equivalent instance of what happens in the serially concatenated system thanks to the addition of the channel encoder and the binary interleaver. These facts proved that the conjecture about the need for a higher complexity in the relationship between input binary message and output chaotic samples is in fact true, and that it can be implemented without special or totally new developments.
8. The examples analyzed have shown how the growing analogy between digital communications systems and chaos coded modulations can be managed to

design and evaluate good chaos-based communications systems. This analogy allowed the use of devices like the EXIT charts, the BIOS channel assumption, the evaluation of bounds based on input binary error events related to output chaotic error loops, and so on. This is a most important achievement, since in this way the evaluation and design tasks of the related chaos-based systems can be addressed in known and controllable ways. Nevertheless, we have found out that such devices are not always fitted to the chaos-based systems without some mismatch because of their special characteristics, and this points towards the need of more accurate developments. On the other hand, all the examples have shown that the discretization of the chaotic sequence seems to have a limited effect in the properties of the system. They have proved to be more related to the dynamical properties of the underlying map than to the value of the quantization factor.

As a final conclusion, we can say that we have demonstrated how, following a recently opened path that conveniently links digital and chaos-based communications, *it is possible to build efficient chaos-based systems both in non-dispersive and dispersive channels in a pretty straightforward way.* Now that we can have chaotic signals in the channel without the penalty of a severe degradation in performance, we foresee the real starting of practical chaos-based communications.



# Bibliography

- Abel, A. and Schwarz, W., 'Chaos Communications - Principles, Schemes and System Analysis', Proc. IEEE **90**, 691 (2002).
- Abramowitz, M. and Stegun, I. A., 'Handbook of Mathematical Functions with Formulas, Graphs and Mathematical Tables', Dover Publications, Inc., New York (1965).
- Altunbas, I. and Narayanan, K. R., 'Serial Concatenated Coding Scheme with Minimum Shift Keying', Electron. Lett. **37**, 1393 (2001).
- Anderson, J. B. and Svensson, A., 'Coded Modulation Systems', Kluwer Academic / Plenum Publishers, New York (2003).
- Andersson, H., 'Error-Correcting Codes Based on Chaotic Dynamical Systems', PhD thesis, Linköping University, Linköping, Sweden (1998).
- Argüello, F., Bugallo, M., and Amor, M., 'Multi-User Receivers for Spread Spectrum Communications Based on Chaotic Sequences', Int. J. Bifurcation Chaos **12**, 847 (2002).
- Argüello, F., Bugallo, M., and López, J., 'Blind Multiuser Receivers for Chaos Based Communications', Int. J. Bifurcation Chaos **13**, 2353 (2003).
- Avudainayagam, A., Shea, J. M., and Roongta, A., 'On Approximating the Density Function of Reliabilities of the Max-Log-MAP Decoder', in *Proceedings of the International Conference on Communications Systems and Applications (IASTED) 2004*, Banff, Canada (2004).
- Babich, F., Montorsi, G., and Vatta, F., 'Performance Bounds of Continuous and Blockwise Decoded Turbo Codes in Rician Fading Channel', Electron. Lett. **34**, 1646 (1998).
- Bahl, L. R., Cocke, J., Jelinek, F., and Raviv, J., 'Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate', IEEE Trans. Inform. Theory **20**, 284 (1974).
- Baptista, M. S. and López, L., 'Information Transfer in Chaos-based Communication', Phys. Rev. E **65**, 055201 (2002).
- Baptista, M. S., Macau, E. E., and Grebogi, C., 'Conditions for Efficient Chaos-Based Communication', CHAOS **13**, 145 (2003).
- Baranovsky, A. and Daems, D., 'Design of One-Dimensional Chaotic Maps with Prescribed Statistical Properties', Int. J. Bifurcation Chaos **5**, 1585 (1995).
- Benedetto, S., Divsalar, D., Montorsi, G., and Pollara, F., 'A Soft-Input Soft-Output Maximum a Posteriori (MAP) Module to Decode Parallel and Serial Concatenated Codes', Technical Report 42-127, Jet Propulsion Laboratory, California

- Institute of Technology (1996).
- Benedetto, S., Divsalar, D., Montorsi, G., and Pollara, F., ‘A Soft-Input Soft-Output APP Module for Iterative Decoding of Concatenated Codes’, *IEEE Commun. Lett.* **1**, 22 (1997).
- Benedetto, S., Divsalar, D., Montorsi, G., and Pollara, F., ‘Serial Concatenation of Interleaved Codes: Performance Analysis, Design and Iterative Decoding’, *IEEE Trans. Inform. Theory* **44**, 909 (1998).
- Benedetto, S., Marsan, M. A., Albertengo, G., and Giachin, E., ‘Combined Coding and Modulation: Theory and Applications’, *IEEE Trans. Inform. Theory* **34**, 223 (1988).
- Benedetto, S. and Montorsi, G., ‘Unveiling Turbo Codes: Some Results on Parallel Concatenated Coding Schemes’, *IEEE Trans. Inform. Theory* **42**, 409 (1996).
- Berrou, C., Glavieux, A., and Thitimajshima, P., ‘Near Shannon Limit Error-Correcting Coding and Decoding: Turbo-Codes’, in *Proceedings of the International Conference on Communications*, Vol. 2, pp 1064–1070, Geneva, Switzerland (1993).
- Biglieri, E., Caire, G., and Taricco, G., ‘Approximating the Pairwise Error Probability for Fading Channels’, *Electron. Lett.* **31**, 1625 (1995).
- Biglieri, E. and McLane, P. J., ‘Uniform Distance and Error Properties of TCM Schemes’, *IEEE Trans. Commun.* **39**, 41 (1991).
- Biglieri, E., Proakis, J., and Shamai, S., ‘Fading Channels: Information-Theoretic and Communications Aspects’, *IEEE Trans. Inform. Theory* **44**, 2619 (1998).
- Blackledge, J. M., ‘Cryptanalysis: Chaos and Fractals’, Horwood Publishing Ltd., London (2000).
- Bollt, E., ‘Review of Chaos Communication by Feedback Control of Symbolic Dynamics’, *Int. J. Bifurcation Chaos* **13**, 269 (2003).
- Bollt, E., Lai, Y.-C., and Grebogi, C., ‘Coding, Channel Capacity, and Noise Resistance in Communicating with Chaos’, *Phys. Rev. Lett.* **79**, 3787 (1997).
- Caire, G., Taricco, G., and Biglieri, E., ‘Bit-Interleaved Coded Modulation’, *IEEE Trans. Inform. Theory* **44**, 927 (1998).
- Callegari, S., Rovatti, R., and Setti, G., ‘Chaotic Modulations can Outperform Random ones in Electromagnetic Interference Reduction Tasks’, *Electron. Lett.* **38**, 543 (2002).
- Carlisle, C. J., Taylor, D. P., Shafi, M., and Kennedy, W. K., ‘Performance Bounds for Trellis-Coded Modulation on Time-Dispersive Channels’, *IEEE Trans. Commun.* **42**, 2534 (1994).
- Chen, B. and Wornell, G. W., ‘Analog Error-Correcting Codes Based on Chaotic Dynamical Systems’, *IEEE Trans. Commun.* **46**, 881 (1998).
- Chen, H. and Haimovich, A., ‘EXIT Charts for Turbo Trellis-Coded Modulation’, *IEEE Commun. Lett.* **8**, 668 (2004).
- Ciftci, M. and Williams, D., ‘Optimal Estimation and Sequential Channel Equalization Algorithms for Chaotic Communications Systems’, *EURASIP Journal on Applied Signal Processing* **4**, 249 (2001).
- Cover, T. M. and Thomas, J. A., ‘Elements of Information Theory’, John Wiley &

- Sons, Inc., New Jersey (2006).
- Divsalar, D., Dolinar, S., and Pollara, F., ‘Transfer Function Bounds on the Performance of Turbo Codes’, Technical Report 42-122, Jet Propulsion Laboratory, California Institute of Technology (1995).
- Divsalar, D. and Pollara, F., ‘Turbo Codes for PCS Applications’, in *Proceedings of the International Conference on Communications*, Vol. 1, pp 54–59, Seattle, USA (1995).
- Divsalar, D. and Pollara, F., ‘Turbo Trellis Coded Modulation with Iterative Decoding for Mobile Satellite Communications’, in *Proceedings of the International Mobile Satellite Conference (IMSC) 1997*, Pasadena, USA (1997).
- Dmitriev, A. S., Hasler, M., Panas, A. I., and Zakharchenko, K. V., ‘Basic Principles of Direct Chaotic Communications’, *Nonlin. Pheno. In Complex Sys.* **6**, 488 (2003).
- Dolinar, S. and Divsalar, D., ‘Weight Distributions for Turbo Codes Using Random and Nonrandom Permutations’, Technical Report 42-122, Jet Propulsion Laboratory, California Institute of Technology (1995).
- Escribano, F. J., López, L., and Sanjuán, M. A. F., ‘Iteratively Decoding Chaos Encoded Binary Signals’, in *Proceedings of the Eighth IEEE International Symposium on Signal Processing and Its Applications (ISSPA) 2005*, Vol. 1, pp 275–278, Sydney, Australia (2005).
- Escribano, F. J., López, L., and Sanjuán, M. A. F., ‘Evaluation of Channel Coding and Decoding Algorithms Using Discrete Chaotic Maps’, *CHAOS* **16**, 013103 (2006a).
- Escribano, F. J., López, L., and Sanjuán, M. A. F., ‘Exploiting Symbolic Dynamics in Chaos Coded Communications with *Maximum a Posteriori* Algorithm’, *Electron. Lett.* **42**, 984 (2006b).
- Escribano, F. J., López, L., and Sanjuán, M. A. F., ‘Serial Concatenation of Channel and Chaotic Encoders’, in *Proceedings of the Fourteenth International Workshop on Nonlinear Dynamics of Electronic Systems (NDES) 2006*, pp 30–33, Dijon, France (2006c).
- Firmanto, W., Vucetic, B., Yuan, J., and Chen, Z., ‘Space-Time Turbo Trellis Coded Modulation for Wireless Data Communications’, *EURASIP Journal on Applied Signal Processing* **5**, 459 (2002).
- Fröhlich, J., Novikov, S. P., and Ruelle, D., ‘Dynamical Systems, Ergodic Theory and Applications’, Springer Verlag, New York (2000).
- G. D. Forney, J., ‘Concatenated Codes’, MIT Press, Cambridge, MA (1966).
- Garello, R., Pierleoni, P., and Benedetto, S., ‘Computing the Free Distance of Turbo Codes and Serially Concatenated Codes with Interleavers: Algorithms and Applications’, *IEEE J. Select. Areas Commun.* **19**, 800 (2001).
- Goldsmith, A., ‘Wireless Communications’, Cambridge University Press, New York (2005).
- Guillén i Fàbregas, A., Martínez, A., and Caire, G., ‘Error Probability of Bit-Interleaved Coded Modulation using the Gaussian Approximation’, in *Proceedings of the Conference on Information Sciences and Systems (CISS) 2004*,

- Princeton, USA (2004).
- Gulati, V. and Narayanan, K. R., 'Concatenated Codes for Fading Channels Based on Recursive Space-Time Trellis Codes', *IEEE Trans. Wireless Commun.* **2**, 118 (2003).
- Hall, E. K. and Wilson, S. G., 'Design and Analysis of Turbo Codes on Rayleigh Fading Channels', *IEEE J. Select. Areas Commun.* **16**, 160 (1998).
- Hasler, M. and Schimming, T., 'Optimal and Suboptimal Chaos Receivers', *Proc. IEEE* **90**, 733 (2002).
- Hayes, S., Grebogi, C., and Ott, E., 'Communicating with chaos', *Phys. Rev. Lett.* **70**, 3031 (1993).
- Haykin, S., 'Communications Systems', John Wiley & Sons, Inc., New York (2001).
- Heidari-Bateni, G. and McGillem, C. D., 'A Chaotic Direct-Sequence Spread-Spectrum Communication System', *IEEE Trans. Commun.* **42**, 1524 (1994).
- Hen, I. and Merhav, N., 'On the Threshold Effect in the Estimation of Chaotic Sequences', *IEEE Trans. Inform. Theory* **50**, 2894 (2004).
- Hokfelt, J., Edfors, O., and Maseng, T., 'On the Theory and Performance of Trellis Termination Methods for Turbo Codes', *IEEE J. Select. Areas Commun.* **19**, 838 (2001).
- Howard, S. L. and Schlegel, C. B., 'Differential Turbo-Coded Modulation with APP Channel Estimation', *IEEE Trans. Commun.* **54**, 1397 (2006).
- Kambo, N. S., Ali, S. A., and Ince, E. A., 'A Simple Moment Based Approximation to Pairwise Error Probability for Memoryless Fading Channels', in *Proceedings of the IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC) 2005*, pp 2543–2546, Berlin, Germany (2005).
- Kennedy, M. P., Kolumban, G., and Jako, Z., 'Performance Evaluation of FM-DCSK Modulation in Multipath Environments', *IEEE Trans. Circuits Syst. I* **47**, 1702 (2000a).
- Kennedy, M. P., Rovatti, R., and Setti, G., 'Chaotic Electronics in Telecommunications', CRC Press, Boca Raton (2000b).
- Kisel, A., Dedieu, H., and Schimming, T., 'Maximum Likelihood Approaches for Noncoherent Communications with Chaotic Carriers', *IEEE Trans. Commun.* **48**, 533 (2001).
- Kliwer, J., Ng, S. X., and Hanzo, L., 'Efficient Computation of EXIT Functions for Nonbinary Iterative Decoding', *IEEE Trans. Commun.* **54**, 2133 (2006).
- Koetter, R., Singer, A. C., and Tuchler, M., 'Turbo Equalization', *IEEE Signal Processing Mag.* **21**, 67 (2004).
- Kolumban, G., 'UWB Technology: Chaotic Communications Versus Noncoherent Impulse Radio', in *Proceedings of the European Conference on Circuit Theory and Design (ECCTD) 2005*, Cork, Ireland (2005).
- Kolumban, G. and Kennedy, M. P., 'The Role of Synchronization in Digital Communications Using Chaos - Part III. Performance Bounds for Correlation Receivers', *IEEE Trans. Circuits Syst. I* **47**, 1673 (2000).
- Kolumban, G., Kennedy, M. P., and Chua, L., 'The Role of Synchronization in Digital Communications Using Chaos - Part I: Fundamentals of Digital Commu-

- nications', IEEE Trans. Circuits Syst. I **44**, 927 (1997).
- Kolumban, G., Kennedy, M. P., and Chua, L., 'The Role of Synchronization in Digital Communications Using Chaos - Part II: Chaotic Modulation and Chaotic Synchronization', IEEE Trans. Circuits Syst. I **45**, 1129 (1998).
- Kolumban, G., Kennedy, M. P., Jako, Z., and Kis, G., 'Chaotic Communications with Correlator Receivers: Theory and Performance Limits', Proc. IEEE **90**, 711 (2002).
- Kozic, S., 'Channel Coding and Modulation Based on Chaotic Systems', PhD thesis, EPFL, Lausanne, Switzerland (2006).
- Kozic, S. and Hasler, M., 'Belief Propagation Decoding of Codes Based on Discretized Chaotic Maps', in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS) 2006*, Kos, Greece (2006).
- Kozic, S., Oshima, K., and Schimming, T., 'How to Repair CSK Using Small Perturbation Control - Case Study and Performance Analysis', in *Proceedings of the European Conference on Circuit Theory and Design (ECCTD) 2003*, Krakow, Poland (2003a).
- Kozic, S., Oshima, K., and Schimming, T., 'Minimum Distance Properties of Coded Modulations Based on Iterated Chaotic Maps', in *Proceedings of the Eleventh International Workshop on Nonlinear Dynamics of Electronic Systems (NDES) 2003*, pp 141–144, Scuol, Switzerland (2003b).
- Kozic, S. and Schimming, T., 'Coded Modulation Based on Higher Dimensional Chaotic Maps', in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS) 2005*, Kobe, Japan (2005).
- Kozic, S., Schimming, T., and Hasler, M., 'Controlled One- and Multi-dimensional Modulations Using Chaotic Maps', IEEE Trans. Circuits Syst. I **53**, 2048 (2006).
- Laney, D. C., Maggio, G. M., Lehmann, F., and Larson, L., 'Multiple Access for UWB Impulse Radio with Pseudochaotic Time Hopping', IEEE J. Select. Areas Commun. **20**, 1692 (2002).
- Laroya, R., 'Coding for Intersymbol Interference Channels - Combined Coding and Precoding', IEEE Trans. Inform. Theory **42**, 1053 (1996).
- Laroya, R., Tretter, S. A., and Farvardin, N., 'A Simple and Effective Precoding Scheme for Noise Whitening on Intersymbol Interference Channels', IEEE Trans. Commun. **41**, 1460 (1993).
- Larson, L., Liu, J.-M., and Tsimring, L. S., 'Digital Communications Using Chaos and Nonlinear Dynamics', Springer, New York (2006).
- Lau, F. C. M. and Tse, C. K., 'Chaos-Based Digital Communication Systems', Springer, Berlin (2003).
- Li, J., Narayanan, K. R., and Georgiades, C. N., 'An Efficient Algorithm to Compute the Euclidean Distance Spectrum of a General Intersymbol Interference Channel and its Applications', IEEE Trans. Commun. **52**, 2041 (2004).
- Li, X. and Ritchey, J. A., 'Bit-Interleaved Coded Modulation with Iterative Decoding', IEEE Commun. Lett. **1**, 169 (1997).
- Li, Y. and Mow, W. H., 'Iterative Decoding of Serially Concatenated Convolutional Codes over Multipath Intersymbol-Interference Channels', in *Proceedings of the*

- International Conference on Communications*, Vol. 2, pp 947–951, Vancouver, Canada (1999).
- Li, Y., Vucetic, B., and Sato, Y., ‘Optimum Soft-Output Detection for Channels with Intersymbol Interference’, *IEEE Trans. Commun.* **41**, 704 (1995).
- Lin, S. and Costello, Jr., D. J., ‘Error Control Coding’, Pearson Prentice Hall, Upper Saddle River, NJ (2004).
- López, L., ‘Information, Communication and Nonlinearity: from the Simple Elements to the Complex Networks’, PhD thesis, Universidad Rey Juan Carlos, Madrid, Spain (2003).
- Luengo, D., ‘Estimación Óptima de Secuencias Caóticas con Aplicaciones en Comunicaciones’, PhD thesis, Universidad de Cantabria, Santander, Spain (2006).
- Luengo, D., Pantaleón, C., and Santamaría, I., ‘Bayesian Estimation of Discrete Chaotic Signals by MCMC’, in *Proceedings of EUSIPCO 2002*, Vol. 1, pp 333–336, Toulouse, France (2002).
- Maggio, G. M., ‘Applications of Symbolic Dynamics to Chaos-Based Communications’, in *Proceedings of Euroattractor 2000*, Warsaw, Poland (2000).
- Maggio, G. M. and De Feo, O., ‘T-CSK: a Robust Approach to Chaos-Based Communications’, in *Proceedings of the International Workshop on Nonlinear Dynamics of Electronic Systems (NDES) 2000*, Catania, Italy (2000).
- Maggio, G. M. and Galias, Z., ‘Applications of Symbolic Dynamics to Differential Chaos Shift Keying’, *IEEE Trans. Circuits Syst. I* **49**, 1729 (2002).
- Maggio, G. M., Rulkov, N., and Reggiani, L., ‘Pseudo-Chaotic Time Hopping for UWB Impulse Radio’, *IEEE Trans. Circuits Syst. I* **48**, 1424 (2001).
- Mariño, I. P., ‘Synchronization and Control of Chaotic Systems. Spatio-Temporal Structures and Applications in Communications’, PhD thesis, Universidad de Santiago de Compostela, Santiago de Compostela, Spain (1999).
- Mariño, I. P., López, L., Míguez, J., and Sanjuán, M. A. F., ‘A Novel Channel Coding Scheme Based on Continuous-Time Chaotic Dynamics’, in *Proceedings of DSP2002*, Vol. 2, pp 1321–1324, Hellas, Greece (2002a).
- Mariño, I. P., López, L., and Sanjuán, M. A. F., ‘Channel Coding in Communications Using Chaos’, *Phys. Rev. A* **295**, 185 (2002b).
- Martínez, A., Guillén i Fàbregas, A., and Caire, G., ‘Error Probability Analysis for Bit-Interleaved Coded Modulation’, *IEEE Trans. Inform. Theory* **52**, 262 (2006).
- Mitra, J. and Lampe, L., ‘Simple Concatenated Codes Using Differential PSK’, in *Proceedings of the Global Communications Conference (GLOBECOM) 2006*, Vol. 1, pp 1–6, San Francisco, USA (2006).
- Narayanan, K. R., ‘Effect of Precoding on the Convergence of Turbo Equalization for Partial Response Channels’, *IEEE J. Select. Areas Commun.* **19**, 686 (2001).
- Narayanan, K. R. and Stüber, G. L., ‘A Serial Concatenation Approach to Iterative Demodulation and Decoding’, *IEEE Trans. Commun.* **47**, 956 (1999).
- Oppenheim, A. V. and Schaffer, R. W., ‘Discrete-Time Signal Processing’, Prentice Hall, New Jersey (1989).
- Otnes, R. and Tüchler, M., ‘Exit Chart Analysis Applied to Adaptive Turbo Equalization’, in *Proceedings of Nordic Signal Processing Symposium*, on board Hur-

- tigruten, Norway (2002).
- Pantaleón, C., Luengo, D., and Santamaría, I., ‘Optimal Estimation of Chaotic Signals Generated by Piecewise-Linear Maps’, *IEEE Signal Processing Lett.* **7**, 235 (2000).
- Pantaleón, C., Luengo, D., and Santamaría, I., ‘Optimal Estimation of a Class of Chaotic Signals’, in *Proceedings of ICSP*, Vol. 1, pp 276–280, Beijing, China (2002).
- Pantaleón, C., Vielva, L., Luengo, D., and Santamaría, I., ‘Bayesian Estimation of Chaotic Signals Generated by Piecewise-Linear Maps’, *Signal Processing* **83**, 659 (2003).
- Papadopoulos, H. C. and Wornell, G. W., ‘Maximum-Likelihood Estimation of a Class of Chaotic Signals’, *IEEE Trans. Inform. Theory* **41**, 312 (1995).
- Papoulis, A. and Pillai, S. U., ‘Probability, Random Variables and Stochastic Processes’, McGraw-Hill, Inc., Boston (2002).
- Perez, L. C., Seghers, J., and Costello, D. J., ‘A Distance Spectrum Interpretation of Turbo Codes’, *IEEE Trans. Inform. Theory* **42**, 1698 (1996).
- Pfister, H. D. and Siegel, P. H., ‘The Serial Concatenation of Rate-1 Codes Through Uniform Random Interleavers’, *IEEE Trans. Inform. Theory* **49**, 1425 (2003).
- Prabhu, V., ‘Intersymbol Interference Performance of Systems with Correlated Digital Signals’, *IEEE Trans. Commun.* **21**, 1147 (1973).
- Press, W. H., Flannery, B. P., Teukolsky, S. A., and Vetterling, W. T., ‘Numerical Recipes in C: The Art of Scientific Computing’, Cambridge University Press, Cambridge (1992).
- Proakis, J. G., ‘Digital Communications’, McGraw-Hill, Inc., Boston (2001).
- Robertson, P. and Wörz, T., ‘Coded Modulation Scheme Employing Turbo Codes’, *Electron. Lett.* **31**, 1546 (1995).
- Robertson, P. and Wörz, T., ‘Bandwidth-Efficient Turbo Trellis-Coded Modulation Using Punctured Component Codes’, *IEEE J. Select. Areas Commun.* **16**, 206 (1998).
- Robertson, P. and Wörz, T., ‘Wiley Encyclopedia of Telecommunications’, Chapt. Turbo Trellis Coded Modulation (TTCM) Employing Parity Bit Puncturing and Parallel Concatenation, pp 1–38, John Wiley & Sons, Inc., New York (2002).
- Rovatti, R., Mazzini, G., and Setti, G., ‘Enhanced Receivers for Chaos-Based DS-CDMA’, *IEEE Trans. Circuits Syst. I* **48**, 818 (2001).
- Rovatti, R., Mazzini, G., Setti, G., and Giovanardi, A., ‘Statistical Modeling of Discrete-Time Chaotic Processes: Advanced Finite-Dimensional Tools and Applications’, *Proc. IEEE* **90**, 820 (2002).
- Ryan, W. E. and Tang, Z., ‘Reduced-Complexity Error-State Diagrams in TCM and ISI Channel Performance Evaluation’, *IEEE Trans. Commun.* **52**, 2047 (2004).
- Sason, I. and Shamai, S., ‘Improved Upper Bounds on the ML Decoding Error Probability of Parallel and Serial Concatenated Turbo Codes via their Ensemble Distance Spectrum’, *IEEE Trans. Inform. Theory* **46**, 24 (2000).
- Sayana, K. and Gelfand, S. B., ‘A General Approach to Performance Evaluation of Higher Order Coded Modulation Systems on Fading Channels’, in *Proceedings of*

- the IEEE Wireless Communications and Networking Conference (WCNC) 2004*, pp 71–76, Atlanta, USA (2004).
- Schimming, T., ‘Chaos Based Modulations from an Information Theory Perspective’, in *Proceedings of the IEEE International Symposium on Signals and Systems (ISCAS) 2001*, Vol. 3, pp 309–312, Sydney, Australia (2001).
- Schimming, T., ‘Statistical Analysis and Optimization of Chaos Based Broadband Communications’, PhD thesis, EPFL, Lausanne, Switzerland (2002).
- Schimming, T., Götz, M., and Schwarz, W., ‘Signal Modeling Using Piecewise Linear Chaotic Generators’, in *Proceedings of EUSIPCO 1998*, pp 1377–1380, Rhodes, Greece (1999).
- Schimming, T. and Hasler, M., ‘Optimal Detection of Differential Chaos Shift Keying’, *IEEE Trans. Circuits Syst. I* **47**, 1712 (2000).
- Schimming, T. and Hasler, M., ‘Potential of Chaos Communication over Noisy Channels - Channel Coding Using Chaotic Piecewise Linear Maps’, in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS) 2002*, Vol. 4, pp 568–571, Scottsdale, USA (2002).
- Schimming, T. and Hasler, M., ‘Coded Modulations Based on Controlled 1-D and 2-D Piecewise Linear Chaotic Maps’, in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS) 2003*, Vol. 3, pp 762–765, Bangkok, Thailand (2003).
- Schimming, T., Oshima, K., and Hasler, M., ‘Coded Modulations Using Chaotic Systems Controlled by Small Perturbations’, in *Proceedings of NOLTA*, Xian, China (2002).
- Schlegel, C. B., ‘Evaluating Distance Spectra and Performance Bounds of Trellis Codes on Channels with Intersymbol Interference’, *IEEE Trans. Inform. Theory* **37**, 627 (1991).
- Schlegel, C. B. and Pérez, L. C., ‘Trellis and Turbo Coding’, John Wiley & Sons, Inc., New York (2004).
- Schweizer, J., ‘The Performance of Chaos Shift Keying: Synchronization versus Symbolic Backtracking’, in *Proceedings of the IEEE International Symposium on Signals and Systems (ISCAS) 1998*, Vol. 4, pp 469–472, Monterey, USA (1998).
- Schweizer, J. and Hasler, M., ‘Multiple Access Communications using Chaotic Signals’, in *Proceedings of the IEEE International Symposium on Signals and Systems (ISCAS) 1996*, Vol. 3, pp 108–111, Atlanta, USA (1996).
- Schweizer, J. and Schimming, T., ‘Symbolic Dynamics for Processing Chaotic Signals-I: Noise Reduction of Chaotic Sequences’, *IEEE Trans. Circuits Syst. I* **48**, 1269 (2001a).
- Schweizer, J. and Schimming, T., ‘Symbolic Dynamics for Processing Chaotic Signals-II: Communication and Coding’, *IEEE Trans. Circuits Syst. I* **48**, 1283 (2001b).
- Setti, G., Mazzini, G., Rovatti, R., and Callegari, S., ‘Statistical Modeling of Discrete-Time Chaotic Processes - Basic Finite-Dimensional Tools and Applications’, *Proc. IEEE* **90**, 662 (2002).
- Setti, G., Rovatti, R., and Mazzini, G., ‘Performance of Chaos-Based Asynchronous

- DS-CDMA with Different Pulse Samples', *IEEE Commun. Lett.* **8**, 416 (2004).
- Sprott, J. C., 'Chaos and time-series analysis', Oxford University Press, Oxford (2003).
- Tam, W. M., Lau, F. C. M., and Tse, C. K., 'A Multiple Access Scheme for Chaos-Based Digital Communication Systems Utilizing Transmitted Reference', *IEEE Trans. Circuits Syst. I* **51**, 1868 (2004).
- ten Brink, S., 'Convergence Behavior of Iteratively Decoded Parallel Concatenated Codes', *IEEE Trans. Commun.* **49**, 1727 (2001).
- Tullberg, H. M. and Siegel, P. H., 'Serial Concatenated Trellis Coded Modulation with Inner Rate-1 Accumulate Code', in *Proceedings of the Global Communications Conference (GLOBECOM) 2001*, Vol. 2, pp 936-940, San Antonio, USA (2001).
- Tullberg, H. M. and Siegel, P. H., 'Serial Concatenated TCM With an Inner Accumulate Code - Part I: Maximum-Likelihood Analysis', *IEEE Trans. Commun.* **53**, 64 (2005a).
- Tullberg, H. M. and Siegel, P. H., 'Serial Concatenated TCM With an Inner Accumulate Code - Part II: Density-Evolution Analysis', *IEEE Trans. Commun.* **53**, 252 (2005b).
- Ungerboeck, G., 'Channel Coding With Multilevel/Phase Signals', *IEEE Trans. Inform. Theory* **28**, 55 (1982).
- Viterbi, A. J., 'Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm', *IEEE Trans. Inform. Theory* **13**, 260 (1967).
- Vucetic, B., 'Bandwidth Efficient Concatenated Coding Schemes for Fading Channels', *IEEE Trans. Commun.* **41**, 50 (1993).
- Wu, Y.-J. and Ogiwara, H., 'Symbol-Interleaver Design for Turbo Trellis-Coded Modulation', *IEEE Commun. Lett.* **8**, 632 (2004).
- Xiaofeng, G., Xingang, W., Meng, Z., and Lai, C. H., 'Chaotic Digital Communication by Encoding Initial Conditions', *CHAOS* **14**, 358 (2004).
- Yeh, P., Zummo, S. A., and Stark, W. E., 'Error Probability of Bit-Interleaved Coded Modulation in Wireless Environments', *IEEE Trans. Veh. Technol.* **55**, 722 (2006).
- Yuan, J., Feng, W., and Vucetic, B., 'Performance of Parallel and Serial Concatenated Codes on Fading Channels', *IEEE Trans. Commun.* **50**, 1600 (2002).



# Curriculum Vitae

## PUBLICATIONS

- Francisco J. Escribano, L. López and M. A. F. Sanjuán, ‘Iteratively Decoding Chaos Encoded Binary Signals’, *Proceedings of the Eighth IEEE International Symposium on Signal Processing and its Applications*, Vol. 1, pp. 275-278 (2005).
- Francisco J. Escribano, L. López and M. A. F. Sanjuán, ‘Evaluation of Channel Coding and Decoding Algorithms Using Discrete Chaotic Maps’, *CHAOS* **16**, 013103 (2006).
- Francisco J. Escribano, L. López and M. A. F. Sanjuán, ‘Exploiting Symbolic Dynamics in Chaos Coded Communications with *Maximum a Posteriori* Algorithm’, *Electron. Lett.* **42**, 984 (2006).
- Francisco J. Escribano, S. Kozic, L. López, M. A. F. Sanjuán and M. Hasler, ‘Turbo-Like Structures for Chaos Coding and Decoding’, *IEEE Trans. Commun.* (Submitted).
- Francisco J. Escribano, L. López and M. A. F. Sanjuán, ‘Chaos Coded Modulations over ISI Channels’, *IEEE Commun. Lett.* (Submitted).
- Francisco J. Escribano, L. López and M. A. F. Sanjuán, ‘Chaos Coded Modulations over Rician and Rayleigh Flat Fading Channels’, *IEEE Commun. Lett.* (Submitted).
- Francisco J. Escribano, L. López and M. A. F. Sanjuán, ‘Bit-Interleaved Chaos Coded Modulations over AWGN and Fading Channels’, *IEEE Trans. Commun.* (Submitted).

## CONFERENCES

- **Conference:** No Lineal 2004  
**Oral Presentation:** ‘Codificación de Canal usando Caos’  
**Authors:** Francisco J. Escribano, L. López and M. A. F. Sanjuán  
**Location and Date:** Toledo, Spain, 1-4 June, 2004
- **Conference:** International Workshop on Nonlinear Dynamics and Complexity in Information and Communication Technology (NDICT)  
**Poster Presentation:** ‘Chaos-Based Channel Coding and Decoding’  
**Authors:** Francisco J. Escribano, L. López and M. A. F. Sanjuán  
**Location and Date:** Bologna, Italy, 6-8 September, 2004

- **Conference:** Eighth IEEE International Symposium on Signal Processing and its Applications (ISSPA)  
**Poster Presentation:** ‘Iteratively Decoding Chaos Encoded Binary Signals’  
**Authors:** Francisco J. Escribano, L. López and M. A. F. Sanjuán  
**Location and Date:** Sydney, Australia, 28-31 August, 2005
- **Conference:** Fourteenth International Workshop on Nonlinear Dynamics in Electronics Systems (NDES)  
**Oral Presentation:** ‘Serial Concatenation of Channel and Chaotic Encoders’  
**Authors:** Francisco J. Escribano, L. López and M. A. F. Sanjuán  
**Location and Date:** Dijon, France, 6-9 June, 2006

### RESEARCH VISITS

- **Place:** Lausanne, Switzerland  
**Department:** Laboratory of Nonlinear Systems at the École Polytechnique Fédéral de Laussane (EPFL)  
**Dates:** September-December 2005  
**Funded by:** Programa Propio de Investigación 2005, Universidad Rey Juan Carlos, Spain



Departamento de Física

# Communications Systems Based on Chaos

## Sistemas de Comunicaciones Basados en Caos

Francisco Javier Escribano Aparicio

Universidad Rey Juan Carlos

Mayo, 2007



# Resumen

## Introducción

Desde su introducción y posterior desarrollo a lo largo del siglo XX, la Teoría del Caos ha logrado trascender el dominio de lo puramente matemático para irse convirtiendo en una herramienta potencialmente útil en el estudio de multitud de fenómenos físicos o químicos, como ciertas reacciones, flujos turbulentos, sistemas físicos no lineales, osciladores y fenómenos de resonancia, etc. Pero también esta teoría ha encontrado aplicación en el estudio de sistemas complejos, sobre los que se puede modelar una gran variedad de fenómenos que atañen, no ya sólo a ciencias como la física, la matemática y la química, sino también a la sociología, a la ingeniería, o a la economía, entre otros campos. De esta forma, dentro del mundo de la ingeniería, encontramos que los sistemas caóticos se están usando ampliamente como herramientas de modelización en procesado de señal, así como en aplicaciones prácticas dentro del ámbito de los conversores analógico-digitales, los conversores de potencia, etc.

En este contexto en el que se ha ido empleando la Teoría del Caos como proveedora de modelos o como ayuda para el estudio de fenómenos complejos o sistemas no lineales, también han merecido especial atención las señales mismas generadas por los sistemas caóticos y sus propiedades específicas. Dadas las características de las señales caóticas, fáciles de producir mediante procedimientos recursivos pero dotadas de una elevada complejidad, no sorprende que pronto se aventurara su aplicación en los sistemas de telecomunicaciones. En tal sentido, a lo largo de los años 90 se intensificaron las investigaciones y los desarrollos que buscaban introducir este tipo de señales en las llamadas tecnologías de la información, bien formando parte de nuevas formas de transmisión y modulación dentro de sistemas de comunicación, por ejemplo en sistemas de espectro ensanchado, o como posibles vehículos para encriptación y protección de datos. Las propiedades que hacen deseables las señales caóticas en este tipo de aplicaciones son su aspecto ruidoso, su carácter de señales de banda ancha, su baja autocorrelación y su facilidad de generación.

Sin embargo, a un comienzo prometedor no le siguió un desarrollo inmediato en el que las modulaciones caóticas mostraran una especial aptitud para competir con los sistemas habituales de transmisión digital, situación que se prolongó con altibajos hasta que, a partir de principios de la presente década, se fueron abriendo nuevas vías que, mediante el establecimiento de una base común para los sistemas de comunicación digital y los sistemas de modulación codificados basados en señales caóticas, permitían contemplar su incorporación en sistemas de comunicación digitales bajo fundamentos bien establecidos más allá de desarrollos ad-hoc no siempre bien justificados. Esta aproximación entre ambos campos permite que puedan estudiarse bajo un mismo esquema conceptual, con las mismas herramientas con las que se desarrollan los sistemas de comunicación, y que se pueda por fin afrontar el problema de su diseño dentro de los mismos parámetros que aquéllos. Además, a lo largo de este breve tiempo se han logrado proponer sistemas basados en caos con

complejidad y rendimientos que van convergiendo con éxito creciente hacia los de los sistemas en uso, con lo que podemos decir que nos hallamos en la antesala de la aplicación directa de las comunicaciones caóticas.

## Antecedentes

Las primeras aproximaciones que pretendían emplear las señales caóticas de forma práctica en las tecnologías de la información se basaban fundamentalmente o bien en desarrollos teóricos difícilmente verificables y poco abordables dentro de los parámetros habituales de las tareas de evaluación y diseño de los sistemas propios de dichas tecnologías, o bien recurrían, ya de forma más práctica, a sistemas de generación analógicos difícilmente controlables o sincronizables y en los que la información no era sencilla de introducir. Una posibilidad de transmisión directa que obviaba estas dificultades y que marcó un hito al ampliar el concepto tradicional de modulaciones por desplazamiento binario, fue la propuesta de sistemas con aplicaciones caóticas alternadas por desplazamiento binario, en los que los bits iban determinando con qué aplicación, de entre dos posibilidades, se iba generando un cierto bloque de muestras, de forma que, en el receptor, se podían ir decodificando dichos bits mediante la identificación de la aplicación con la que se había generado cada uno de los bloques. Sin embargo, estos sistemas no mostraron en principio un buen comportamiento, y siempre estaba el problema de estimar convenientemente en el receptor, a partir de un conjunto de muestras distorsionadas, la aplicación generadora. El inconveniente principal residía en que, si se recurría a algoritmos de estimación sencillos, el resultado era una degradación importante, mientras que la evaluación de la presencia de la aplicación o del valor de la muestra inicial mediante algoritmos más complejos tampoco aseguraba un resultado satisfactorio.

El establecimiento de los conceptos de la dinámica simbólica para el análisis de las señales caóticas vino a añadir un peldaño más al facilitar la propuesta de nuevas formas de manejar y transferir la información dentro de las órbitas caóticas. Además, la aplicación de la dinámica simbólica sobre muestras caóticas cuantificadas permitía identificar el sistema de codificación basado en caos con codificadores convolucionales o, más concretamente, con moduladores codificados de tipo enrejillado, en los que el sistema se asimila conceptualmente con una cadena de Markov en la que existe una correspondencia uno a uno entre un rango de muestras caóticas y un estado del codificador y en el que los bits de la información inducen transiciones binarias, unívocas e identificables dentro del conjunto finito de estados posibles. De esta forma, en el lado del decodificador se habilitaba el uso de técnicas estándar de decodificación de sistemas descritos por enrejillados, con lo que se han ido acumulando en la literatura multitud de formas de transmitir y recuperar la información transportada en una señal caótica que no requieren de especiales desarrollos en el codificador o en el decodificador, y sin que se pierdan esencialmente, aparte de cierto grado de cuantificación, las propiedades de la señal caótica en el canal de comunicación.

Así pues, a día de hoy estamos en condiciones de enfrentar la tarea que suscitó en su momento el interés por las señales caóticas en sistemas de comunicaciones, que

era su potencial frente a determinado tipo de canales, pues en gran medida se ha superado la fase de diseño de codificadores y decodificadores realizables en el ámbito de la transmisión digital de datos. En este sentido, nos centraremos en las llamadas modulaciones caóticas codificadas y en sus posibilidades de diseño a partir de aplicaciones unidimensionales, mientras se evalúa realmente qué tipo de canales son los más adecuados para esta forma de aplicación del caos en las telecomunicaciones.

## Objetivos

### 1. Límites de las modulaciones caóticas basadas en aplicaciones discretas.

En este apartado se revisarán las posibilidades de las aplicaciones caóticas unidimensionales para transmitir la información de forma fiable en canales donde la única fuente de distorsión es el ruido blanco gaussiano aditivo (RBGA). A este fin, se examinará la posibilidad de usar la función de densidad de probabilidad invariante como un posible criterio para el diseño de aplicaciones de este tipo, y se evaluará asimismo qué tipo de propiedades se pueden esperar para las modulaciones codificadas construidas sobre tales aplicaciones en el canal RBGA. También se tratará de encontrar métodos de decodificación de baja complejidad para estas modulaciones basados en el criterio de máximo a posteriori (MAP), a fin de ampliar las posibilidades de los módulos de decodificación de señales caóticas, ya que este criterio minimiza la tasa de error de bit. Este estudio es necesario porque las propuestas MAP realizadas hasta el momento no constituyen alternativas válidas a los métodos de decodificación de señales caóticas basados en el criterio de máxima verosimilitud (MV), debido sobre todo a su elevada complejidad y a sus escasas posibilidades prácticas. Una vez establecidas estas propuestas de diseño de moduladores codificados basados en caos y de decodificadores MAP, se evaluarán sus prestaciones en el canal RBGA mediante los métodos habituales en los sistemas de comunicaciones, tanto por medio de simulaciones como por medio de cotas, en el caso en que sea posible.

### 2. Modulaciones caóticas codificadas en canales dispersivos.

Normalmente la evaluación de las modulaciones caóticas codificadas se ha realizado sólo en canales RBGA, en entornos monousuario y multiusuario (usando aquí señales caóticas como secuencias ensanchadoras, o aplicaciones diversas como medio para discriminar el usuario) y en otras situaciones relacionadas, pero falta realizar un estudio comprensivo sobre las posibilidades de estas modulaciones en entornos dispersivos en los que la fuente de distorsión principal sea la interferencia intersimbólica o los desvanecimientos. En tanto en el canal RBGA las modulaciones caóticas codificadas basadas en aplicaciones unidimensionales no han logrado habitualmente ofrecer ganancias de codificación en comparación con la simple señalización binaria sin codificación de canal, es de esperar que las propiedades de las señales caóticas en el canal dispersivo ofrezcan una mejora de esta situación. Para ello, se examinarán los modelos

de estos canales dispersivos y se evaluará el rendimiento de las modulaciones caóticas codificadas basadas en aplicaciones unidimensionales por medio de simulaciones y de cotas, siempre que sea posible, de forma que se manifiesten sus ventajas potenciales y se puedan ofrecer métodos y criterios de diseño tentativos.

**3. Modulaciones caóticas codificadas concatenadas mediante entrelazadores binarios.**

Recientes resultados apuntan a la necesidad de añadir otro tipo de redundancia a las modulaciones caóticas codificadas, de forma que se incremente la dimensionalidad del sistema dinámico involucrado y se ofrezca una mayor robustez frente a las distorsiones del canal. Mientras una vía explorada con cierto éxito ha sido la ampliación directa de las aplicaciones unidimensionales a aplicaciones n-dimensionales, o la creación de sistemas que alternan aplicaciones unidimensionales bajo el control de la información binaria, nosotros trataremos de conjugar los buenos resultados de los sistemas de codificadores de canal concatenados o de moduladores codificados concatenados con las posibilidades de las señales caóticas en los canales dispersivos. Para ello, se propondrán dos tipos de sistemas, en los que la concatenación se realizará, por un lado, en paralelo y con entrelazado binario, al estilo de los turbocódigos, y, por otro, en serie, al estilo de la concatenación en serie de modulaciones codificadas de tipo enrejillado. Los sistemas propuestos se examinarán en el canal RBGA y en los canales dispersivos evaluados con anterioridad. Asimismo, se comprobarán sus prestaciones, tanto mediante simulación como mediante cotas cuando sea posible, en comparación con los sistemas análogos habituales en transmisión digital de complejidad parecida, a fin de verificar su idoneidad como alternativas realizables frente a dichos sistemas.

## Conclusiones

Las conclusiones generales de la presente tesis se presentan a continuación.

1. Hemos verificado que, en el caso de emplear el tipo de codificación con condiciones iniciales, el diseño de un sistema de comunicaciones bajo el criterio de la densidad de probabilidad invariante de las muestras caóticas no conduce a buenos resultados. De hecho, la densidad de probabilidad invariante puede explicar el comportamiento general de la tasa de error de bit, pero no da cuenta de los detalles, de forma que la dinámica implícita en la aplicación se mantiene como el factor principal que hay que tener en cuenta. Además, hemos visto que en el canal RBGA las aplicaciones no lineales a trozos se comportan peor que su contrapartida lineal a trozos para el tipo de simetría y de sistemas estudiados.
2. Cuando el proceso de decodificación de la secuencia caótica se basa en la estimación de las condiciones iniciales, hemos visto que el rendimiento está domi-

nado básicamente por las muestras de referencia que caen fuera del intervalo de definición después de la adición del ruido. La mejor forma de superar esta situación es realizar la estimación sobre la secuencia completa, con algoritmos MAP o MV, o usar de forma iterativa la decodificación basada en la estimación de las condiciones iniciales. La razón para esto reside en que, a través de dichas posibilidades, la redundancia presente de manera implícita en la secuencia caótica completa se propaga convenientemente y se puede aprovechar así de forma más eficiente. Además, es conocido el hecho de que la decodificación *local* lleva a la aparición de un efecto de umbral.

3. Los malos resultados de los sistemas de codificación y decodificación basados en aplicaciones lineales a trozos o no lineales a trozos nos condujo a considerar la clase de modulaciones caóticas codificadas que, basadas en aplicaciones lineales a trozos controladas por pequeñas perturbaciones, ya habían conducido previamente a sistemas caóticos de comunicaciones más eficientes. Sin embargo, nos limitamos en principio a un caso de bajas prestaciones en el canal RBGA a fin de mostrar la potencial utilidad de las propiedades de codificación del sistema caótico en canales de tipo dispersivo. De hecho, después de proponer un módulo de decodificación MAP adecuado para el sistema de control por pequeñas perturbaciones, mostramos que los efectos de distorsión selectiva en frecuencia (interferencia intersimbólica) y de distorsión no selectiva en frecuencia y variante en el tiempo (desvanecimiento plano) pueden verse atenuados incluso con una modulación caótica codificada tan poco prometedora.
4. Para constuir codificadores con mejores prestaciones tanto en entornos no dispersivos (RBGA) como en entornos dispersivos, se puede recurrir al incremento de la redundancia que se obtiene cuando la tasa de transmisión del sistema es  $R < 1$ . Para realizar esto, decidimos explotar la analogía que los sistemas de modulación caótica codificada poseen con los codificadores convolucionales o con las modulaciones codificadas de tipo enrejillado, y consideramos la posibilidad de diseñar sistemas concatenados en serie y en paralelo. En ambos casos, el objetivo era conseguir una relación más compleja entre el mensaje de entrada binario y las muestras caóticas a la salida como forma de incrementar de forma efectiva la robustez y la redundancia disponible en el sistema.
5. La concatenación en serie con un codificador convolucional como codificador exterior, un entrelazador binario y una modulación caótica codificada como codificador interior condujo a un sistema análogo al de concatenación en serie de moduladores codificados de tipo enrejillado. Mostramos que se pueden diseñar con el esquema caótico codificadores concatenados y decodificadores iterativos prácticos, y que los resultados potenciales pueden analizarse a través de herramientas usuales tomadas del campo de las comunicaciones digitales. El análisis de la convergencia y del error binario, junto con los resultados de las simulaciones, mostraron que el rendimiento final mejora enormemente y que estos nuevos sistemas caóticos pueden competir con las modulaciones codifi-

cadascun con entrelazado binario y con la concatenación en serie de moduladores codificados de tipo enrejillado. De hecho, los resultados fueron especialmente prometedores en canales con interferencia intersimbólica.

6. La concatenación en paralelo de dos modulaciones caóticas codificadas con un entrelazador binario dio como resultado un sistema análogo al de turbo modulación codificada de tipo enrejillado. Describimos de nuevo la posibilidad de conseguir codificadores concatenados y decodificadores iterativos prácticos, y mostramos que el análisis del sistema es también posible a través de herramientas ya conocidas de las comunicaciones digitales. Los resultados mostraron una vez más que se pueden diseñar sistemas caóticos competitivos con sus contrapartidas no caóticas. Además, la concatenación en paralelo de modulaciones caóticas concatenadas ha proporcionado muy buenos resultados en canales con desvanecimiento plano.
7. El análisis de la concatenación en paralelo mostró que los mejores resultados se relacionaban con los sistemas codificadores que están más lejos de cumplir la propiedad de uniformidad de errores. Estos sistemas exhiben una relación muy complicada entre los eventos de error binario a la entrada y la distancia euclídea a la salida, lo cual también sucede con el sistema concatenado en serie gracias al añadido del codificador de canal y del entrelazador binario. Esto prueba que la conjetura acerca de la necesidad de una mayor complejidad en la relación entre mensaje binario a la entrada y muestra caóticas a la salida era cierta, y que se puede llevar a cabo sin desarrollos especiales o totalmente novedosos.
8. Los ejemplos de los Capítulos 3, 4 y 5 han mostrado cómo se puede explotar de forma muy útil la analogía entre los sistemas de comunicación digital y las modulaciones caóticas codificadas. Esto ha permitido el uso de recursos tales como los diagramas de intercambio de información extrínseca, el modelo de canal simétrico con entrada/salida binaria, la evaluación de cotas basadas en eventos de error binario a la entrada relacionados con bucles de errores caóticos a la salida, etcétera. Esto es de la mayor importancia, dado que, de esta forma, las tareas de evaluación y diseño de los sistemas caóticos involucrados se pueden abordar con herramientas conocidas. Sin embargo, hemos hallado que tales expedientes no se adecúan siempre a los esquemas caóticos sin un cierto desajuste debido a sus especiales características, y esto apunta hacia la necesidad de refinar los desarrollos. Por otro lado, todos los ejemplos han mostrado que la cuantificación de la secuencia caótica parece tener un efecto limitado en las propiedades del sistema, que demostró estar más relacionada con las propiedades dinámicas de la aplicación subyacente.

Como conclusión final, podemos decir que hemos demostrado cómo, siguiendo una vía de reciente apertura que une convenientemente las comunicaciones digitales y caóticas, *es posible construir de una forma bastante inmediata sistemas caóticos*

---

*eficientes tanto en canales no dispersivos como en canales dispersivos.* Ahora que podemos tener señales caóticas en el canal sin la penalización de una degradación muy severa en el rendimiento, nos permitimos aventurar el comienzo real de las comunicaciones caóticas codificadas prácticas.