UNIVERSITÀ DEGLI STUDI DI PARMA

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

Dottorato di Ricerca in Tecnologie dell'Informazione

XX Ciclo

Dario Fertonani

LOW-COMPLEXITY ITERATIVE RECEIVERS

Dissertazione presentata per il conseguimento

del titolo di Dottore di Ricerca

Gennaio 2008

Foreword

This thesis presents the main results of the research activity I have carried out during the three-year period spent as a Ph.D. student at the Department of Information Engineering of the University of Parma. The problem of how to provide reliable communications over channels with memory is addressed, with particular emphasis on channels impaired by intersymbol interference and channels impaired by impulse noise. In particular, the receivers to date considered as the most effective for such scenarios, namely the iterative (or *turbo*) receivers, are investigated. Throughout this work, a convenient performance/complexity tradeoff, rather than the performance only, is considered as the key quality figure for a communication system. The main contribution the thesis provides is a set of low-complexity receivers that impressively outperform the existing receivers with similar complexity.

Many people have helped me during these years as a Ph.D. student, and it is now time to thank them. First, I thank my supervisor Prof. Giulio Colavolpe, for his continuous and valuable guidance. Then, I thank all my mates at the Department, who have been and are friends more than colleagues. Finally, I specially thank my parents and my sister, for their unconditioned love and support.

Dario Fertonani January 21, 2008

Contents

Chapter 1

Introduction

This preliminary chapter gives a general overview on the problems underlying the communication systems and anticipates some of the issues addressed in the remainder of the thesis. In particular, Section 1.1 presents historical perspectives on reliable digital communications, Section 1.2 introduces a general system model and part of the notation used throughout the thesis, and Section 1.3 gives an overview of the following chapters.

1.1 Reliable Digital Communications

A very general block scheme of a communication system is depicted in Fig. 1.1. The problem underlying any of such systems can be stated as follows: given a message source, a message sink, and a channel, that is a transmission medium between the source and the sink, how can we design the transmitter and the receiver such that a suitable quality of service (QoS) is assured? The target QoS strictly depends on the application scenarios—for instance, a common definition of the QoS is given in terms of error rate in recovering the transmitted message. In this thesis, we will restrict ourselves to consider digital messages consisting of binary symbols (bits) emitted at a constant transmission rate. Also, we will assume that the message bits (often referred to as information bits) are independent and uniformly distributed—they can be such by nature or made such by compression. Hence, the source being fixed, we will focus on the analysis and the design of the transmitter and the receiver, with the aim of properly coping with the channel impairments.

In general, the transmitter can be seen as a device that outputs a different

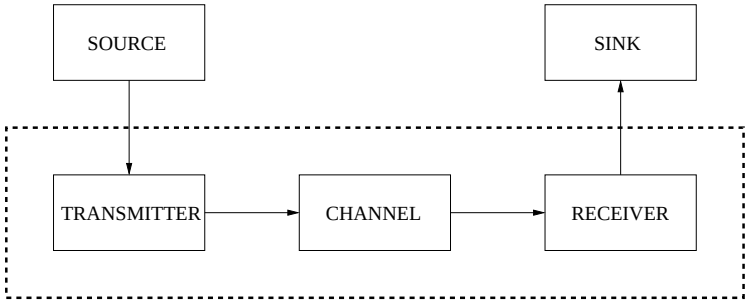

Figure 1.1: Block scheme of a communication system. This thesis focuses on the blocks delimited by the dotted box.

signal for each possible message it wishes to transmit, while a receiver can be seen as a device that processes the signal at the output of the channel for recovering the message. More insights on how effective transmitters and receivers should be have been known since 1948, when Shannon gave birth to the information theory by proving the *noisy channel coding theorem* in the seminal work [1], explaining that a transmitter should encode the message with some redundancy that provides more robustness to the channel impairments, to be then exploited at the receiver side for recovering the original message. Also, Shannon computed the maximum transmission rate at which it is possible to provide error-free transmissions, that is the *capacity* of a channel, and described a coding scheme able to achieve this ultimate limit [1]. Unfortunately, this scheme is unfeasible in practical systems since it requires that the message stream is grouped into information sequences of infinite length, each of them coded into a code sequence of infinite length. Moreover, even when finite-length sequences are considered, the coding scheme proposed by Shannon calls for a decoding algorithm whose complexity grows exponentially with the length of the code sequence [2], and thus results feasible only for very short (and very weak) codes. Hence, the coding scheme in [1] should be seen as an ideal benchmark for any practical scheme.

The Shannon limit had been far from being approached for more than 40 years, up to the invention of the *turbo codes* in 1993 [3]. Such codes, based on a revolutionary encoding rationale and, above all, on an iterative decoding procedure whose complexity grows linearly with the length of the code sequence, inspired many research team that soon after proposed novel coding schemes based on similar principles. Among them, we particularly men-

tion the serially-concatenated convolutional codes [4], the low-density parity-check (LDPC) codes [5, 6], and the repeat-accumulate codes [7, 8].[1] All these schemes can be conveniently investigated by means of the graphical tool presented in [12], basically extending the tool in [13]. As shown in [14], this approach allows to think of the iterative decoding procedure as an instance of the *belief propagation*, known since the 1980's in the field of the computer science [15]. Similar insights are provided by the tool of the *factor graphs* (FGs) and the *sum-product algorithm* (SPA) [16], widely used also in this thesis.

In practice, the problem of how to (very nearly) achieve the Shannon limit is nowadays solved, at least for the most classical communication channel, that is the Gaussian channel [1]. It was indeed proved that the LDPC codes, unlike the turbo codes [17], can provide arbitrarily low error rates under ideal (and unfeasible) decoding [18], as well as under iterative (and feasible) decoding [19]. On the other hand, the problem of providing reliable communications is still open when more challenging channels than the Gaussian one are considered, especially in scenarios where low-complexity receivers are mandatory. This thesis addresses such issues for the case of channels with memory.

1.2 General System Model

In this section, we introduce the system model adopted throughout the thesis, as well as the related mathematical notation. With respect to the very general scheme depicted in Fig. 1.1, we will assume various restrictive hypotheses. On the other hand, we point out that most results of this work can be easily extended to systems that do not match our hypotheses.

We first describe the considered transmitter, whose block scheme is depicted in Fig. 1.2. A sequence of N_I information bits, that is the message from the source, is encoded into a sequence of N_C code bits according to some suitable binary coding scheme with rate $R_C = N_I/N_C$. The code bits, possibly scrambled by means of an interleaver, then feed the modulator, namely a device that emits continuous-time waveforms over the channel. We will only consider linearly-modulated transmissions, so that the lowpass representation

[1] The invention of the LDPC codes dates back to 1963, when Gallager proposed them in [9]. Similarly, the idea of serially concatenating two (or more) convolutional codes was originally proposed by Forney in 1966 [10]. On the other hand, the papers cited above should be acknowledged for the reinterpretation of such schemes based on the *turbo principle* [11].

Figure 1.2: Block scheme of the considered transmitter.

of the transmitted signal at the time instant t yields [20]

$$s(t) = \sum_{k=1}^{K} c_k h_0(t - kT) \tag{1.1}$$

where T is the signaling period, $h_0(t)$ is a suitable shaping pulse, and $\boldsymbol{c}_1^K$ is a sequence of K symbols $\{c_k\}_{k=1}^K$ in one-to-one correspondence with the code bits that feed the modulator.[2] It is also assumed that all symbols $\boldsymbol{c}_1^K$ belong to an M-ary complex-valued alphabet (also referred to as constellation), with M integer power of two. The one-to-one correspondence between the code bits and the modulation symbols considered here just consists of partitioning the N_C code bits into groups of $\log_2 M$ consecutive bits, and mapping each of these M-ary groups onto the constellation, according to some suitable mapping rule. Most modulation alphabets adopted in this thesis will be phase-shift keying (PSK) constellations, combined with a Gray mapping rule [20]. We notice that the considered scheme is an instance of bit-interleaved coded modulation (BICM) [21]. Finally, we assume that the encoder and the mapper are such that the modulation symbols are statistically independent and identically distributed (IID), so that the following factorization holds

$$\mathbb{P}(\boldsymbol{c}_1^K) = \prod_{k=1}^{K} \mathbb{P}(c_k) \tag{1.2}$$

where $\mathbb{P}(\cdot)$ denotes a probability mass function (PMF), and the PMF $\mathbb{P}(c_k)$ does not depend on the time epoch k.

The channel can impair the transmitted signal (1.1) in several ways, so that the received waveform appears as a degraded version of the transmitted one—details on the considered impairments are given in the following chapters. All receivers investigated here can be seen as the cascade of two devices. The first device just processes the received waveform by means of a suitable filtering,

[2] In the remainder of the thesis, we will often use the notation $\boldsymbol{v}_a^b$ for denoting the elements from index $k = a$ to index $k = b$ of a sequence $\{v_k\}$.

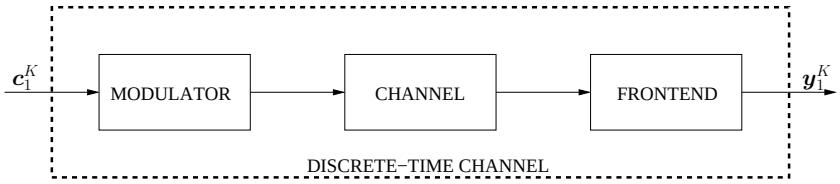

Figure 1.3: Block scheme of the equivalent discrete-time channel.

and, after a proper sampling operation, outputs a sequence of samples [20], while the second device is addressed to recovering the information message by processing such samples. It will be shown in the following chapters that, for the systems considered in this thesis, the sequence at the output of the first device (hereafter referred to as frontend) includes K complex-valued samples, denoted by $\boldsymbol{y}_1^K$, exactly as the transmitted sequence $\boldsymbol{c}_1^K$. It thus turns out convenient to define a discrete-time channel, equivalent to the cascade of the modulator, the actual physical channel, and the frontend (see the block scheme in Fig. 1.3), whose input is $\boldsymbol{c}_1^K$ and whose output is $\boldsymbol{y}_1^K$. This discrete-time model will be used throughout the thesis for both analysis and design purposes.

In general, the discrete-time channel is defined by the probability density function (PDF) $p(\boldsymbol{y}_1^K|\boldsymbol{c}_1^K)$, that is a stochastic description of the received samples $\boldsymbol{y}_1^K$ given the transmission of the sequence $\boldsymbol{c}_1^K$.[3] We will consider systems where, at each time epoch k, it is possible to define a *state* variable s_k (the *memory* of the channel) such that the couple (c_k, s_k) suffices for completely describing the stochastic channel output y_k. Formally, we assume that

$$p(\boldsymbol{y}_1^K|\boldsymbol{c}_1^K, \boldsymbol{s}_1^{K+1}) = \prod_{k=1}^{K} p(y_k|c_k, s_k) \tag{1.3}$$

where an auxiliary final state s_{K+1} is added for reasons that will appear clear later. Moreover, we assume that the state evolves according to a first-order Markov chain [22] over a time-invariant finite-element set, so that the PMF of the state sequence $\boldsymbol{s}_1^{K+1}$ conditioned to the transmission of the sequence $\boldsymbol{c}_1^K$ can be factorized as

$$\mathbb{P}(\boldsymbol{s}_1^{K+1}|\boldsymbol{c}_1^K) = \mathbb{P}(s_1) \prod_{k=1}^{K} \mathbb{P}(s_{k+1}|c_k, s_k) \ . \tag{1.4}$$

[3]The notation $p(\cdot)$ will be used for all PDFs and conditional PDFs. The same notation will be used also for joint events, provided that at least one of them has a continuous support.

Finally, the PDF $p(\boldsymbol{y}_1^K|\boldsymbol{c}_1^K)$ that describes the channel is easily obtained by marginalization, over all possible state sequences, of the PDF $p(\boldsymbol{y}_1^K, \boldsymbol{s}_1^{K+1}|\boldsymbol{c}_1^K)$, factorized as

$$\begin{aligned} p(\boldsymbol{y}_1^K, \boldsymbol{s}_1^{K+1}|\boldsymbol{c}_1^K) &= \mathbb{P}(s_1) \prod_{k=1}^{K} \mathbb{P}(s_{k+1}|c_k, s_k) p(y_k|c_k, s_k) \\ &= \mathbb{P}(s_1) \prod_{k=1}^{K} p(y_k, s_{k+1}|c_k, s_k) \end{aligned} \tag{1.5}$$

by exploiting (1.3) and (1.4). Hence, the channel is completely described by two terms, namely the PMF $\mathbb{P}(s_1)$ and the conditional PDF $p(y_k, s_{k+1}|c_k, s_k)$—we point out the implicit assumption that the latter term does not depend on the time epoch k. All channels considered in this thesis will be characterized by a specific expression of all these terms.

1.3 Overview of the Thesis

The remainder of the thesis is organized as follows.

In Chapter 2, we introduce some general frameworks addressed to all systems matching the finite-state Markovian model given in Section 1.2. In particular, we describe the reference algorithm for soft-output symbol detection, explain how to compute the ultimate performance limits imposed by the considered system model, and introduce a general iterative scheme that characterizes all receivers considered in the thesis.

In Chapter 3, we consider the design of receivers addressed to channels impaired by intersymbol interference, with particular emphasis on the search for low-complexity solutions. First, we give the channel model and show that it matches the general finite-state Markovian model, and thus that the frameworks presented in Chapter 2 can be adopted. Then, we describe the receivers known to be the most effective over such channels, pointing out that their complexity is unmanageable in various scenarios of practical interest. Hence, we present three different approaches for designing low-complexity algorithms, and report several simulation results that prove the better effectiveness the proposed schemes provide with respect to the existing solutions with similar complexity.

In Chapter 4, we consider the design of receivers addressed to channels impaired by impulse noise. First, we introduce a Markovian channel model

able to describe the typical bursty nature that characterizes the impulse noise in most scenarios of practical interest, and show, by means of information-theoretical arguments, that significant performance improvements with respect to the memoryless models generally considered in the literature can be achieved by exploiting the memory of the impulse noise. Moreover, we describe a couple of practical communication schemes that fairly approach such ultimate performance limits, with focus on a novel scheme able to provide an impressive performance/complexity tradeoff. Then, we address the problem of designing robust receivers, which can effectively work even in the absence of reliable information on the statistical properties of the impulse noise.

In Chapter 5, we finally give some concluding remarks on the main obtained results.

Chapter 2

General Frameworks for Finite-State Channels

This chapter introduces some general frameworks addressed to all systems matching the model given in Section 1.2. In particular, Section 2.1 describes the reference algorithm for soft-output symbol detection, Section 2.2 explains how to compute the ultimate performance limits imposed by the considered system model, and Section 2.3 introduces a general iterative scheme that characterizes all receivers considered in this thesis.

2.1 Soft-Output Symbol Detection

This section addresses the *detection* problem, basis of most issues related to a communication system. In general, it can be formulated as the problem of determinating suitable information on the transmitted symbols $\boldsymbol{c}_1^K$ by exploiting the observation of the received sequence $\boldsymbol{y}_1^K$. This process, in its basic version referred to as *hard-output* detection, outputs a sequence $\hat{\boldsymbol{c}}_1^K$ of modulation symbols that, according to some proper criterion, provide the best estimates of the transmitted one. For instance, over a channel matching the model given in Section 1.2, hard-output sequence detection can be carried out by means of the Viterbi algorithm when the maximum likelihood (ML) criterion is adopted [23, 24]. We are here interested in a more complex process, referred to as *soft-output* symbol detection, which produces, for each time epoch k and for each possible value of the random variable c_k, either the *likelihood* information $p(\boldsymbol{y}_1^K|c_k)$ or the *a posteriori* probability (APP) $\mathbb{P}(c_k|\boldsymbol{y}_1^K)$. The need

for soft-output symbol detection will be pointed out in Section 2.3.

In the following, we derive an algorithm for soft-output symbol detection by means of probabilistic arguments. This algorithm is often referred to as Bahl-Cocke-Jelinek-Raviv (BCJR) algorithm for acknowledging the paper [25], although it first appeared in [26, 27].[1] We first focus on the likelihood information $p(\boldsymbol{y}_1^K|c_k)$, which can be written as

$$\begin{aligned} p(\boldsymbol{y}_1^K|c_k) &= \sum_{s_k}\sum_{s_{k+1}} p(\boldsymbol{y}_1^K, s_k, s_{k+1}|c_k) \\ &= \sum_{s_k}\sum_{s_{k+1}} p(\boldsymbol{y}_1^{k-1}, s_k) p(\boldsymbol{y}_{k+1}^K|s_{k+1}) p(y_k, s_{k+1}|c_k, s_k) \end{aligned} \quad (2.1)$$

by exploiting the Markovian properties described in Section 1.2. To make the notation simpler, we introduce the coefficients (or *metrics*)

$$\alpha_k(s_k) = p(\boldsymbol{y}_1^{k-1}, s_k) \quad (2.2)$$

$$\beta_k(s_k) = p(\boldsymbol{y}_k^K|s_k) \quad (2.3)$$

so that the dependence on the received samples no longer explicitly appears. The key points of the algorithm are two recursive equations for computing the coefficients $\{\alpha_k(s_k)\}$ and $\{\beta_k(s_k)\}$ as follows [25–27]

$$\alpha_{k+1}(s_{k+1}) = \sum_{s_k}\sum_{c_k} \alpha_k(s_k) p(y_k, s_{k+1}|s_k, c_k)\mathbb{P}(c_k) \quad (2.4)$$

$$\beta_k(s_k) = \sum_{s_{k+1}}\sum_{c_k} \beta_{k+1}(s_{k+1}) p(y_k, s_{k+1}|s_k, c_k)\mathbb{P}(c_k)\ . \quad (2.5)$$

By accounting for (2.2) and (2.3), the *forward recursion* (2.4) is initialized with $\alpha_1(s_1) = \mathbb{P}(s_1)$ for all values of s_1, while the *backward recursion* (2.5) is initialized with

$$\beta_{K+1}(s_{K+1}) = \begin{cases} 0 & \text{if } \mathbb{P}(s_{K+1}) = 0 \\ 1 & \text{otherwise} \end{cases}\ . \quad (2.6)$$

Finally, the so-called *completion* stage (2.1) yields

$$p(\boldsymbol{y}_1^K|c_k) = \sum_{s_k}\sum_{s_{k+1}} \alpha_k(s_k)\beta_{k+1}(s_{k+1}) p(y_k, s_{k+1}|c_k, s_k)\ . \quad (2.7)$$

[1] We notice that the same algorithm can be derived by means of the SPA over a proper FG [16]. Since this approach hides some probabilistic details widely exploited in this thesis, the probabilistic derivation given in [25–27] is here preferred. The graphical derivation given in [16] is recalled in Section 2.1.1.

When APPs are desired instead of the likelihood information, the same algorithm can be used. Indeed, from the following equation[2]

$$\mathbb{P}(c_k|\boldsymbol{y}_1^K) = \frac{\mathbb{P}(c_k)p(\boldsymbol{y}_1^K|c_k)}{p(\boldsymbol{y}_1^K)} \propto \mathbb{P}(c_k)p(\boldsymbol{y}_1^K|c_k) \tag{2.8}$$

and by exploiting (2.7), we get

$$\mathbb{P}(c_k|\boldsymbol{y}_1^K) \propto \mathbb{P}(c_k) \sum_{s_k} \sum_{s_{k+1}} \alpha_k(s_k)\beta_{k+1}(s_{k+1})p(y_k, s_{k+1}|c_k, s_k) \; . \tag{2.9}$$

If required, the correct scaling factor can be easily computed by accounting that, for any value of k, the summation of the terms $\{\mathbb{P}(c_k|\boldsymbol{y}_1^K)\}$ over all values of c_k must be unitary.

Finally, we point out that the described *forward-backward* algorithm (FBA) may raise numerical problems, since many metrics often go to zero after (relatively) few steps of the recursions. A way for solving this problem consists of properly scaling, after each step of the recursions, the computed metrics—for instance, we scale the metrics such that, at each time epoch of both recursions, the maximum value is unitary. In this case, the equality symbols in (2.2), (2.3), and (2.7) should be replaced by proportionality symbols, but the detection process is not affected. Further improvements in terms of numerical stability are obtained by implementing the algorithm in the logarithmic domain [28].

2.1.1 Interpretation by Factor Graphs

The same FBA can be alternatively derived by means of the SPA over a proper FG [16]. The starting point of this approach is the factorization

$$\mathbb{P}(\boldsymbol{c}_1^K, \boldsymbol{s}_1^{K+1}|\boldsymbol{y}_1^K) \propto \mathbb{P}(s_1) \prod_{k=1}^{K} p(y_k, s_{k+1}|c_k, s_k)\mathbb{P}(c_k) \tag{2.10}$$

deriving from (1.2) and (1.5). The FG corresponding to (2.10) is depicted in Fig. 2.1, where we defined the function nodes $\{P_k\}$ and $\{G_k\}$ as

$$P_k = \mathbb{P}(c_k) \tag{2.11}$$

$$G_k = \begin{cases} \mathbb{P}(s_1) & \text{if } \; k = 0 \\ p(y_k, s_{k+1}|c_k, s_k) & \text{otherwise} \end{cases} \tag{2.12}$$

[2]The proportionality symbol $\propto$ will be used when two quantities differ by factors irrelevant for the detection process [16].

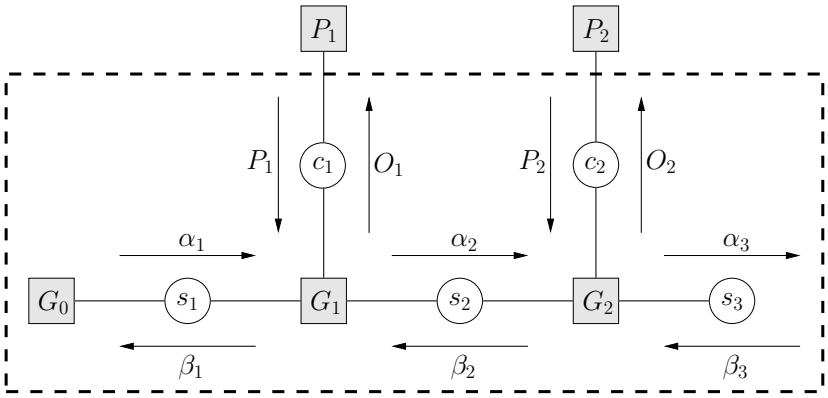

Figure 2.1: Factor graph corresponding to (2.10), when $K = 2$. The dotted box will be exploited in Section 2.3.

so that the arguments of the function nodes are not explicitly indicated, to make the notation simpler. By applying the SPA [16], we obtain exactly the recursive equations (2.4) and (2.5), as well as the completion stage (2.9). Moreover, the SPA allows to identify the messages $\{O_k\}$ incoming to the function nodes $\{P_k\}$ as the likelihood PDFs $\{p(\boldsymbol{y}_1^K|c_k)\}$.

2.2 Ultimate Achievable Information Rate

An interesting problem related to the analysis and the design of a communication system is the evaluation of the ultimate performance limit imposed by a given channel, to be interpreted as an ideal benchmark for any practical system over the same channel. In particular, the performance limits considered in this thesis are given in terms of ultimate achievable information rate [29].

In the case of the system described in Section 1.2, the information rate I, defined as the number of information bits transmitted per channel use (that is per signaling period), is equal to the product of the rate of the binary code and the number of bits per modulation symbol, and thus yields $I = R_C \log_2 M$. One of the key results of the information theory is that error-free communications are, in principle, possible when the actual information rate I does not exceed the limit [1, 29]

$$I_0 = \lim_{K\to\infty} \frac{1}{K} \log_2 \left[\frac{p(\boldsymbol{y}_1^K|\boldsymbol{c}_1^K)}{p(\boldsymbol{y}_1^K)} \right] . \tag{2.13}$$

Although the rigorous definition of the ultimate achievable information rate I_0 would require a statistical expectation of the right-hand side of (2.13), the definition considered here is correct under the assumptions stated in Section 1.2 [29].

In most cases of interest, it is unfortunately not possible to analytically compute the quantity I_0. On the other hand, there exist simulation-based recursive algorithms that can provide accurate estimations of I_0—see [30, 31] for earlier specific results and [32–34] for the general algorithm recalled in the following.[3] The simulation consists of generating, according to the statistics of the source and the channel, a sequence $\boldsymbol{c}_1^K$ of modulation symbols and the corresponding sequence $\boldsymbol{y}_1^K$ of received samples—the choice of the value of K is clarified later. The terms $p(\boldsymbol{y}_1^K|\boldsymbol{c}_1^K)$ and $p(\boldsymbol{y}_1^K)$ required for the evaluation of (2.13) can be computed by marginalization, as follows

$$p(\boldsymbol{y}_1^K|\boldsymbol{c}_1^K) = \sum_{s_{K+1}} p(\boldsymbol{y}_1^K, s_{K+1}|\boldsymbol{c}_1^K) \tag{2.14}$$

$$p(\boldsymbol{y}_1^K) = \sum_{s_{K+1}} p(\boldsymbol{y}_1^K, s_{K+1}) = \sum_{s_{K+1}} \alpha_{K+1}(s_{K+1}) \tag{2.15}$$

where we exploited (2.2). Equation (2.15) explains how to compute the term $p(\boldsymbol{y}_1^K)$ by means of the forward recursion of the detection algorithm described in Section 2.1. A very similar method allows to compute the term $p(\boldsymbol{y}_1^K|\boldsymbol{c}_1^K)$. Indeed, if we define the modified forward recursion

$$\mu_{k+1}(s_{k+1}) = \sum_{s_k} \mu_k(s_k) p(y_k, s_{k+1}|s_k, c_k) \tag{2.16}$$

initialized with $\mu_1(s_1) = \mathbb{P}(s_1)$ for all values of s_1, it is simple to prove that the metrics $\{\mu_k(s_k)\}$ have the following probabilistic meaning

$$\mu_k(s_k) = p(\boldsymbol{y}_1^{k-1}, s_k|\boldsymbol{c}_1^{k-1}) \tag{2.17}$$

and thus that (2.14) can be rewritten as

$$p(\boldsymbol{y}_1^K|\boldsymbol{c}_1^K) = \sum_{s_{K+1}} \mu_{K+1}(s_{K+1}) \,. \tag{2.18}$$

[3] In [34], even the sequence $\boldsymbol{c}_1^K$ is allowed to be Markovian, while we assumed in (1.2) that $\boldsymbol{c}_1^K$ includes independent symbols.

Hence, the simulation-based estimation of the ultimate achievable information rate yields

$$\frac{1}{K}\log_2\left[\frac{\sum_{s_{K+1}}\mu_{K+1}(s_{K+1})}{\sum_{s_{K+1}}\alpha_{K+1}(s_{K+1})}\right] \tag{2.19}$$

from (2.13), (2.15), and (2.18). As discussed at the end of Section 2.1, a simple way for avoiding problems of numerical stability consists of properly scaling the metrics after each step of the recursions,[4] and further improvements are obtained by implementing the algorithm in the logarithmic domain [28].

The choice of the value of K is critical for the estimation procedure. The value of K should be such large that the estimation based on (2.19) converges to the true value given by the limit in (2.13), according to some suitable confidence criterion. A pragmatic approach for the choice of the value of K, explained in [34], is recalled in the following. Let K_G be a guess on a suitable value of K, and run some tens of simulations, each with a different seed for the random generator and with $K = K_G$. Then, if all simulation-based estimations match the desired accuracy (for instance, if the maximum and the minimum outcomes differ for less than 0.03 bits per channel use) output their average as a final estimation, otherwise, increase the value of K_G and repeat the procedure. All simulation results reported in this thesis were obtained by means of this approach. Although we found that the minimum value of K providing a given target accuracy strongly depends on the system parameters, it was never required to adopt values of K larger than 10^7.

2.2.1 Mismatched Decoding

We will often address the problem of evaluating the ultimate information rate achievable when a finite-state Markovian model is assumed for design purposes, but the actual system does not match the assumed model. This condition, generally referred to as mismatched decoding [35], is of interest when the actual system cannot be described by means of a finite-state Markovian model, as well as when the actual system is finite-state Markovian, but is somehow mismatched with respect to the assumed model.[5] For instance, this latter case occurs when the receiver is provided with erroneous information on the

[4] In this case, the additional constraint of preserving, at each time epoch k, the ratio between the terms $\{\mu_k(s_k)\}$ and $\{\alpha_k(s_k)\}$ must be accounted for, as clear from (2.19).

[5] The condition in which the assumed model and the actual one are equal will be referred to as *matched decoding*.

statistics of the actual Markov chain, or when the assumed Markov chain is deliberately an approximation of the actual one that leads to a simpler finite-state representation. Both cases will be addressed in the thesis.

The ultimate information rate achievable by a mismatched receiver can be computed by means of the same algorithm described before, with a modification in the simulation procedure [34]. Namely, the sequences $\boldsymbol{c}_1^K$ and $\boldsymbol{y}_1^K$ should be generated according to the statistics of the actual source and the actual channel, while the metrics $\{\mu_k(s_k)\}$ and $\{\alpha_k(s_k)\}$ should be computed according to the statistics of the assumed source and the assumed channel.[6] The obtained ultimate information rate is obviously lower than that achievable by a matched receiver, and the difference between them vanishes only if the assumed model and the actual one coincide with unitary probability—see [34] for a formal proof. On the other hand, we will show that there exist various cases where the impact of mismatched decoding is limited, such that (intentional) mismatches can be exploited for deriving low-complexity receivers.

2.3 Iterative Receivers

In principle, error-free communications are possible when the system provides an information rate I that does not exceed the ultimate limit I_0 if the (ideal) coding scheme defined by Shannon is adopted [1, 29]. In this section, we describe the practical (that is with finite complexity) receivers that, to date, are considered the most effective in approaching the benchmark performance of the ideal scheme when combined with the BICMs addressed in this thesis.

Let $\boldsymbol{a}_1^{N_I}$ be the sequence of information bits, and $\boldsymbol{b}_1^{N_C}$ be the sequence of (interleaved) code bits. A receiver that wishes to minimize the bit-error rate (BER) should estimate the information message $\boldsymbol{a}_1^{N_I}$ based on the maximum *a posteriori* (MAP) criterion on the symbol probability [25], which consists of estimating a_k as

$$\begin{cases} 1 & \text{if } \mathbb{P}(a_k = 1|\boldsymbol{y}_1^K) > \mathbb{P}(a_k = 0|\boldsymbol{y}_1^K) \\ 0 & \text{otherwise} \end{cases} \quad , \quad k = 1, 2, \ldots, N_I \; . \tag{2.20}$$

[6] Formally, the claims above are valid under the assumptions that the actual system and the assumed one share the same channel-input domain and the same channel-output domain [34]. These hypotheses are assured for the considered discrete-time channel model (see Section 1.2), where both domains are sequences of K complex numbers.

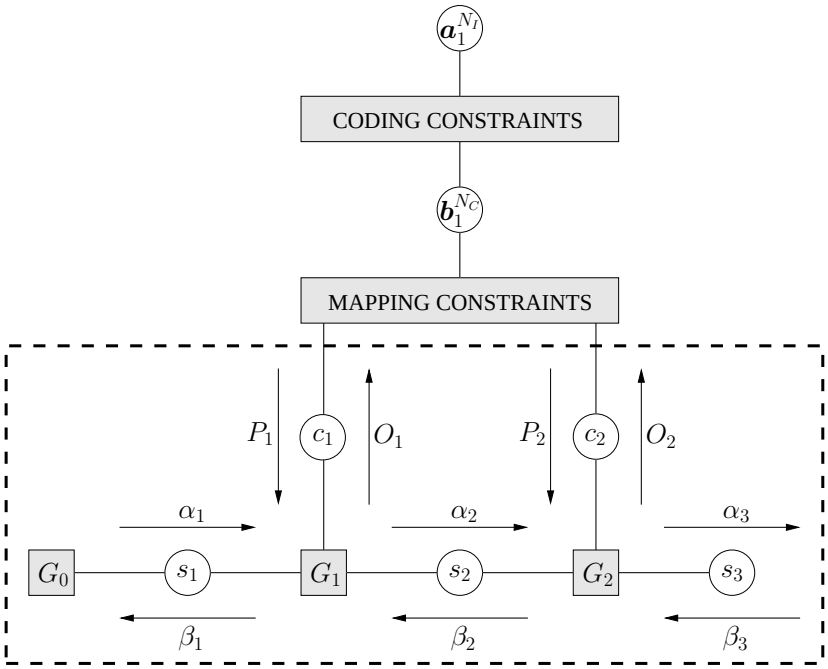

Figure 2.2: Factor graph corresponding to (2.21), when $K = 2$. The part included in the dotted box is the same as in Fig. 2.1.

Hence, the APPs $\{\mathbb{P}(a_k|\boldsymbol{y}_1^K)\}$ should be computed. This problem can be formulated as marginalization of the PMF [16]

$$\mathbb{P}(\boldsymbol{a}_1^{N_I}, \boldsymbol{b}_1^{N_C}, \boldsymbol{c}_1^K, \boldsymbol{s}_1^{K+1}|\boldsymbol{y}_1^K) \propto p(\boldsymbol{y}_1^K, \boldsymbol{s}_1^{K+1}|\boldsymbol{c}_1^K)\mathbb{P}(\boldsymbol{c}_1^K|\boldsymbol{b}_1^{N_C})\mathbb{P}(\boldsymbol{b}_1^{N_C}|\boldsymbol{a}_1^{N_I}) \quad (2.21)$$

where the right-hand side results from simple manipulations that exploit the properties described in Section 1.2. The FG corresponding to (2.21) is depicted in Fig. 2.2, where we factorized the term $p(\boldsymbol{y}_1^K, \boldsymbol{s}_1^{K+1}|\boldsymbol{c}_1^K)$ as in (1.5), and we referred to the terms $\mathbb{P}(\boldsymbol{c}_1^K|\boldsymbol{b}_1^{N_C})$ and $\mathbb{P}(\boldsymbol{b}_1^{N_C}|\boldsymbol{a}_1^{N_I})$ as mapping constraints and coding constraints, respectively. We point out that the part of the FG included in the dotted box is exactly the same as in Fig. 2.1, and thus can be managed by means of the FBA described in Section 2.1. On the other hand, since the messages $\{P_k\}$ are now the output of the mapping subgraph instead of *a priori* probabilities as in (2.12), the probabilistic meanings of the various messages are different. In particular, the messages $\{O_k\}$ are no longer equal to the

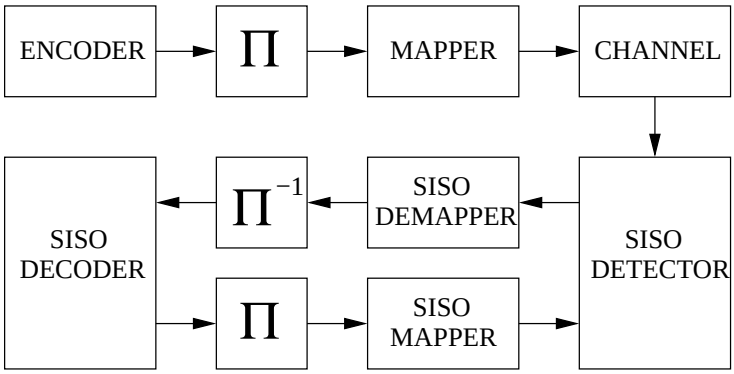

Figure 2.3: General block scheme adopted in the remainder of the thesis.

likelihood PDFs $\{p(\boldsymbol{y}_1^K|c_k)\}$. The messages $\{P_k\}$ and $\{O_k\}$ are often referred to, respectively, as *intrinsic* information and *extrinsic* information related to the detection algorithm [16, 36].

For all systems of practical interest, the marginalization of (2.21) can be exactly carried out only by brute-force marginalization—in the context of [16], this is explained by noticing that the FG in Fig. 2.2 contains cycles. Since the brute-force marginalization is practically unmanageable for all suitable (that is large) values of K, suboptimal algorithms are required that well approximate the optimal one. Given the general framework in [16], it turns out natural to adopt a scheme based on the iterative exchange of information between three soft-input soft-output (SISO) devices, that is one for each term at the right-hand side of (2.21) or, equivalently, for each subgraph in Fig. 2.2. Namely, we will consider a SISO decoder addressed to the coding constraints, a SISO mapper/demapper addressed to the mapping constraints, and a SISO detector. The key idea any iterative receiver is based on is that the quality of the messages over the graph, that is their reliability, can be refined by exchanging information between SISO devices addressed to specific subgraphs—see [16] for the general framework and [37,38] for the landmark application known as *turbo equalization*. This rationale leads to a receiver whose block scheme is depicted in Fig. 2.3, where the transmitter and the channel are also represented. We notice that the interleaver Π and the deinterleaver Π^{-1} could alternatively be considered as part of the coding/decoding blocks, as implicitly done in Fig. 2.2. We also notice that in Fig. 2.2 the SISO detector coincides with the

FBA, but in general it can be a different SISO algorithm. One of the aims of this thesis indeed consists of designing alternative SISO detectors that provide a convenient performance/complexity tradeoff. On the other hand, we will not address in detail the other SISO blocks—see the papers cited in Section 1.1 for SISO decoding and [39] for SISO mapping/demapping.

We finally remark that the SPA, when applied to graphs with cycles, cannot lead to a unique *schedule* nor to a unique stopping criterion for the message passing [16]. When not otherwise stated, we will consider the following schedule:

1. initialize all messages over the graph;
2. run the SISO detector;
3. run the SISO demapper;
4. run the SISO decoder;
5. if the stopping criterion is satisfied go to step 8;
6. run the SISO mapper;
7. go to step 2;
8. stop the algorithm and output the decoded message.

In general, we will use a stopping criterion based on the maximum number of runs of the SISO decoder. When possible, for instance when LDPC codes are adopted, we will also use syndrome-based stopping criteria [20]. We finally notice that even the subgraphs can, in general, contain cycles—this is the case, for instance, of the subgraph related to the coding constraints when turbo codes or LDPC codes are employed. Hence, we will also give details on the schedule characterizing any subgraph that contains cycles.

2.3.1 Tools for Convergence Analysis of Iterative Receivers

As stated before, the described iterative receiver provides an approximate solution for the practically unmanageable marginalization of the PMF (2.21). It is thus of great interest to find some suitable criterion that provides insights on the effectiveness of a given iterative scheme, in terms of capability of converging to a (nearly) exact marginalization and in terms of convergence speed.

Unfortunately, while there exist instantaneous criteria for finding out ineffective schemes, for instance the presence of degree-4 cycles in the graph [16], there do not exist instantaneous criteria for finding out effective schemes. In the following a basic overview of some analysis tools other than the error-rate simulations is given.

Currently, the highest standard for a graph-based analysis of iterative algorithms is definitely the so-called *density evolution* (DE), first proposed in [19] for investigating the convergence properties of the SPA when applied to LDPC decoding and then applied to many other scenarios—for instance, see [40] for the case of multi-user scenarios. DE basically consists of evaluating the PDFs that describe the messages over the graph, and tracking how they evolve during the iterative process. This method is very involved, such that it often becomes computationally unmanageable for the finite-state Markovian systems of interest here. Approximations of DE exist that result much simpler since they assume the target PDF to match a suitable canonical PDF, and thus just describe the evolution of the (few) parameters that characterize the canonical PDF. For instance, see [41] for a Gaussian approximation of DE. A different approach, proposed in [42], turns out to be very effective over a subset of the channels considered in this thesis, namely the finite-state Markovian channels with data-independent memory, where $\mathbb{P}(\boldsymbol{s}_1^{K+1}|\boldsymbol{c}_1^K) = \mathbb{P}(\boldsymbol{s}_1^{K+1})$.

We will use convergence tools based on the extrinsic information transfer (EXIT) charts, which were originally addressed to simpler schemes [43], but, in the version presented in [44], are suitable for the schemes of interest here. The EXIT charts provide fast insights on the convergence properties, although they do not assure rigorous conclusions due to the presence of several approximations in the derivation of the method [43, 44]. On the other hand, we will show that the EXIT charts suffice for the purposes of this work.

Chapter 3

Channels Impaired by Intersymbol Interference

This chapter addresses the design of algorithms for SISO detection over channels impaired by intersymbol interference, with particular emphasis on the search for low-complexity solutions. The chapter is organized as follows. Section 3.1 introduces the channel model, while Section 3.2 recalls the algorithms for optimal SISO detection and points out that their complexity is unmanageable in various scenarios of practical interest. Section 3.3, Section 3.4, and Section 3.5 present three different approaches for designing low-complexity algorithms.

3.1 Channel Model

We consider a channel that degrades the transmitted signal (1.1) with both distortion effects and additive white Gaussian noise (AWGN). In particular, we assume that the distortions can be modeled by means of a linear filter with impulse response $d(t)$, so that the received signal yields [20]

$$\begin{aligned} r(t) &= s(t) \otimes d(t) + w(t) \\ &= \sum_k c_k h_0(t - kT) \otimes d(t) + w(t) \end{aligned} \tag{3.1}$$

where $\otimes$ denotes convolution, and $w(t)$ is a zero-mean circularly-symmetric white Gaussian process with power spectral density (PSD) $2N_0$. By defining

the shaping pulse

$$h(t) = h_0(t) \otimes d(t) \tag{3.2}$$

equivalent to the cascade of the transmission and channel filters, the received signal can be simply written as

$$r(t) = \sum_k c_k h(t - kT) + w(t) \tag{3.3}$$

and can be thus thought of as a signal linearly modulated by means of the shaping pulse $h(t)$, impaired by AWGN only. We will exploit this interpretation, referring hereafter to the shaping pulse $h(t)$ and its Fourier transform $H(f)$ without any distinction about the specific contributions of the transmission and channel filters. Moreover, we will assume that the shaping pulse $h(t)$ is normalized to unitary energy and that the modulation symbols are independent, uniformly distributed, and zero-mean.

The amount of information carried by the continuous-time signal (3.3) is the same, at least for the detection purposes of interest here, as that carried by the sequence of samples $\{x_k\}$, where

$$x_k = \int_{-\infty}^{\infty} r(t) h^*(t - kT) dt \tag{3.4}$$

at each time epoch k [45,46]. In practice, the sequence $\{x_k\}$ is the output, sampled at symbol rate, of the *matched filter* [20] related to the shaping pulse $h(t)$, fed by the received signal $r(t)$. Once defined the sequence $\{g_i\}$ such that

$$g_i = \int_{-\infty}^{\infty} h(t) h^*(t - iT) dt \quad , \quad \forall i \in \mathbb{Z} \tag{3.5}$$

and the integer L such that

$$\begin{cases} |g_L| > 0 \\ g_i = 0 \quad , \quad \forall i : |i| > L \end{cases} \tag{3.6}$$

the following discrete-time channel model turns out

$$x_k = \sum_{i=-L}^{L} g_i c_{k-i} + n_k \tag{3.7}$$

where the noise samples $\{n_k\}$ are zero-mean circularly-symmetric Gaussian random variables with autocorrelation

$$\mathbb{E}\{n_k n_{k-i}^*\} = 2N_0 g_i \quad , \quad \forall i \in \mathbb{Z} \tag{3.8}$$

$\mathbb{E}\{\cdot\}$ being the expectation operator. In general, the received samples are thus impaired by intersymbol interference (ISI), that is by the contributions of modulation symbols transmitted at different time epochs, and correlated (or colored) noise. We point out that the minimum possible value of L is zero, since the energy g_0 of the shaping pulse is assumed unitary, while, rigorously, the existence of a finite value of L is not assured. On the other hand, since for all practical shaping pulses the following limit holds

$$\lim_{i \to \pm\infty} g_i = 0 \tag{3.9}$$

it is always possible to choose a finite value of L that makes the model (3.7) as accurate as desired. To manage the border effects in (3.7), we will assume that the modulation symbols $\boldsymbol{c}_1^K$ are preceded (and followed) by at least L modulation symbols.

A model alternative to (3.7) that provides a discrete-time description of the ISI channels can be found in [45] and is recalled in the following. The model is derived by employing a procedure known as spectral factorization [47], which basically consists of finding out a sequence $\boldsymbol{f}_0^L$ such that

$$\sum_{n=i}^{L} f_n f_{n-i}^* = g_i \quad , \quad \forall i : 0 \leq i \leq L \; . \tag{3.10}$$

Then, after a proper filtering [45] of the sequence $\{x_k\}$ that produces the sequence $\{y_k\}$, the following discrete-time channel model results

$$y_k = \sum_{i=0}^{L} f_i c_{k-i} + w_k \tag{3.11}$$

where the noise samples $\{w_k\}$ are zero-mean circularly-symmetric Gaussian random variables with autocorrelation

$$\mathbb{E}\{w_k w_{k-i}^*\} = \begin{cases} 2N_0 & \text{if } \; i = 0 \\ 0 & \text{otherwise} \end{cases} \; . \tag{3.12}$$

Hence, the noise samples $\{w_k\}$ are independent (or white), unlike the noise samples $\{n_k\}$. To acknowledge the papers [45,46], the models (3.7) and (3.11) will be referred to as Ungerboeck model and Forney model, respectively.

We finally notice that the sequence $\boldsymbol{g}_0^L$ suffices for characterizing $\boldsymbol{g}_{-L}^L$, thanks to the Hermitian symmetry deriving from (3.5). Moreover, we point out that, for a given sequence $\boldsymbol{g}_0^L$, there can exist several sequences $\boldsymbol{f}_0^L$ solving (3.10) [47], and thus that different Forney models can correspond to the same Ungerboeck model. This fact will be clarified in the next section.

3.1.1 Examples of Channels

This section introduces three different classes of ISI channels that will be often addressed in the following. For simplicity, only real-valued sequences $\boldsymbol{g}_0^L$ and $\boldsymbol{f}_0^L$ are reported.

The first class is the root raised cosine (RRC) class, which is very particular since it is characterized by $L = 0$ and thus does not introduce ISI at all [20]—we will adopt this class as a benchmark for those that actually introduce ISI. A shaping pulse $h(t)$ belongs to the RRC class if $|H(f)|^2$ is proportional to

$$S_{\mathrm{RRC}}(f) = \begin{cases} 1 & \text{if } |f| \leq \frac{1-\alpha}{T} \\ 0 & \text{if } |f| > \frac{1+\alpha}{T} \\ \cos^2\left[\frac{\pi T}{2\alpha}|f| - \frac{\pi}{2}\frac{1-\alpha}{\alpha}\right] & \text{otherwise} \end{cases} \tag{3.13}$$

where $\alpha \in [0,1]$ is the roll-off factor [20].

The second class consists of faster-than-Nyquist (FTN) signaling [48]. This class provides a very attractive spectral efficiency by introducing intentional ISI [48, 49]. After few manipulations of the formulations given in [48, 49], it turns out that a shaping pulse $h(t)$ belongs to the FTN class if $|H(f)|^2$ is proportional to $S_{\mathrm{RRC}}(f/\tau)$, where $\tau \in (0,1]$ is the time-compression factor—the FTN class reduces to the RRC class when $\tau = 1$. Once defined $s_{\mathrm{RRC}}(t)$ as the inverse Fourier transform of $S_{\mathrm{RRC}}(f)$, further manipulations allow to write

$$g_i \propto s_{\mathrm{RRC}}(i\tau) \quad , \quad \forall i \in \mathbb{Z} \, . \tag{3.14}$$

We point out that (3.14) does not lead to a finite value of L whenever $\tau < 1$. Hereafter, we will thus redefine L as the minimum integer such that

$$\frac{|g_i|}{g_0} < 10^{-2} \quad , \quad \forall i : |i| > L \, . \tag{3.15}$$

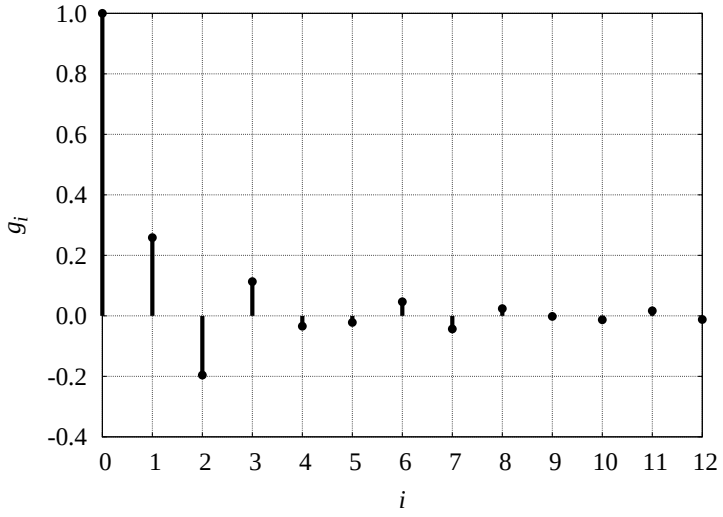

Figure 3.1: Sequence $\boldsymbol{g}_0^L$ equivalent to FTN signaling with roll-off factor $\alpha = 0.1$ and time-compression factor $\tau = 0.78$.

Interestingly, we found that none of the results that follow significantly change if the right-hand side of (3.15) is decreased, that is if a better approximation is adopted. The sequence $\boldsymbol{g}_0^L$ corresponding to FTN signaling with $\alpha = 0.1$ and $\tau = 0.78$ is reported in Fig. 3.1.[1]

The last example class, widely used in models for magnetic-storage systems, is characterized by shaping pulses $h(t)$ proportional to

$$h_{\mathrm{LOR}}(t) - h_{\mathrm{LOR}}(t - T) \tag{3.16}$$

where $h_{\mathrm{LOR}}(t)$ is the Lorentzian pulse

$$h_{\mathrm{LOR}}(t) = \left[1 + \left(\frac{2t}{DT}\right)^2\right]^{-1} \tag{3.17}$$

$D > 0$ being a real parameter called density [20]. These channels will be referred to as magnetic channels, for simplicity. The corresponding sequence $\boldsymbol{g}_0^L$ related to the density $D = 3$ is reported in Fig. 3.2. For the same case, we also

[1]The parameters $\alpha = 0.1$ and $\tau = 0.78$ are of interest since they allow to achieve the Mazo limit [48, 49].

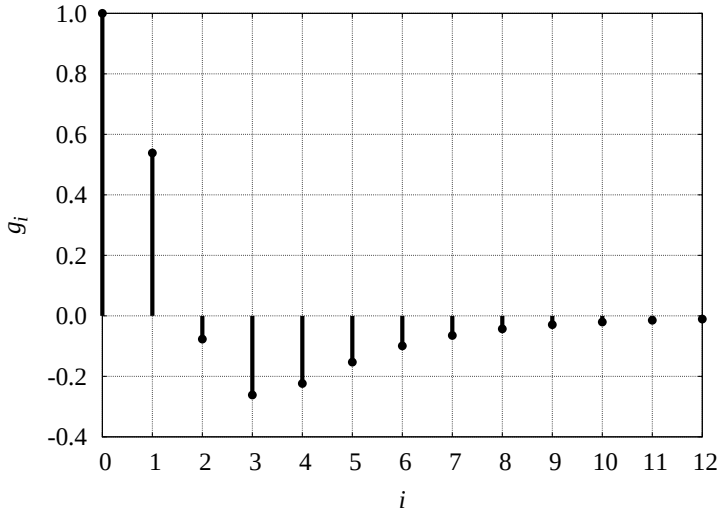

Figure 3.2: Sequence $\boldsymbol{g}_0^L$ equivalent to a magnetic channel with density $D = 3$.

report two different sequences $\boldsymbol{f}_0^L$ in Fig. 3.3 and Fig. 3.4—let us recall that different Forney models can correspond to the same Ungerboeck model. Given an Ungerboeck representation, the corresponding Forney representations can be partitioned into *minimum-phase*, *mixed-phase*, and *maximum-phase* sequences, according to the definitions in [50]. We here remark a property that will be exploited in the following sections: the sequences with minimum (respectively, maximum) phase are those that concentrate, as much as possible, their energy over the first (respectively, last) elements, while all remaining sequences are mixed-phase [50]. This property is well clarified by comparing the minimum-phase sequence in Fig. 3.3 and the mixed-phase sequence in Fig. 3.4, both corresponding to the same Ungerboeck representation reported in Fig. 3.2.

3.2 Optimal Soft-Output Symbol Detection

This section introduces two algorithms for exact MAP symbol detection, one addressed to the Ungerboeck model and one to the Forney model. These algorithms are the reference solutions for SISO detection in iterative receivers (see the discussion in Section 2.3)—in the case of ISI channels, the iterative

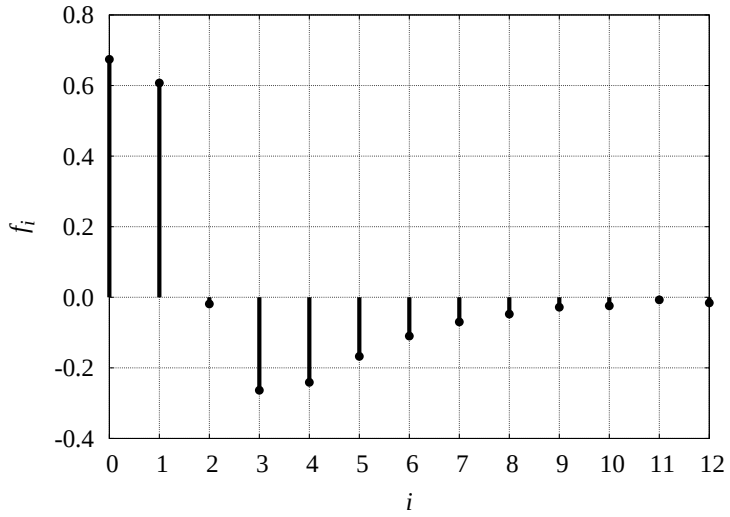

Figure 3.3: Minimum-phase sequence $\boldsymbol{f}_0^L$ corresponding to a magnetic channel with density $D = 3$.

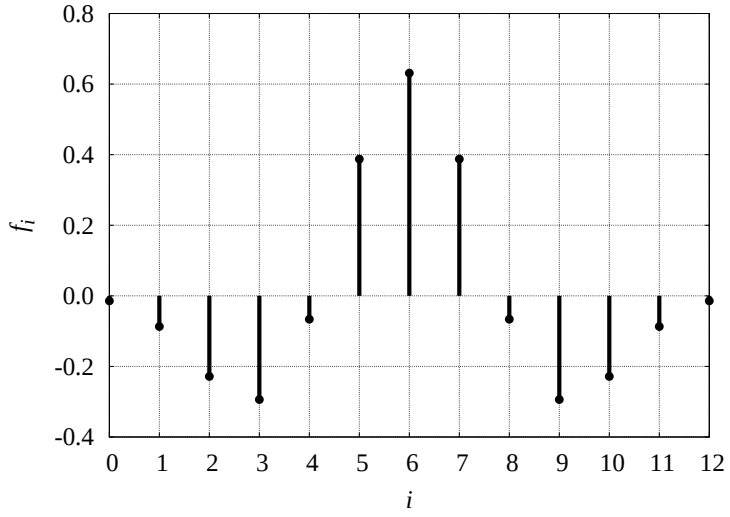

Figure 3.4: Mixed-phase sequence $\boldsymbol{f}_0^L$ corresponding to a magnetic channel with density $D = 3$.

procedure is often referred to as *turbo equalization* [37, 38]. We point out that the following treatment assumes the presence of ISI, namely that $L > 0$, since otherwise MAP symbol detection is simply carried out by means of a memoryless algorithm. Moreover, we notice that ideal channel estimation is assumed hereafter, that is the value of N_0 and the ISI coefficients are assumed perfectly known to the receiver.

A key difference between the Ungerboeck and Forney models, beside the correlation of the noise, is due to the modulation symbols that generate ISI. Namely, at each time epoch k, other than the *current* symbol c_k, the sample x_k is impaired by L *past* symbols and L *future* symbols, while the sample y_k only experiences ISI from L past symbols. Hence, only the Forney model is finite-state Markovian, with the channel state defined as

$$s_k = \boldsymbol{c}_{k-L}^{k-1} \tag{3.18}$$

so that the conditional PDF describing the received samples yields

$$p(y_k|c_k, s_k) = \frac{1}{2\pi N_0} \exp\left[-\frac{1}{2N_0} \left| y_k - \sum_{i=0}^{L} f_i c_{k-i} \right|^2 \right]. \tag{3.19}$$

Since the PMF $\mathbb{P}(s_{k+1}|c_k, s_k)$ is implicit in the state definition (3.18), the only terms required to complete the model given is Section 1.2 is the PMF $\mathbb{P}(s_1)$ of the initial state s_1, which can be computed by means of the equation

$$\mathbb{P}(s_k) = \prod_{i=1}^{L} \mathbb{P}(c_{k-i}) \tag{3.20}$$

deriving from (1.2) and (3.18). Hence, when the Forney model is adopted, MAP symbol detection can be exactly carried out by means of the FBA recalled in Section 2.1.

It does not seem possible to derive, by means of probabilistic arguments, an algorithm for exact MAP symbol detection when the Ungerboeck model is adopted, this model being non-Markovian. On the other hand, it was shown in [51] that exact MAP symbol detection can be carried out by exploiting the SPA over a proper FG. This approach leads to the following factorization [51]

$$\mathbb{P}(\boldsymbol{c}_1^K, \boldsymbol{s}_1^{K+1}|\boldsymbol{x}_1^K) \propto \mathbb{P}(s_1) \prod_{k=1}^{K} \mathbb{P}(s_{k+1}|c_k, s_k) U_k(c_k, s_k) \mathbb{P}(c_k) \tag{3.21}$$

where

$$U_k(c_k, s_k) = \exp\left[\frac{1}{N_0}\mathrm{Re}\left\{x_k c_k^* - \frac{1}{2}g_0|c_k|^2 - \sum_{i=1}^{L} g_i c_k^* c_{k-i}\right\}\right] \tag{3.22}$$

and $\mathrm{Re}\{\cdot\}$ denotes the real component of a complex number—we point out that the definition of the state s_k is the same as for the Forney model, hence equations (3.18) and (3.20) hold for the Ungerboeck model too. The right-hand side in (3.21) can be represented by means of the FG depicted in Fig. 2.1, that is the same FG adopted for the Forney model, once redefined the function node G_k as

$$G_k = \begin{cases} \mathbb{P}(s_1) & \text{if } \ k = 0 \\ \mathbb{P}(s_{k+1}|c_k, s_k)U_k(c_k, s_k) & \text{otherwise} \end{cases} . \tag{3.23}$$

Since the FGs corresponding to the Forney and Ungerboeck models are identical, the marginalization required to compute the APPs of the modulation symbols can be carried out by means of the same FBA. We remark that both algorithms are exact, since the corresponding FGs do not contain cycles, and thus perform identically. On the other hand, we also remark that the redefinition of the function nodes makes the state metrics $\{\alpha_k(s_k)\}$ and $\{\beta_k(s_k)\}$ different in the two cases [51]. In particular, the probabilistic meanings of the state metrics discussed in Section 2.1 and exploited later in this chapter no longer hold for the Ungerboeck model.

3.2.1 Ultimate Achievable Information Rate

Some examples that show how the ultimate achievable information rate varies with the signal-to-noise ratio (SNR) are reported in the following, for different ISI channels. We will adopt the SNR definition

$$\mathrm{SNR} = \frac{g_0}{2N_0}\mathbb{E}\{|c_k|^2\} = \frac{1}{2N_0}\mathbb{E}\{|c_k|^2\} \tag{3.24}$$

discussed in [20]. In all cases, matched decoding is considered, that is the actual channel and the channel assumed by the receiver coincide.

The results reported in Fig. 3.5 refer to systems that adopt a binary PSK (BPSK) modulation over the channels described in the following.

- Channel C1: $L = 0$, $\boldsymbol{g}_0^L = 1$.

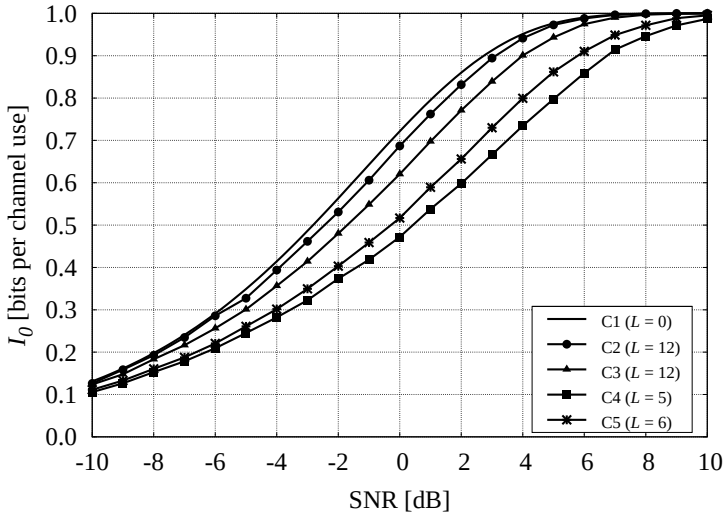

Figure 3.5: Information rate achievable over different channels when a BPSK modulation is adopted and matched decoding is assumed.

- Channel C2: $L = 12$, $\boldsymbol{g}_0^L$ reported in Fig. 3.1.
- Channel C3: $L = 12$, $\boldsymbol{g}_0^L$ reported in Fig. 3.2.
- Channel C4: $L = 5$, $\boldsymbol{g}_0^L = \frac{1}{6}(6, 5, 4, 3, 2, 1)$.
- Channel C5: $L = 6$, $\boldsymbol{g}_0^L = \frac{1}{140}(140, 112, 85, 60, 38, 20, 7)$.

In particular, channel C1 corresponds to the RRC class with arbitrary roll-off factor (that is an AWGN channel without ISI), channel C2 corresponds to the FTN class with $\alpha = 0.1$ and $\tau = 0.78$, and channel C3 corresponds to a magnetic channel with density $D = 3$. Channel C4 and C5 do not correspond to any class introduced in Section 3.1.1, but present some features, discussed later, that provide several insights on the topics discussed in the remainder of the chapter. We point out that, since the evaluation of the achievable information rate, carried out as explained in Section 2.2, requires that a Markovian model is assumed, the Forney model was exploited for generating the results reported in Fig. 3.5. On the other hand, in conditions of matched decoding, the theoretical limit characterizing a given channel obviously does not depend on the adopted model, and thus is exactly the same for the Ungerboeck model

and for any equivalent Forney model (irrespectively of its phase), and is achievable by a receiver that employs the corresponding algorithm for MAP symbol detection [34]. We notice that the presence of ISI degrades the achievable information rate with respect to the pure AWGN channel C1.[2] Interestingly, channel C4, although characterized by a value of L much lower than channels C2 and C3, turns out to be the worst channel in Fig. 3.5. This suggests that, rather than the value of L only, a key role is played by the energy distribution over the channel coefficients.

3.2.2 Complexity Issues

The FBA works on a trellis representing the finite-state machine that describes the channel, whose number of states, according to (3.18), is equal to M^L per time epoch. Hence, long channel impulse responses and large modulation alphabets make the implementation of the exact FBA impractical and motivate the search for suboptimal algorithms with a convenient performance/complexity tradeoff. Most reduced-complexity algorithms in the literature address the Forney model, maintain the three-stage structure of the FBA (forward recursion, backward recursion, and completion stage), and obtain the complexity reduction by performing a simplified trellis search. We can approximately classify the various algorithms in the literature based on two different rationales, described in the following.

In the first class, inspired to reduced-state sequence detection (RSSD) [53–55] originally addressed to the Viterbi algorithm, we mention the reduced-state BCJR (RS-BCJR) algorithm [56] and the generalized reduced-state algorithms proposed in [57]. They exploit the fact that the forward and backward recursions of the BCJR algorithm reduce to the Viterbi algorithm when the *max-log* approximation [28] is adopted. Hence, the concept of survivor results. Then, assuming that only a part of the information corresponding to the full state is embedded in a properly defined reduced state, they recover the missing information by decision feedback, in a way similar to RSSD. The RS-BCJR algorithm is particularly effective on minimum-phase channels, where generally provides a high-quality soft-output. For this reason, when employed in turbo equalization schemes, it ensures a good convergence of the iterative process.

[2] If Markovian sources are allowed, the presence of ISI can instead improve the achievable information rate [34]. On the other hand, it is conjectured in [52] that such improvements are (partially) due to the fact that the SNR definition (3.24) is no longer fair when Markovian sources are employed. For simplicity, Markovian sources are not considered in the thesis.

The algorithms presented in [57], which extend the concept of state reduction, achieve a good performance on a larger subset of ISI channels, but the corresponding performance loss with respect to the full-state BCJR algorithm increases when they are employed in turbo equalization schemes, because of the poor quality of the generated soft-output.

The algorithms in the second class perform a reduced search on the original full-complexity trellis, instead of a full search on a reduced-state trellis. The M-BCJR algorithm [58] belongs to this class. It provides a good performance/complexity tradeoff on a large subset of ISI channels, but it often does not result effective in producing a high-quality soft-output [59]. As a consequence of the fact that the M-BCJR algorithm gives a predominant role to the forward recursion, since only the trellis paths selected during this stage are explored in the backward recursion, it cannot properly cope with maximum-phase or mixed-phase channels. A couple of solutions to this problem, based on an independent trellis search in the backward recursion, were proposed in [60] and in [61].

Among the reduced-complexity algorithms that do not belong to the described classes, we mention the algorithm described in [62], based on a confidence criterion that can be used to detect reliable information symbols early on during decoding, and the algorithm presented in [63], which addresses the particular case of sparse ISI channels, namely channels with a number of non-zero interferers much lower than the value of L. In the following sections, we present alternative techniques for complexity reduction.

3.3 Intentional Mismatched Decoding

This section presents low-complexity detection schemes that exploit intentional mismatched decoding. Namely, the channel assumed by the receiver is deliberately an approximation of the actual one, characterized by a lower number of interferers and thus by a simpler finite-state machine. The approximation proposed in the following is based on the Forney model, but can be easily extended to the Ungerboeck model.

Let $\boldsymbol{f}_0^L$ be a Forney representation of a given ISI channel, and let Q be an integer such that $0 \leq Q \leq L$. The received sample (3.11) can be written as

$$y_k = \sum_{i=0}^{Q} f_i c_{k-i} + i_k \tag{3.25}$$

where

$$i_k = \sum_{i=Q+1}^{L} f_i c_{k-i} + w_k \ . \tag{3.26}$$

We are interested in evaluating the performance of a receiver that neglects the dependence of the terms $\{i_k\}$ on the modulation symbols, just considering them as noise. In particular, we approximate the terms $\{i_k\}$ as a sequence $\{\hat{w}_k\}$ of IID circularly-symmetric Gaussian random variables, with the same mean and variance as the terms $\{i_k\}$. After few manipulations, it turns out that the noise samples $\{\hat{w}_k\}$ have mean zero and variance

$$2\hat{N}_0 = \mathbb{E}\{|i_k|^2\} = 2N_0 \ + \ \mathbb{E}\{|c_k|^2\} \sum_{i=Q+1}^{L} |f_i|^2 \ . \tag{3.27}$$

Hence, at the the receiver side, the actual channel is approximated as

$$\hat{y}_k = \sum_{i=0}^{Q} f_i c_{k-i} + \hat{w}_k \tag{3.28}$$

so that the assumed channel still matches the Forney model, but is now characterized by a finite-state machine with M^Q states instead of M^L states. We also point out that, whenever $Q < L$, the assumed noise variance $2\hat{N}_0$ is larger than the actual one $2N_0$, according to (3.27)—in practice, the noise variance is increased by a term exactly equal to the signal energy neglected by the simpler finite-state machine.

It is natural to wonder which one, between the different Forney representations that correspond to a given ISI channel, results the most suited to the described intentional mismatches. Since the approximation (3.28) basically consists of a truncation of the actual channel to the first $Q + 1$ coefficients, it is expected to be more effective when most of the channel energy is concentrated in those coefficients. Hence, the minimum-phase representations are expected to be the most convenient. We will show in the next section that the simulation results definitely validate this conjecture.[3]

[3]This conclusion obviously changes if a different approach is adopted. For instance, if a truncation that only accounts for the last $Q + 1$ coefficients, instead of the first ones, is employed, the maximum-phase representations are to be preferred.

3.3.1 Simulation Results

The ultimate information rate achievable by a receiver that assumes the channel model (3.28) when the actual channel is (3.11) can be evaluated as explained in Section 2.2.1. Some results related to BPSK transmissions over channel C2 are reported in Fig. 3.7. The minimum-phase representation was adopted. First, we notice that the choice $Q = 5$ practically does not affect the achievable information rate with respect to the case of matched decoding (that is $Q = L = 12$). Hence, there is no need for a FBA working on a trellis with $2^{12} = 4096$ states, since (nearly) the same theoretical limit is provided by a FBA working on a trellis with $2^5 = 32$ states. Then, we notice that the SNR degradation due to mismatched decoding tends to increase with the value of the achievable information rate. In particular, the choice $Q = 1$ causes a degradation lower than 1 dB when the target information rate is equal to 0.5 bits per channel use, and a degradation larger than 2 dB when the target information rate is larger than 0.9 bits per channel use. It is natural to conjecture that this behavior is due to the fact that, at low values of the achievable information rate, that is at low values of the SNR, worse approximations in the channel model can be tolerated since the additive noise is anyway the dominant impairment.

Other results, related to BPSK transmissions over a magnetic channel with density $D = 3$ (that is channel C3), are reported in Fig. 3.7. In all cases but that denoted as mixed-phase, the minimum-phase representation reported in Fig. 3.3 was adopted. We notice that the *qualitative* behaviors discussed relatively to the results in Fig. 3.6 are confirmed, even if the *quantitative* degradation due to mismatched decoding is larger for a given value of Q. In particular, if the minimum-phase representation is adopted, the choice $Q = 6$ practically does not affect the achievable information rate with respect to the case of matched decoding, and thus we can adopt, without any performance degradation, a FBA working on a trellis with only $2^6 = 64$ states. On the other hand, if the mixed-phase representation is adopted, the choice $Q = 6$ leads to a very poor performance, as expected.

We also assessed, by means of computer simulations, the performance of the mismatched receivers when combined with practical coding schemes, in terms of BER versus SNR. Some of these results are reported in Fig. 3.8 and Fig. 3.9, for the case of BPSK transmissions over channel C3, that is for the case whose ultimate performance limits are reported in Fig. 3.7. Two different LDPC codes with codeword length of $50,000$ bits are used, namely one with

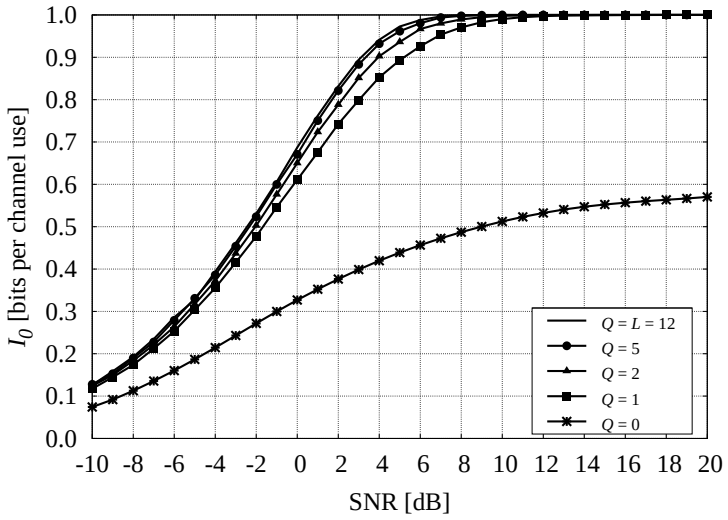

Figure 3.6: Information rate achievable over channel C2 in various conditions of intentional mismatched decoding, when a BPSK modulation is adopted.

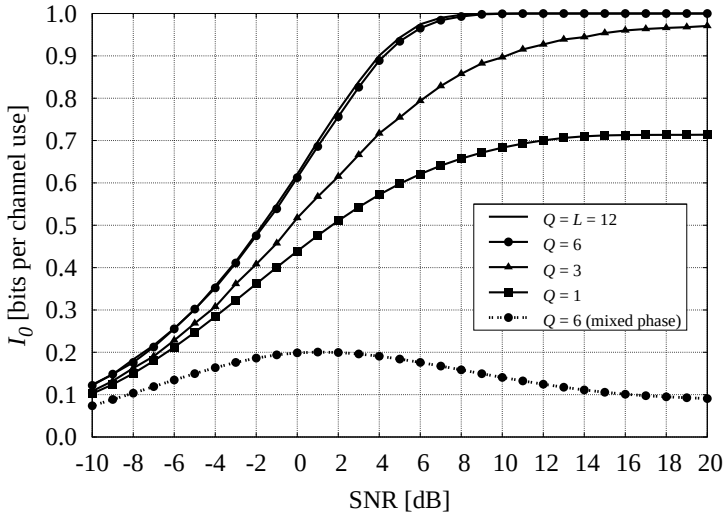

Figure 3.7: Information rate achievable over channel C3 in various conditions of intentional mismatched decoding, when a BPSK modulation is adopted.

code rate 1/2 in Fig. 3.8 and one with code rate 7/8 in Fig. 3.9. A detection instance is executed before each iteration of the SISO decoder, for a maximum of 50 iterations. The process also stops if, by checking the code syndrome, a valid codeword is found before the 50^{th} iteration. No interleaver is used because of the random nature of the LDPC code. To ensure better numerical stability, all algorithms are implemented in the logarithmic domain [28]. We point out that the simulation results related to the matched receivers are, in both cases, incomplete, since it is nearly unfeasible to obtain reliable BER curves for turbo equalizer working on a 4096-state trellis. On the other hand, we notice that such results suffice for stating that the performance degradations due to intentional mismatched decoding well agree with the theoretical previsions. In particular, in both cases, a mismatched FBA working on a 32-state trellis assures a very limited degradation, thus providing a very convenient performance/complexity tradeoff.

Other simulation results, omitted here, validate the theoretical previsions also for channel C2, confirming that intentional mismatched decoding can be very convenient on minimum-phase channels. On the other hand, we point out that there exist ISI channels where intentional mismatched decoding is not effective. In particular, we found that huge degradations occur in both theoretical limits and performance of practical coding schemes when channel C4 is considered, even if the choice $Q = L - 1$ is adopted. Channel C4 is very particular, since all Forney representations coincide[4] and are mixed-phase, so that alternative techniques for complexity reduction, as those presented in the next sections, should be employed.

3.4 Reduced Trellis Search

This section presents three different algorithms that obtain complexity reduction by exploiting the probabilistic meanings of the state metrics propagated in the FBA when the Forney model is adopted—we previously presented these algorithms in [64, 65]. In principle, the same techniques can be applied to the FBA when the Ungerboeck model is adopted, since the structure of the algorithms is identical. On the other hand, as explained in Section 3.2, the state

[4] In our notation, two (or more) Forney representations coincide when they only differ for a multiplicative complex factor with unitary magnitude. Indeed, if the sequence $\boldsymbol{f}_0^L$ satisfies (3.10), a sequence obtained by multiplying all elements of $\boldsymbol{f}_0^L$ by an arbitrary complex number with unitary magnitude satisfies (3.10) too.

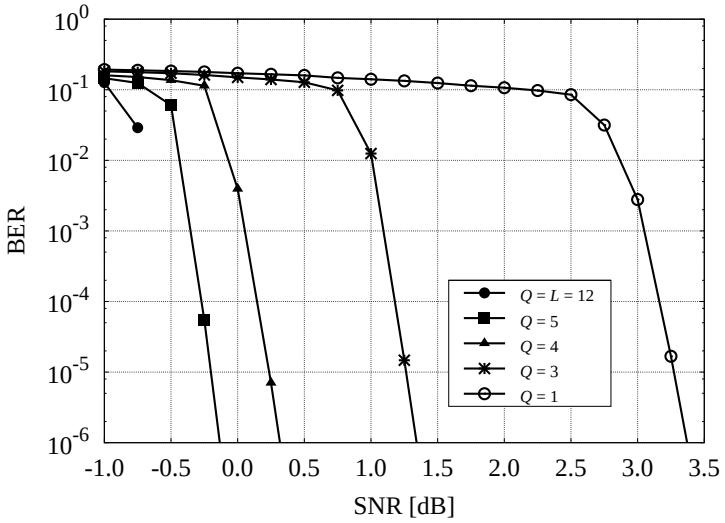

Figure 3.8: Performance over channel C3, when a coded BPSK modulation is adopted, in various conditions of intentional mismatched decoding. An LDPC code with rate 1/2 is employed.

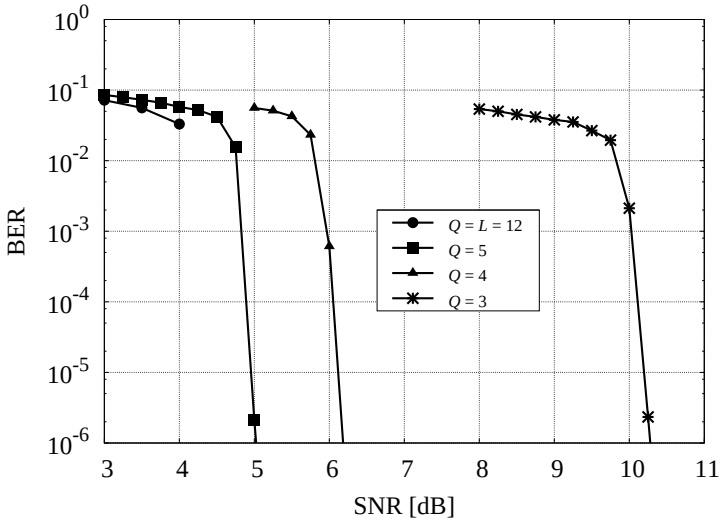

Figure 3.9: Performance over channel C3, when a coded BPSK modulation is adopted, in various conditions of intentional mismatched decoding. An LDPC code with rate 7/8 is employed.

metrics have the probabilistic meanings exploited here only when the Forney model is adopted, and thus the proposed algorithms cannot be expected to properly work if the Ungerboeck model is considered. Since this fact was also confirmed by simulation results, only the Forney model is addressed in this section—see [66] for applications of reduced trellis search to the FBA based on the Ungerboeck model.

A formulation of the FBA simpler than the general one recalled in Section 2.1, yet equivalent, is given in the following. It is easily derived by exploiting that the PMF $\mathbb{P}(s_{k+1}|c_k, s_k)$ is, in this case, an indicator function, equal to one if c_k, s_k, and s_{k+1} satisfy (3.18), and to zero otherwise. In particular, once defined the *branch metric* as

$$F_k(c_k, s_k) = \mathbb{P}(c_k) \exp\left[-\frac{1}{2N_0}\left|y_k - \sum_{i=0}^{L} f_i c_{k-i}\right|^2\right] \tag{3.29}$$

the forward and backward recursions can be written as

$$\alpha_{k+1}(s_{k+1}) = \sum_{s_k}\sum_{c_k} \mathbb{P}(s_{k+1}|c_k, s_k) F_k(c_k, s_k)\alpha_k(s_k) \tag{3.30}$$

$$\beta_k(s_k) = \sum_{s_{k+1}}\sum_{c_k} \mathbb{P}(s_{k+1}|c_k, s_k) F_k(c_k, s_k)\beta_{k+1}(s_{k+1}) \tag{3.31}$$

while the completion yields

$$\mathbb{P}(c_k|\boldsymbol{y}_1^K) \propto \sum_{s_{k+1}} \mathbb{P}(c_k|s_{k+1})\alpha_{k+1}(s_{k+1})\beta_{k+1}(s_{k+1}) \tag{3.32}$$

where the PMF $\mathbb{P}(c_k|s_{k+1})$ is an indicator function, equal to one if c_k and s_{k+1} satisfy (3.18), and to zero otherwise. In this case, as explained in Section 2.1, the state metrics have the following probabilistic meanings

$$\alpha_k(s_k) \propto \mathbb{P}(s_k|\boldsymbol{y}_1^{k-1}) \tag{3.33}$$

$$\beta_k(s_k) \propto p(\boldsymbol{y}_k^K|s_k)\,. \tag{3.34}$$

A further equivalent formulation of the FBA can be derived by means of probabilistic arguments similar to those exploited in [25–27]. In particular, given the following alternative definitions of the branch metric and the

recursions

$$\tilde{F}_k(c_k, s_k) = \mathbb{P}(c_{k-L}) \exp\left[-\frac{1}{2N_0}\left|y_k - \sum_{i=0}^{L} f_i c_{k-i}\right|^2\right] \tag{3.35}$$

$$\tilde{\alpha}_{k+1}(s_{k+1}) = \sum_{s_k}\sum_{c_k} \mathbb{P}(s_{k+1}|c_k, s_k)\tilde{F}_k(c_k, s_k)\tilde{\alpha}_k(s_k) \tag{3.36}$$

$$\tilde{\beta}_k(s_k) = \sum_{s_{k+1}}\sum_{c_k} \mathbb{P}(s_{k+1}|c_k, s_k)\tilde{F}_k(c_k, s_k)\tilde{\beta}_{k+1}(s_{k+1}) \,. \tag{3.37}$$

the completion yields

$$\mathbb{P}(c_k|\boldsymbol{y}_1^K) \propto \sum_{s_{k+1}} \mathbb{P}(c_k|s_{k+1})\tilde{\alpha}_{k+1}(s_{k+1})\tilde{\beta}_{k+1}(s_{k+1}) \,. \tag{3.38}$$

It is straightforward to prove that, in this case, the state metrics have the following probabilistic meanings

$$\tilde{\alpha}_k(s_k) \propto p(\boldsymbol{y}_1^{k-1}|s_k) \tag{3.39}$$

$$\tilde{\beta}_k(s_k) \propto \mathbb{P}(s_k|\boldsymbol{y}_k^K) \tag{3.40}$$

which also clarify how to initialize the recursions. By comparing (3.36) with (3.30), (3.37) with (3.31), and (3.38) with (3.32), it is clear that both formulations have exactly the same complexity, since the only difference is the presence of the term $\mathbb{P}(c_{k-L})$ instead of $\mathbb{P}(c_k)$ in the branch metrics.

The reduced-complexity algorithms described in the following exploit the probabilistic meanings of the state metrics, thus it is worth to remark them. Let us focus on (3.33) and (3.34): the maximization of the metric $\alpha_k(s_k)$ provides the selection, based on the first $k-1$ received samples, of the state s_k according to the MAP criterion, while the maximization of the metric $\beta_k(s_k)$ provides the selection, based on the remaining received samples, of the state s_k according to the ML criterion. On the other hand, equations (3.39) and (3.40) show that maximizing $\tilde{\alpha}_k(s_k)$ provides a ML-based selection, while maximizing $\tilde{\beta}_k(s_k)$ provides a MAP-based selection. This reverse symmetry of the probabilistic interpretations motivates the choice for referring, hereafter, to the former formulation as *classical* and to the latter as *reverse*.

3.4.1 Description of the Algorithms

To obtain reduced-complexity algorithms, we adopt the *reduced search* technique [58]: the algorithms still work on a full-state trellis, that is no state

reduction [56] is performed, but they explore only a subset of the possible paths on the trellis. Since the lower the metric (either forward or backward) of a state, the more negligible its contribution to the summations of the recursions and completion, it is natural to explore only the paths extending from the states with the largest metrics. Let us suppose to keep memory, at each time epoch, only of the S largest forward and the S largest backward metrics, and to explore only the paths extending from the related states while performing the recursions. In the state metrics computation, the contribution corresponding to unexplored paths is considered null. If we consider the number S of saved metrics as a measure of the complexity of the algorithms[5], the reduction factor with respect to the full-complexity algorithm is about M^L/S.

By connecting the states whose forward metrics have been saved, a set of forward-selected-paths (FSPs) is progressively built, and the same can be done in order to define a set of backward-selected-paths (BSPs). As shown before, the selection of the most promising paths can follow either a MAP or a ML criterion, depending on which formulation of the recursions is chosen between the classical and the reverse ones. These different criteria significantly impact on the performance of the reduced-search algorithms since they generate potentially different sets of selected paths. It is intuitive to conjecture that the MAP approach is to be preferred, and extensive computer simulations confirm this fact. In order to combine the recursions providing MAP measures, that is the classical forward recursion and the reverse backward recursion, the following completion can be adopted

$$\mathbb{P}(c_k|\boldsymbol{y}_1^K) \propto \sum_{s_{k+1}} \mathbb{P}(c_k|s_{k+1})\alpha_{k+1}(s_{k+1})\tilde{\beta}_{k+1}(s_{k+1})\frac{1}{\mathbb{P}(s_{k+1})} \ . \tag{3.41}$$

In this case, the computational complexity slightly increases with respect to traditional formulations as (3.32) or (3.38), because of the presence of the term $1/\mathbb{P}(s_{k+1})$ in the summations.

The values of the forward (respectively, backward) metrics related to states that do not belong to the set of FSPs (respectively, BSPs) are not available during the completion stage. The unavailable metrics are often replaced by zero, thus neglecting the terms containing them while computing the sum-

[5] Although a more rigorous evaluation of the complexity would take into account also the additional computational load in implementing the various algorithms, the adopted measure is the most common in the literature. More detailed complexity comparisons will be presented later.

mations. In this case, it is easy to prove that any state can give a non-zero contribution to the completion only if it belongs to both FSPs and BSPs, so that the effective trellis is given by the intersection of FSPs and BSPs. This heavily degrades the performance of the reduced-complexity algorithms when the sets of FSPs and BSPs are built independently of each other, since their intersection could be almost empty. For this reason, in [58] it is suggested not to build any set of BSPs and to perform the backward search over the set of FSPs, thus giving a predominant role to the forward recursion. In the following, we propose three different solutions, each of them resulting effective on a different channel type, among the minimum-phase, maximum-phase, and mixed-phase channels.

We recall that, in the case of minimum-phase channels, the channel energy is mainly located in the first coefficients. As a consequence of this fact, the estimates of the APPs provided by the forward recursion are much more reliable than those provided by the backward recursion [61]. Hence, relatively to minimum-phase channels, the best choice is to perform the following sequence of steps:

1. run the classical forward recursion (3.30) building the set of FSPs;
2. run the classical backward recursion (3.31) over the set of FSPs only;
3. run the classical completion (3.32) over the set of FSPs only.

This solution, which will be referred to as *forward-trellis* (FT) algorithm, reduce to the M-BCJR algorithm [58] when none of the optimization techniques described in the next section is employed.

On the other hand, in the case of maximum-phase channels, the last coefficients contain the greatest part of the channel energy; consequently, the most reliable estimates of the APPs are provided by the backward recursion [61]. Thus, it is convenient to perform the following sequence of steps:

1. run the reverse backward recursion (3.37) building the set of BSPs;
2. run the reverse forward recursion (3.36) over the set of BSPs only;
3. run the reverse completion (3.38) over the set of BSPs only.

This solution will be referred to as *backward-trellis* (BT) algorithm.

Finally, in the case of mixed-phase channels, no physical reason to privilege one recursion instead of the other exists, so that the best choice is to build the

sets of FSPs and BSPs independently of each other. The resulting algorithm can be summarized by the following sequence of steps:

1. run the classical forward recursion (3.30) building the set of FSPs;
2. run the reverse backward recursion (3.37) building the set of BSPs;
3. run the completion (3.41) combining FSPs and BSPs.

In this case, the order of executing the recursions is irrelevant—the recursions could be also implemented in parallel. This solution will be referred to as *double-trellis* (DT) algorithm. From a computational viewpoint, this algorithm causes a slight increase in complexity, since it requires to compare and sort the metrics in both the recursions, while the FT and BT algorithms require this procedure only in one recursion.

Let us point out that any set of selected paths, either FSPs or BSPs, will be assumed properly initialized. Namely, the FSPs will always include the actual first state, while the BSPs will always include the actual last state. In practice, this requires the insertion of L pilot symbols that force the initial/final state.

3.4.2 Optimization of the Algorithms

As stated before, a traditional completion replacing the unavailable metrics by null values works on a subset of paths given by the intersection of FSPs and BSPs. While in the case of the FT (respectively, BT) algorithm the intersection coincides with the set of FSPs (respectively, BSPs), in the case of the DT algorithm the intersection could result almost empty, since the sets of FSPs and BSPs are built independently of each other. This issue is addressed in [60], where the authors propose a completion on a window of multiple trellis sections, thus implying a significant increase in the computational complexity of the completion stage. We propose a simpler solution allowing the completion stage to work not on the intersection but on the union of the sets of FSPs and BSPs, so that all metrics saved during the recursions can give a contribution to the final result. As also discussed in [61] and [67], an effective solution consists of replacing the unavailable metrics in (3.41) by proper non-zero values. At each time epoch k, let α_k^{MIN} be the lowest metric saved during the classical forward recursion, and let $\tilde{\beta}_k^{\mathrm{MIN}}$ be the lowest metric saved during the reverse backward recursion. When a given state s_{k+1} does not belong to the set of FSPs at time epoch $k+1$, any non-zero value lower or equal to $\alpha_{k+1}^{\mathrm{MIN}}$ could

be a reasonable choice for replacing the unavailable metric $\alpha_{k+1}(s_{k+1})$ while performing the completion (3.41). Since we found, by means of extensive computer simulations, that overestimating the unavailable metrics provides a better performance than underestimating them, we choose the largest value in the allowed range. Hence, when the factor $\alpha_{k+1}(s_{k+1})$ in (3.41) is not available, we replace it by $\alpha_{k+1}^{\mathrm{MIN}}$. Similarly, when $\tilde{\beta}_{k+1}(s_{k+1})$ is not available, we replace it by $\tilde{\beta}_{k+1}^{\mathrm{MIN}}$. When both the factors are not available, we instead ignore the whole product $\alpha_{k+1}(s_{k+1})\tilde{\beta}_{k+1}(s_{k+1})$. In Section 3.4.3, it is shown that the increase in computational complexity due to this solution, which will be referred to as *non-zero* (NZ) completion, is not crucial.

Typically, even when they provide a hard-output almost equal to the optimal one, the reduced-complexity algorithms estimate their hard decisions much more reliable than they really are. Extensive computer simulations show that even the proposed algorithms, especially the FT and BT ones, tend to overestimate the reliability of the decisions when the reduction factor M^L/S is large. Unfortunately, the fact of producing optimistic soft-output dramatically affects the convergence of turbo equalization. In [68], referring to the particular case of the soft-output Viterbi algorithm (SOVA) [69], the authors conjecture that the low-quality soft-output is due to the correlation between the intrinsic information $\{\mathbb{P}(c_k)\}$ and the extrinsic information $\{O_k(c_k)\}$ generated by the SOVA, and propose to mitigate this effect by passing the extrinsic information through an adaptive attenuator. We adopt a much simpler solution that consists of saturating, at each time epoch k, the extrinsic information such that the ratio between the lowest extrinsic information and the largest one must be at least equal to γ_S, where γ_S is a proper parameter in the range $[0, 1]$. Hence, instead of the computed values $\{O_k(c_k)\}$, the modified values $\{\hat{O}_k(c_k)\}$ are fed to the SISO decoder, according to the following definition

$$\hat{O}_k(c_k) = \max\{O_k(c_k), O_k^{\mathrm{S}}\} \tag{3.42}$$

where, at each time epoch k, the threshold value O_k^{S} is computed as

$$O_k^{\mathrm{S}} = \gamma_S \cdot \max_{c_k}\{O_k(c_k)\} \ . \tag{3.43}$$

The crucial point is the choice of the value of γ_S: when it is too low, the saturation is basically useless, whereas, when it is too large, the information produced by the detection algorithm is practically destroyed. The optimization of the parameter γ_S, which can be performed by means of computer

simulations, leads to values significantly depending on the SNR and the reduction factor of interest. This *output saturation* (OS) practically does not increase the complexity of the algorithms, but, as shown in Section 3.4.3, it can provide a significant performance improvement.

Finally, we describe a generalization of the reduced search technique based on a *state-partitioning* (SP) approach that stays in the middle between the rationales of the M-BCJR algorithm, the RS-BCJR algorithm, and the algorithms based on decision feedback sequence estimation, as that proposed in [67]. This approach ensures a better flexibility to different channel types and can provide significant performance improvements, especially in the case of non-binary modulation formats. While building the set of FSPs (that is only in the case of the FT and DT algorithms), let us define

$$s'_k = \boldsymbol{c}_{k-Q_f}^{k-1} \tag{3.44}$$

$$s''_k = \boldsymbol{c}_{k-L}^{k-Q_f-1} \tag{3.45}$$

Q_f being a design parameter, such that $s_k = (s'_k, s''_k)$. We will adopt the notation *state of the recent symbols* for s'_k and *state of the older symbols* for s''_k. The set of FSPs is built by saving, for each possible value of s'_k, only the states s''_k that give the S'' best metrics $\alpha_k(s_k)$. Hence, the saved metrics per time epoch are $S_f = M^{Q_f} S''$ and, unlike the case of the basic algorithms presented before, they are not necessarily the S_f highest ones. By expanding the factor $\mathbb{P}(s_k|\boldsymbol{y}_1^{k-1})$ in (3.33) as

$$\mathbb{P}(s_k|\boldsymbol{y}_1^{k-1}) = \mathbb{P}(s'_k|\boldsymbol{y}_1^{k-1})\mathbb{P}(s''_k|s'_k, \boldsymbol{y}_1^{k-1}) \tag{3.46}$$

the rationale of the proposed process can be understood: conditioning to each combination of the recent symbols s'_k, we save only the S'' highest APPs of the older symbols s''_k. Hence, the set of FSPs is defined by maintaining complete information on the state of the recent symbols and partial information on the state of the older symbols. While building the set of BSPs (that is only in the case of the BT and DT algorithms), we adopt a different partitioning, and define the state of the recent symbols s'_k and the state of the older symbols s''_k as

$$s'_k = \boldsymbol{c}_{k-L+Q_b}^{k-1} \tag{3.47}$$

$$s''_k = \boldsymbol{c}_{k-L}^{k-L+Q_b-1} \tag{3.48}$$

Q_b being a design parameter, such that $s_k = (s'_k, s''_k)$. Conditioning to each combination of the older symbols s''_k, we save only the S' highest APPs of the recent symbols s'_k, according to

$$\mathbb{P}(s_k|\boldsymbol{y}_k^K) = \mathbb{P}(s''_k|\boldsymbol{y}_k^K)\mathbb{P}(s'_k|s''_k, \boldsymbol{y}_k^K) \; . \tag{3.49}$$

The number of saved metrics per time epoch is thus $S_b = M^{Q_b}S'$. Symmetrically with respect to the forward recursion, the set of BSPs is defined by maintaining complete information on the state of the older symbols and partial information on the state of the recent symbols. We could not find a general rule to choose, for a fixed complexity, the values of the parameters Q_f, Q_b, S', and S'' providing the best performance, but they can be easily optimized by means of computer simulations. The basic algorithms presented before can be obtained by setting $Q_f = Q_b = 0$ and $S_f = S_b = S$.

3.4.3 Simulation Results

In this section, the performance of the proposed algorithms is assessed by means of computer simulations in terms of BER versus SNR. For each considered scenario, a comparison with the performance of the BCJR, M-BCJR, and RS-BCJR algorithms is given. To ensure better numerical stability, all algorithms are implemented in the logarithmic domain [28].

In Fig. 3.10, we consider a BPSK transmission over channel C5, represented according to the minimum-phase sequence $\boldsymbol{f}_0^6 = (7, 6, 5, 4, 3, 2, 1)/\sqrt{140}$. A $(3, 6)$-regular LDPC code of rate $1/2$ and codeword length of $4,000$ bits is used, and a detection instance is executed before each iteration of the SISO decoder, for a maximum of 40 iterations. The process also stops if, by checking the code syndrome, a valid codeword is found before the 40^{th} iteration. No interleaver is used because of the random nature of the LDPC code. As stated before, over this channel it is convenient to adopt the FT algorithm optimized by saturating the extrinsic information to a proper minimum value. Fig. 3.10 shows that the proposed algorithm, implementing the OS technique with $\gamma_S = 1/10$, outperforms both the M-BCJR and RS-BCJR.[6] We remark that the OS technique, which in this case is the only difference between the FT-OS and the M-BCJR algorithms, provides a gain of about 2.5 dB. An alternative way for improving the performance of the M-BCJR algorithm consists

[6] All reported values of γ_S were chosen according to optimizations performed by means of computer simulations.

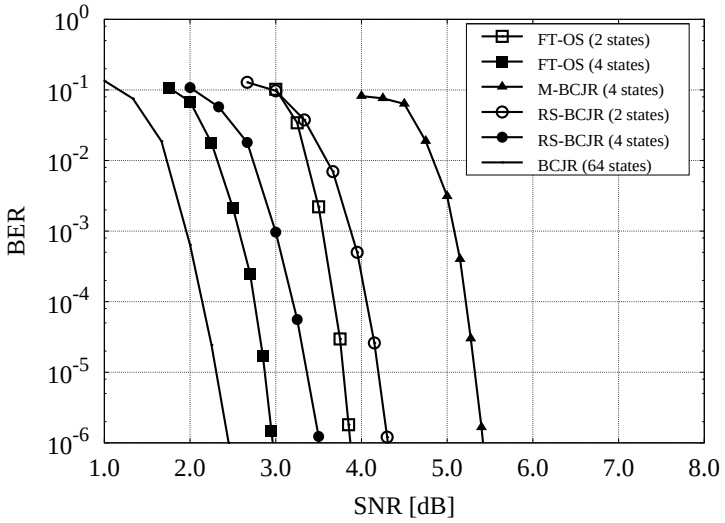

Figure 3.10: Performance of different algorithms over channel C5. A minimum-phase Forney representation and a coded BPSK modulation are adopted.

of performing more than one iteration of the LDPC decoder for each iteration of the detector. In this scenario, we found that the M-BCJR algorithm gains about 0.5 dB of power efficiency when at least three consecutive decoding iterations are performed. Hence, the OS technique is the most convenient solution, since it is both simpler (the saturation is definitely less complex than performing additional decoding iterations) and more effective of about 2.0 dB.

In Fig. 3.11, we consider the same channel, coding scheme, and modulation format as in the system of Fig. 3.10, but the channel is represented according to the maximum-phase sequence $\boldsymbol{f}_0^6 = (1, 2, 3, 4, 5, 6, 7)/\sqrt{140}$. The performance of the RS-BCJR and the M-BCJR algorithms, as expected on maximum-phase channels, is absolutely unacceptable. The performance of the BT* algorithm, which is identical to the BT algorithm with the exception that it builds the set of BSPs on the basis of the classical backward recursion instead of the reverse one, is also reported. Fig. 3.11 definitely proves that the proposed redefinition of the backward recursion ensures an impressive performance gain with respect to the traditional approach. A further gain of almost 1 dB is obtained by employing the OS technique with $\gamma_S = 1/25$.

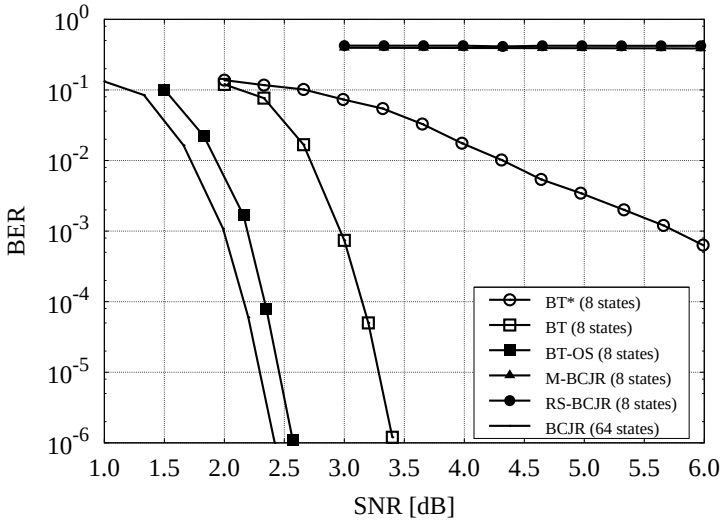

Figure 3.11: Performance of different algorithms over channel C5. A maximum-phase Forney representation and a coded BPSK modulation are adopted.

In Fig. 3.12, we consider channel C4, for which all Forney representations coincide with the sequence $\boldsymbol{f}_0^5 = (1,1,1,1,1,1)/\sqrt{6}$ and are mixed-phase (see the discussion at the end of Section 3.3.1). Again, the coding scheme and the modulation are the same adopted in the system of Fig. 3.10. With respect to the M-BCJR algorithm, the proposed DT algorithm loses about 0.5 dB if implemented with no optimization technique, while it gains about 1.5 dB when the NZ technique is adopted. Fig. 3.12 also confirms the effectiveness of the OS technique: in the considered case, a further gain of about 0.5 dB is obtained by saturating the extrinsic information evaluated by the algorithm and setting $\gamma_S = 1/30$. The curve related to the DT* algorithm, which is identical to the DT algorithm with the exception that it builds the set of BSPs on the basis of the classical backward recursion instead of the reverse one, gives a further proof of the need for a MAP-based selection of the set of BSPs. The M-BCJR algorithm loses about 2 dB from the DT-NZ-OS algorithm, while the RS-BCJR algorithm is ineffective on this mixed-phase channel. All considered reduced-complexity algorithms work on 8 states per trellis epoch, thus providing a reduction factor roughly equal to 4 with respect

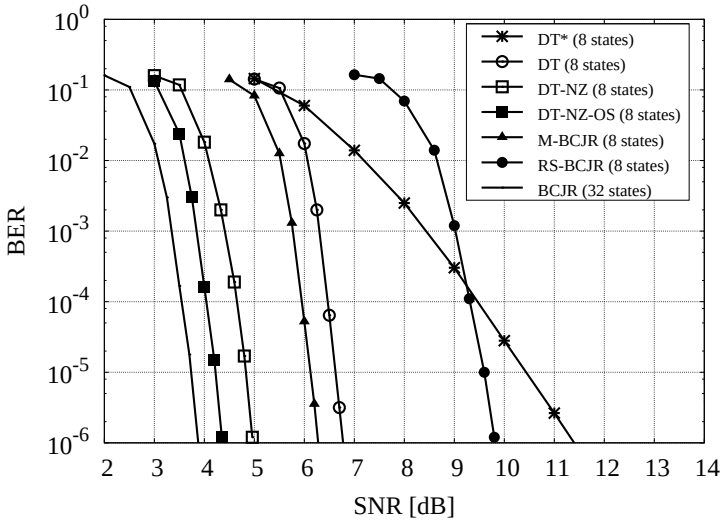

Figure 3.12: Performance of different algorithms over channel C4. A mixed-phase Forney representation and a coded BPSK modulation are adopted.

to the 32-state BCJR algorithm. More detailed measures of the reduction factors are given in Table 3.1 for each of the three stages of the algorithms, together with the overall reduction factors. These results refer to the number of non-trivial[7] multiply-accumulate operations only, while the operations of evaluating the branch metrics, sorting the state metrics, and saturating the output are not taken into account. In some cases, pointed out by means of an asterisk, the reduction factors cannot be evaluated in closed form and the reported results are obtained by means of computer simulations related to a value of the SNR that ensures, for the corresponding algorithm, a BER of about 10^{-6}. As expected, the M-BCJR algorithm provides the highest reduction factor since the backward recursion is forced to explore only the set of FSPs. It is worth to remark that the slight decrease of the reduction factor due to the adoption of the non-zero completion is largely compensated by the provided performance gain. Similar results have been obtained for other ISI channels and are not reported here.

We also investigated if the implementation of reduced-complexity algo-

[7]The terms giving a null contribution to the summations because of unsatisfied indicator functions $\mathbb{P}(s_{k+1}|c_k, s_k)$ or $\mathbb{P}(c_k|s_{k+1})$ are not considered.

Algorithm	Forw. Rec.	Back. Rec.	Completion	Overall
DT	4	4	4.7*	4.2*
DT-NZ	4	4	3.5*	3.9*
DT-NZ-OS	4	4	3.5*	3.9*
M-BCJR	4	6.0*	4	4.6*
RS-BCJR	4	4	4	4

Table 3.1: Reduction factors with respect to the full-complexity BCJR algorithm, for the same scenario as in Fig. 3.12. The asterisks denote results obtained by means of simulations.

rithms requires a greater number of iterations to reach convergence with respect to the BCJR algorithm. For all scenarios discussed so far, we found that the mean number of iterations to reach convergence does not significantly depend on the detection algorithm when a target BER lower than 10^{-5} is considered—less than one additional iteration is required on average with respect to the optimal BCJR algorithm. Hence, the only performance degradation due to the implementation of reduced-complexity algorithms is the loss in terms of power efficiency shown in Fig. 3.10, Fig. 3.11, and Fig. 3.12.

In Fig. 3.13, we finally consider quaternary PSK (QPSK) transmissions over the same mixed-phase channel as in Fig. 3.12, so that the full-complexity BCJR algorithm works on a 1024-state trellis. A non-recursive convolutional code with rate 1/2, generators $(5, 7)_8$, and codewords of $2,000$ bits is used, followed by a random interleaver and a Gray mapper. A detection instance is executed before each iteration of the SISO decoder, for 20 times. The curve related to perfect equalization (AWGN curve) is also reported as reference. While the DT-NZ algorithm does not result particularly effective on this scenario, the SP technique, implemented by setting $Q_f = 1$ and $Q_b = 0$, ensures an astonishing performance gain with respect to the other reduced-complexity algorithms, when they all work on 16 states per trellis epoch. The resulting loss with respect to the BCJR algorithm is lower than 1 dB at a BER of 10^{-6}, with a complexity lower of nearly two orders of magnitude.

3.5 An Alternative Approach to SISO Detection

In this section, we introduce a different approach to SISO detection that provides an impressive performance/complexity tradeoff over several ISI channels.

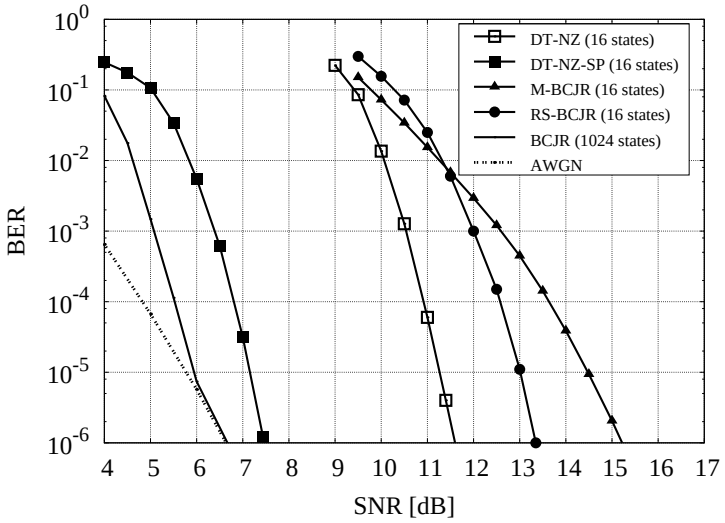

Figure 3.13: Performance of different algorithms over channel C4. A mixed-phase Forney representation and a coded QPSK modulation are adopted.

This approach, addressed to the Ungerboeck model, is based on the application of the SPA to a proper FG that contains cycles, as explained in the following.

3.5.1 Graph-Based Derivation of the Algorithms

In the Ungerboeck model, the conditional PDF characterizing the received sequence $\boldsymbol{x}_1^K$ satisfies the equation [46, 51]

$$p(\boldsymbol{x}_1^K|\boldsymbol{c}_{1-L}^K) \propto \prod_{k=1}^{K} \exp\left[\frac{1}{N_0}\mathrm{Re}\left\{x_k c_k^* - \frac{1}{2}g_0|c_k|^2 - \sum_{i=1}^{L} g_i c_k^* c_{k-i}\right\}\right] \tag{3.50}$$

where again the presence of L modulation symbols $\boldsymbol{c}_{1-L}^0$ that precede the symbols of interest is assumed to manage the border effects. If we define the functions

$$F_k(c_k) = \exp\left[\frac{1}{N_0}\mathrm{Re}\left\{x_k c_k^* - \frac{1}{2}g_0|c_k|^2\right\}\right] \tag{3.51}$$

$$H_i(c_k, c_{k-i}) = \exp\left[-\frac{1}{N_0}\mathrm{Re}\left\{g_i c_k^* c_{k-i}\right\}\right] \tag{3.52}$$

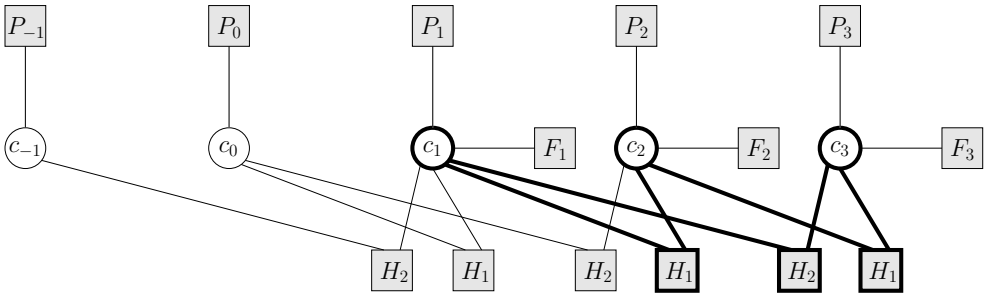

Figure 3.14: Factor graph corresponding to (3.54), when $K = 3$ and L=2. The bold edges, the bold variable nodes, and the bold function nodes mark a cycle of length six.

for each value of k in $\{1, 2, \ldots, K\}$ and each value of i in $\{1, 2, \ldots, L\}$, equation (3.50) yields

$$p(\boldsymbol{x}_1^K|\boldsymbol{c}_{1-L}^K) \propto \prod_{k=1}^{K} \left[F_k(c_k) \prod_{i=1}^{L} H_i(c_k, c_{k-i}) \right] . \tag{3.53}$$

Hence, we can factorize the APP of the transmitted sequence as

$$\begin{aligned} \mathbb{P}(\boldsymbol{c}_{1-L}^K|\boldsymbol{x}_1^K) &\propto \mathbb{P}(\boldsymbol{c}_{1-L}^0)\mathbb{P}(\boldsymbol{c}_1^K)p(\boldsymbol{x}_1^K|\boldsymbol{c}_{1-L}^K) \\ &\propto \mathbb{P}(\boldsymbol{c}_{1-L}^0) \prod_{k=1}^{K} \left[\mathbb{P}(c_k) F_k(c_k) \prod_{i=1}^{L} H_i(c_k, c_{k-i}) \right] \end{aligned} \tag{3.54}$$

where we exploited (1.2) and (3.53). The FG corresponding to (3.54) is depicted in Fig. 3.14, for the case $K = 3$ and L=2—as in the FGs reported in Chapter 2, the function nodes $\{P_k\}$ defined in (2.12) are adopted.

We point out that the marginalization of (3.54) required for computing the APPs of all modulation symbols, that is the target of SISO detection, cannot be *exactly* carried out by applying the SPA to the FG in Fig. 3.14, since it contains cycles [16]. On the other hand (let us quote [16]) "some of the most exciting applications of the sum-product algorithm—for example, the decoding of turbo codes or LDPC codes—arise precisely in situations in which the underlying factor graph *does* have cycles." In particular, provided that the length of the cycles is at least six, the SPA is generally expected to provide good approximations of the exact marginalizations (see [16] for the

general treatment, [19] for the case of LDPC codes, and [63] for an application to channels with sparse ISI). We notice that the FG in Fig. 3.14 has cycles of length six, as that bold-marked in the figure, but does not have any shorter cycle. Hence, the SPA can be confidently expected to effectively work. The interest for this suboptimal approach will appear clear later, when the related complexity will be discussed and compared with that of the optimal FBA.

For each value of k in $\{1, 2, \ldots, K\}$ and each value of i in $\{1, 2, \ldots, L\}$, we adopt the following notation for the various messages over the FG:

- $\mu_{F,k,i}(c_k)$ is the message from the variable node c_k that goes *forward* to the function node H_i connected also with the variable node c_{k+i};[8]
- $\mu_{B,k,i}(c_k)$ is the message on the same edge as $\mu_{F,k,i}(c_k)$ that goes *backward* to the variable node c_k;
- $\mu_{D,k,i}(c_k)$ is the message from the variable node c_k that goes *downward* to the function node H_i connected also with the variable node c_{k-i};
- $\mu_{U,k,i}(c_k)$ is the message on the same edge as $\mu_{D,k,i}(c_k)$ that goes *upward* to the variable node c_k.

Some example of such messages is given in Fig. 3.15 to clarify the conventions the concepts of forward, backward, downward, and upward are related to. We notice that the extrinsic information messages, namely the messages towards the function nodes $\{P_k\}$, are still denoted by $\{O_k\}$, as in the cycle-free FGs.

Once defined $V_k(c_k)$ as the product of all messages incoming to the variable node c_k, namely

$$V_k(c_k) = P_k(c_k)F_k(c_k)\left[\prod_{i=1}^{L}\mu_{U,k,i}(c_k)\right]\left[\prod_{i=1}^{L}\mu_{B,k,i}(c_k)\right] \tag{3.55}$$

[8] We point out that the FG in Fig. 3.14 does not include the variable node c_{k+i} when $k+i > K$. The formulations given hereafter are valid for edges that do not experience border effects, but can be easily extended.

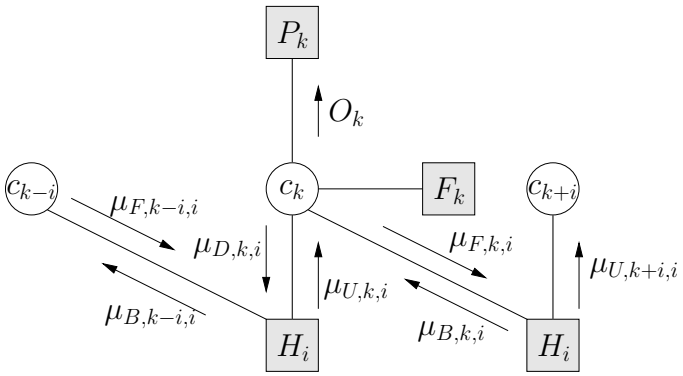

Figure 3.15: Part of the factor graph corresponding to (3.54) and notation adopted for the propagated messages.

the application of the SPA leads to the following rules for message updating:

$$O_k(c_k) = \frac{V_k(c_k)}{P_k(c_k)} \tag{3.56}$$

$$\mu_{F,k,i}(c_k) = \frac{V_k(c_k)}{\mu_{B,k,i}(c_k)} \tag{3.57}$$

$$\mu_{D,k,i}(c_k) = \frac{V_k(c_k)}{\mu_{U,k,i}(c_k)} \tag{3.58}$$

$$\mu_{U,k+i,i}(c_{k+i}) = \sum_{c_k} H_i(c_{k+i}, c_k)\mu_{F,k,i}(c_k) \tag{3.59}$$

$$\mu_{B,k-i,i}(c_{k-i}) = \sum_{c_k} H_i(c_k, c_{k-i})\mu_{D,k,i}(c_k)\ . \tag{3.60}$$

We point out that all messages $\{\mu_{X,k,i}\}$, with $X \in \{F, B, D, U\}$, should be initialized to the same positive value—the choice of this value is irrelevant, since the information is carried by the ratios between the messages rather than their values [16]. Also, we point out the probabilistic meaning of the term $V_k(c_k)$, which is proportional to the (approximated) APP $\mathbb{P}(c_k|\boldsymbol{x}_1^K)$.

Finally, we remark the main feature of the considered approach: the concept of state variable no longer appears, so that all messages over the FG are M-ary. (Let us recall that the FG leading to the FBA includes state variables for which M^L-ary messages should be propagated.) Moreover, we notice that the *exponential* dependence on the value of L does not appear neither in (3.55),

nor in the multiply-accumulate operations in (3.59) and (3.60). Hence, we can state that the complexity of the proposed approach increases *linearly* with the value of L, since this value only impacts on the number of edges in the FG, while the function nodes $\{H_i\}$ are connected with two variable nodes only, irrespectively of the value of L. For comparison, we mention the approach presented in [63], where a FG that does not contain state variables is also considered. On the other hand, the algorithms in [63] still present an exponential dependence on the value of L in the multiply-accumulate operations executed for message updating, since there exist function nodes connected with L variable nodes. To our knowledge, the proposed one is thus the only approach leading to linear-complexity algorithms for SISO detection over ISI channels.

3.5.2 Description of the Algorithms

Due to the presence of cycles in the considered FG, several algorithms can be derived by selecting different schedules and different stopping criteria for message updating. Two possible algorithms are described in the following.

The former algorithm will be referred to as *parallel-schedule* SPA (PS-SPA). It basically exploits that the lower part of the FG in Fig. 3.14 is formally identical to the FGs describing the LDPC codes, and performs the schedule adopted for standard LDPC decoding [19]. The PS-SPA can be formalized by the following sequence of steps:

1. initialize all messages over the graph;
2. update all terms $\{V_k\}$;
3. update all messages $\{\mu_{D,k,i}\}$ and $\{\mu_{F,k,i}\}$;
4. update all messages $\{\mu_{U,k,i}\}$ and $\{\mu_{B,k,i}\}$;
5. if the stopping criterion is not satisfied go to step 3;
6. stop the algorithm and, if required for decoding purposes, output the messages $\{O_k\}$.

We will only consider stopping criteria based on the number of self-iterations, that is on the number of times step 5 is executed. We point out that all operations at the same step can be executed in parallel. Hence, no serial operation is required by the PS-SPA, whose latency thus does not depend on the value of K. This feature makes the PS-SPA very attractive.

The latter algorithm will be referred to as *serial-schedule* SPA (SS-SPA). It is inspired to *shuffled LDPC decoding* [70] and to one of the schedules in [63]. Let us define the *forward recursion* as the following sequence of steps:

1. set $k = 1$
2. update all messages $\{\mu_{U,k,i}\}$ incoming to the variable node c_k;
3. update the term V_k;
4. update all messages $\{\mu_{F,k,i}\}$ outgoing from the variable node c_k;
5. increase k by one;
6. if $k \leq K$ go to step 2;
7. stop the recursion.

Let us also define the *backward recursion* as the following sequence of steps:

1. set $k = K$
2. update all messages $\{\mu_{B,k,i}\}$ incoming to the variable node c_k;
3. update the term V_k;
4. update all messages $\{\mu_{D,k,i}\}$ outgoing from the variable node c_k;
5. decrease k by one;
6. if $k \geq 1$ go to step 2;
7. stop the recursion.

Finally, the SS-SPA can be formalized by the following sequence of steps:

1. initialize all messages over the graph;
2. run the forward recursion;
3. run the backward recursion;
4. if the stopping criterion is not satisfied go to step 2;

5. stop the algorithm and, if required for decoding purposes, output the messages $\{O_k\}$.

Again, we will only consider stopping criteria based on the number of self-iterations, that is on the number of times step 4 is executed. We point out that step 2 and step 3 involve recursions to be serially executed, and thus the SS-SPA is characterized by a latency that increases with the value of K, like the optimal FBA but unlike the PS-SPA.[9]

We investigated other algorithms that was just found to provide improvements over particular channels, and are thus not described here.

3.5.3 Optimization of the Algorithms

Both PS-SPA and SS-SPA, when implemented in the basic version described above, provide a poor performance in most scenarios. After deep investigations on this behavior, we can state that the most significant problem is the overestimation of the reliability of the propagated messages—this is a known issue of the SPA when applied to FG with cycles [16,19]. A very simple way for overcoming this problem consists of adopting in (3.51) and (3.52) a value of N_0 larger than the actual one. The rationale of this trick is the following: since the problem is the overconfidence in the computed messages, we can make the algorithm less confident simply by describing the channel as if it added more noise than it really does. The effectiveness of this trick is proved by the EXIT charts reported in Fig. 3.16, which refer to BPSK transmissions over channel C3 when the value of the SNR is equal to 3 dB—see [44] for details on the EXIT charts related to detection algorithms, including the standard notation adopted also for labelling the axes in Fig. 3.16. The stopping criterion for the PS-SPA and the SS-SPA is set to one self-iteration only. We notice that, when the assumed value of N_0 is properly optimized, the EXIT charts of the PS-SPA and the SS-SPA significantly improve with respect to when the assumed value of N_0 equals the actual one—in particular, N_0 is increased by 9 dB for the PS-SPA and by 5 dB for the SS-SPA. Moreover, we notice that the SS-SPA is noticeably more effective than the PS-SPA in this scenario, and fairly close to the performance of the optimal BCJR algorithm.

[9] The latency can be decreased by executing step 2 and step 3 in parallel. On the other hand, we notice that this solution is not equivalent to the SS-SPA due to the presence of cycles in the FG.

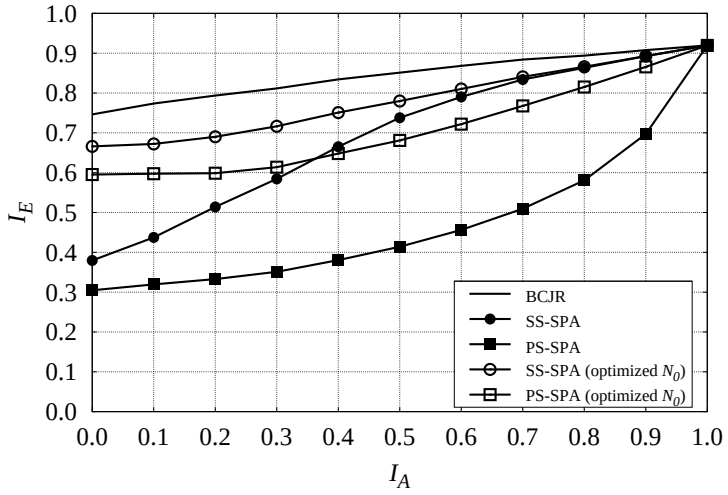

Figure 3.16: EXIT charts for different algorithms over channel C3, when a BPSK modulation is adopted and the value of the SNR is equal to 3 dB.

Other analyses carried out by means of EXIT charts show that, generally, it is not convenient to execute more than one self-iteration neither for the PS-SPA nor for the SS-SPA. Hence, when employed in iterative detection/decoding schemes, both algorithms are expected to provide the best performance when, each time they are provided with intrinsic information, they execute just one self-iteration before feeding out the extrinsic information. Since also extensive BER simulations confirm this behavior, the possibility of executing more self-iterations will be no longer exploited in this thesis. In iterative detection/decoding schemes, a further design option consists of resetting all messages in the detection FG each time the PS-SPA and the SS-SPA are provided with intrinsic information, so that any memory of the previous self-iterations is removed. These algorithms will be referred to as PS-SPA-R and SS-SPA-R.

Finally, we describe a technique for reducing the complexity of the PS-SPA and the SS-SPA. The idea is to neglect, in the Ungerboeck representation $\boldsymbol{g}_0^L$, the coefficients with the lowest magnitudes. Formally, we introduce the integer parameter L_E in $\{0, 1, \ldots, L\}$, and define a new FG that only includes the L_E function nodes $\{H_i\}$ related to the coefficients g_i with the largest magnitudes,

so that a reduction factor equal to about L/L_E results with respect to the original FG. Analyses based on EXIT charts, as well as BER curves, prove that the value of L_E can be much lower than the value of L without noticeably affecting the detection performance.

3.5.4 Simulation Results

In this section, the performance of the proposed algorithms is assessed by means of computer simulations in terms of BER versus SNR. All simulation results reported in the following are related to BPSK transmissions, and all algorithms are implemented in the logarithmic domain [28].[10] Moreover, the PS-SPA and the SS-SPA are implemented after a (coarse) optimization of the assumed values of N_0.

The results reported in Fig. 3.17 and Fig. 3.18 refer to BPSK transmissions over channels C2 and C3, respectively. An LDPC code with codeword length of $50,000$ bits and rate $1/2$ is used in both cases. A detection instance is executed before each iteration of the SISO decoder, for a maximum of 50 iterations. The process also stops if, by checking the code syndrome, a valid codeword is found before the 50^{th} iteration. No interleaver is used because of the random nature of the LDPC code. We point out that the simulation results related to the full-complexity BCJR algorithm are, in both cases, incomplete, since it is nearly unfeasible to obtain reliable BER curves for turbo equalizer working on a 4096-state trellis. Let us notice the impressive performance of the proposed algorithms over channel C2. First, we remark that there is no need for considering more than 5 function nodes $\{H_i\}$, since the SS-SPA practically performs as the BCJR algorithm when implemented with message resetting and $L_E = 5$. Then, we point out that the PS-SPA loses only few tenths of dB when implemented without message resetting and $L_E = 5$, and can be thus considered as the most convenient solution thanks to the low-latency properties discussed before. On the other hand, the proposed algorithms result less effective over channel C3, as expected according to the EXIT charts in Fig. 3.16. In particular, we notice that the SS-SPA cannot lose less than 0.5 dB with respect to the BCJR algorithm, but, since the related complexity is lower of at least two orders of magnitude, this is still a very satisfactory result—to our knowledge, over this magnetic channel, the proposed solution

[10] In particular, we adopted the log-likelihood ratio (LLR) domain. Hence, for the considered BPSK transmissions, all messages over the FG are scalar.

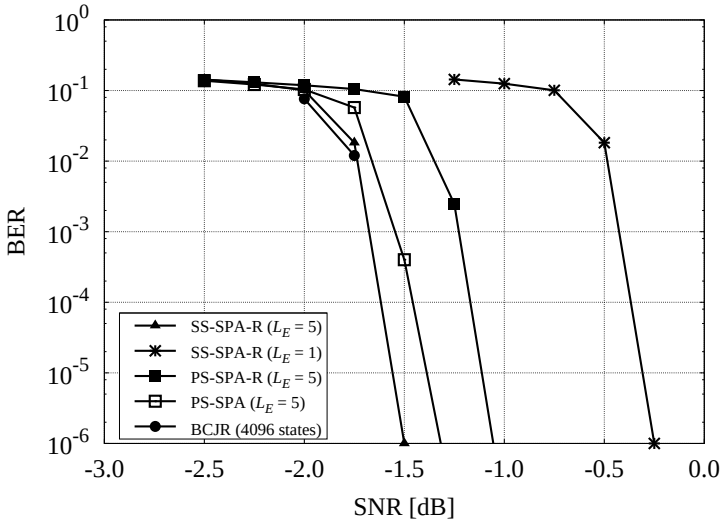

Figure 3.17: Performance of different algorithms over channel C2, when a coded BPSK modulation is adopted.

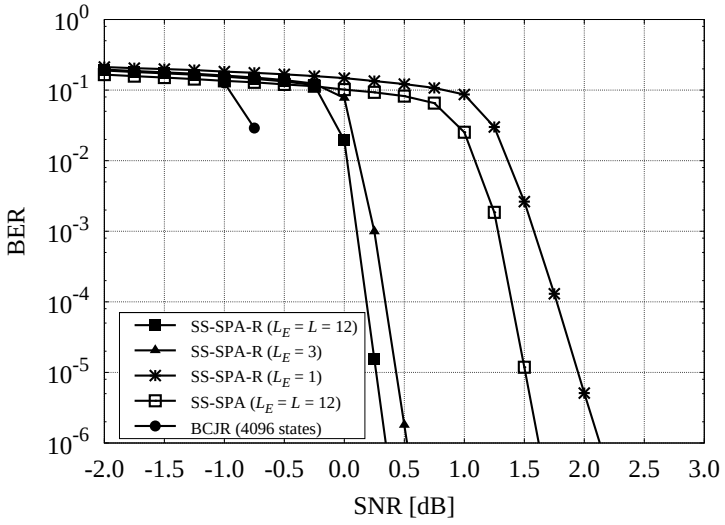

Figure 3.18: Performance of different algorithms over channel C3, when a coded BPSK modulation is adopted.

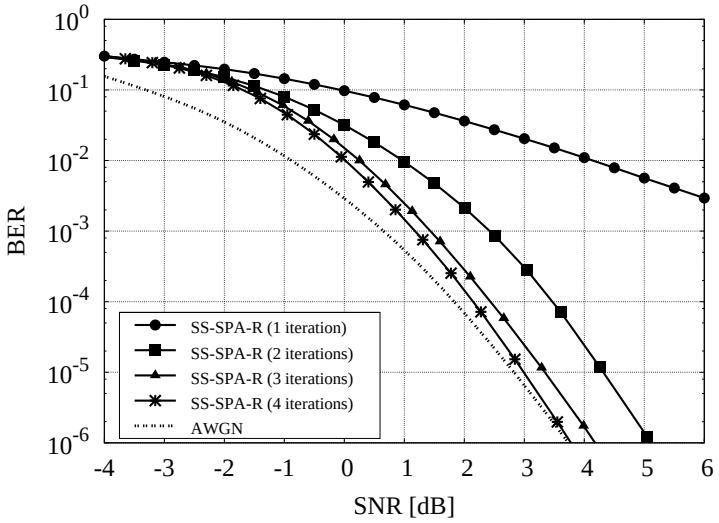

Figure 3.19: Performance of different algorithms over channel C3, when a coded BPSK modulation is adopted.

is by far that providing the best performance/complexity tradeoff. Fig. 3.17 and Fig. 3.18 also show the impact of the value of L_E and message resetting in this scenario. Interestingly, we can state that the SS-SPA performs better when message resetting is implemented, while the PS-SPA does when message resetting is not implemented.

Finally, in Fig. 3.19, we report the performance of the SS-SPA, implemented with message resetting and $L_E = 7$, when a weaker code is adopted, namely a non-recursive convolutional code with rate $1/2$, generators $(5, 7)_8$, and codewords of $2,000$ bits. A detection instance is executed before each iteration of the SISO decoder. The curve related to perfect equalization (AWGN curve) is also reported as reference. Again, channel C3 is considered. We point out that the proposed algorithm converges to perfect equalization after 4 iterations with the SISO decoder. Hence, when simple codes are used, there is no need for employing detection algorithms more complex than the SS-SPA even over this critical channel.

Chapter 4

Channels Impaired by Impulse Noise

This chapter addresses the problem of providing reliable communications over channels impaired by impulse noise. In particular, Section 4.1 introduces a two-state Markovian model that effectively describes the typical bursty nature of the impulse noise, while Section 4.2 presents and discusses the ultimate performance limits imposed by the considered model. Then, Section 4.3 describes a couple of practical communication schemes able to approach such limits, and Section 4.4 addresses the design of robust receivers that do not require any information on the statistical properties of the channel.

4.1 Channel Model

Several communication channels are often characterized by additive interferences that cannot be properly described by means of the classical AWGN model [20]. In these cases, the received signal can still be written as in (3.3), that is

$$r(t) = \sum_k c_k h(t - kT) + w(t) \tag{4.1}$$

but the noise process $w(t)$ should no longer be modeled as AWGN. In this chapter, to focus on the effects of the particular noise model presented later, we assume that the shaping pulse $h(t)$ is such that no ISI arises when the same frontend block described in Section 3.1 is adopted. Hence, the discrete-time

channel model equivalent to (4.1) yields

$$y_k = c_k + n_k \tag{4.2}$$

where $\{n_k\}$ are noise samples whose statistical features depend on the specific communication system. We here consider a particular kind of electromagnetic interferences that exhibit a significant impulsive nature, and often affect the communications over power line networks and some mobile radio scenarios [71]. These phenomena, referred to as *impulse noise*, are generally described by means of the class-A model [72] or the Bernoulli-Gaussian model [73], which both provide a good tradeoff between mathematical simplicity and fitting of the physical phenomenon. On the other hand, the main limit of such models is that, assuming a memoryless channel, they cannot properly describe one of the main features of the actual noise, namely the occurrence of bursts interfering on the received signal for several consecutive samples (see [71] and references therein). Hence, given that the actual channel is characterized by a significant amount of memory, it is interesting to evaluate which performance gain can be achieved when the memory is exploited in the system design, in terms of ultimate theoretical limits as well as performance of practical systems. To address these issues, we follow the guidelines in [71] and consider a channel model which modifies the Bernoulli-Gaussian model [73] such that the channel state is, instead of a Bernoulli process, a two-state Markovian process—due to this analogy, we refer to the considered model, which we previously presented in [74–76], as *Markovian-Gaussian* channel. A two-state Markovian process indeed provides a simple and effective way for describing a bursty evolution of the channel state [30]. Formally, we assume that, at each time epoch k, the statistical properties of the sample n_k are completely defined by the channel state s_k, which belongs to the binary alphabet $\{G, B\}$—G for good channel and B for bad channel, according to the notation in [30]—and evolves according to the transition probabilities

$$P_{IJ} = \mathbb{P}(s_{k+1} = J | s_k = I) \quad , \quad I, J \in \{G, B\} \tag{4.3}$$

which here are all assumed non-zero. We also assume that n_k is a zero-mean circularly-symmetric Gaussian random variable with variance depending on s_k,

namely the PDFs of n_k conditioned to the values of s_k are

$$p(n_k|s_k = G) = \frac{1}{2\pi N_0} \exp\left\{-\frac{|n_k|^2}{2N_0}\right\} \tag{4.4}$$

$$p(n_k|s_k = B) = \frac{1}{2\pi R N_0} \exp\left\{-\frac{|n_k|^2}{2RN_0}\right\} \tag{4.5}$$

where the parameter $R \geq 1$ gives the ratio between the average noise power in the bad channel and that in the good channel. This model can effectively describe a channel that, when in the good state, is impaired only by background Gaussian noise with variance per component equal to N_0, whereas, when in the bad state, is impaired also by impulsive interferers [71,73]. We notice that the considered model includes also the classical AWGN channel, obtained by setting $R = 1$—in this case, there is no distinction between the good state and the bad state, thus the channel is memoryless.

It can be proved that the couple (P_{GB}, P_{BG}) suffices for a complete description of the state process [22]. The probabilities of being in a given state are

$$P_G = \mathbb{P}(s_k = G) = \frac{P_{BG}}{P_{GB} + P_{BG}} \tag{4.6}$$

$$P_B = \mathbb{P}(s_k = B) = \frac{P_{GB}}{P_{GB} + P_{BG}} \tag{4.7}$$

and the average number of consecutive samples of permanence in a given state are

$$T_G = \frac{1}{P_{GB}} \tag{4.8}$$

$$T_B = \frac{1}{P_{BG}} \tag{4.9}$$

for the good state and the bad state, respectively [22]. Equations (4.8) and (4.9) clarify how to derive the statistical parameters (P_{GB}, P_{BG}) from the knowledge of the physical parameters (T_G, T_B), which can be measured on field [71]. In all typical scenarios, the bad state is relatively infrequent, that is $P_G > P_B$ or, equivalently, $T_G > T_B$ [71]. Hence, we will assume that

$$P_B < \frac{1}{2}\,. \tag{4.10}$$

We point out that the considered Markovian-Gaussian model reduces to the Bernoulli-Gaussian model when the transition probabilities (4.3) depend on the arrival state only. In this case, the state process is memoryless and is completely described by the value of P_B. Let us define the parameter

$$\gamma = \frac{1}{P_{GB} + P_{BG}} \tag{4.11}$$

which, as explained later, allows to quantify the channel memory. Provided that $\gamma + P_B \geq 1$, the couple (P_B, γ) is in one-to-one correspondence with the couple (P_{GB}, P_{BG}), and thus suffices for describing the parameterization of the state process. Hence, with respect to the Bernoulli-Gaussian model fully described by P_B, the Markovian-Gaussian model is characterized by the further parameter γ, whose impact on the noise process, when all other channel parameters are kept constant, is shown in Fig. 4.1, and whose meaning is now discussed. First, it can be easily proved that the channel is memoryless if and only if $\gamma = 1$. Moreover, for any given value of P_B, the parameter γ equals the ratio between the actual value of T_I and the value of T_I as if the channel were memoryless, where I can be either G or B. Hence, according to the notation in [30], we can state that, when $\gamma > 1$, the channel has a persistent memory, that is the average permanence in a state is longer with respect to the memoryless case, whereas, when $\gamma < 1$, the channel has an oscillatory memory, that is the average permanence in a state is shorter with respect to the memoryless case.[1] Since we are interested in modeling interferers that occur in bursts, thus with persistent memory by physical nature, we will focus on scenarios where $\gamma \geq 1$. It is clear from Fig. 4.1 that, while the classical memoryless model ($\gamma = 1$) fails in describing the bursts typically affecting the actual channels, the Markovian model properly does it, allowing to set the average duration of such bursts simply by modifying the value of γ.

4.2 Ultimate Achievable Information Rate

The system described in Section 4.1 matches the general finite-state Markovian model given in Section 1.2, the terms required for characterizing the model

[1] In [30], the parameter $\mu = 1 - 1/\gamma$ is adopted for characterizing the channel memory. We prefer the parameter γ for its physical meaning described above.

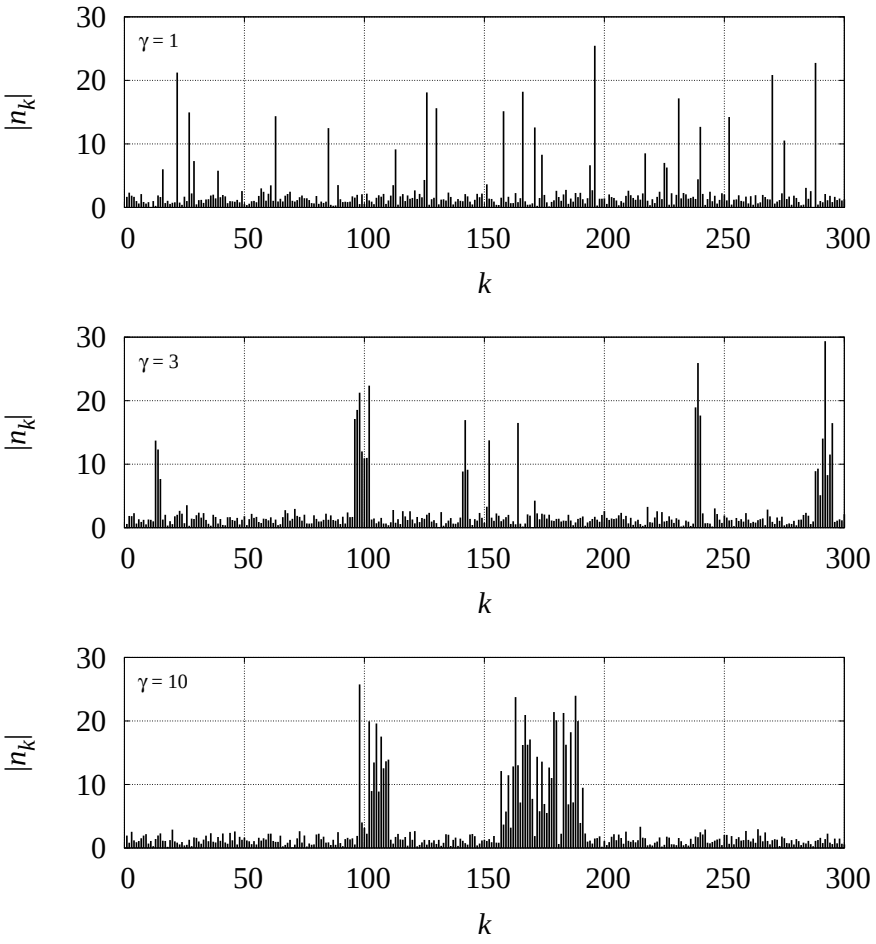

Figure 4.1: Magnitude of the noise process for three different values of γ, when the other channel parameters are $N_0 = 1$, $P_B = 10^{-1}$, and $R = 10^2$.

yielding

$$\mathbb{P}(s_1) = \begin{cases} P_G & \text{if } s_1 = G \\ P_B & \text{if } s_1 = B \end{cases} \tag{4.12}$$

$$p(y_k, s_{k+1}|c_k, s_k) = p(n_k = y_k - c_k|s_k)\mathbb{P}(s_{k+1}|s_k) \; . \tag{4.13}$$

Hence, the general frameworks introduced in Chapter 2 can be adopted for both analysis and design purposes. In particular, MAP symbol detection can be exactly carried out by means of the FBA recalled in Section 2.1. We notice that the Markovian-Gaussian channels include, as particular cases, two memoryless channels, namely the AWGN channel (obtained when $R = 1$) and the Bernoulli-Gaussian channel (obtained when $\gamma = 1$). In these cases, the FBA obviously degenerates into a symbol-by-symbol memoryless detector. Some results that show how the ultimate information rate achievable by MAP symbol detection varies with respect to the system parameters are reported and discussed in the following. Although the variance of the noise samples is

$$\mathbb{E}\{|n_k|^2\} = 2N_0P_G + 2N_0RP_B = 2N_0 + 2N_0(R-1)P_B \tag{4.14}$$

we will adopt the SNR definition

$$\text{SNR} = \frac{1}{2N_0}\mathbb{E}\{|c_k|^2\} \tag{4.15}$$

which accounts for the background Gaussian noise only—this choice allows to better realize the impact of the impulsive interferers, as clarified later. We point out that the state process underlying the channel is the same as in the Gilbert-Elliott model (see [30] and references therein), whose information rate can be analytically computed [30], but the same analytical arguments do not lead to a closed-form expression here, since the channel output is neither finite nor discrete [31]. Hence, the reported results were obtained by means of the simulation-based method described in Section 2.2. In all cases, matched decoding is considered, that is the channel model assumed by the receiver and the actual channel coincide. We will only investigate PSK modulations and quadrature amplitude modulations (QAMs). In these cases, by resorting to the arguments in [31] or simply by exploiting the symmetry of the channel and the modulation alphabet, it is easy to prove that a memoryless source that emits equally likely symbols achieves the maximum allowed information rate. Such a source is thus considered hereafter.

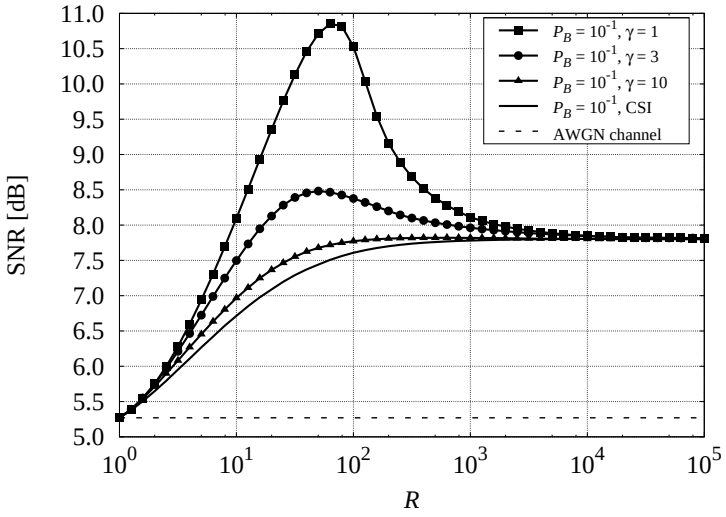

Figure 4.2: Signal-to-noise ratio required to achieve an information rate equal to 1.75 bits per channel use when a QPSK modulation is adopted.

In Fig. 4.2, it is shown how the value of the SNR required to achieve an information rate equal to 1.75 bits per channel use varies when different channels are considered and a QPSK modulation is adopted. Together with the curves related to Markovian-Gaussian channels characterized by $P_B = 10^{-1}$ and different values of γ and R, we also reported the corresponding curves related to a system with ideal channel-state information (CSI), that is a system that knows the actual realizations of the state process, and to an AWGN channel.[2] It is clear that, for a given value of R, the theoretical power efficiency monotonically improves as the memory increases, up to the value related to the system with CSI. This was expected, since the memory helps to track the state process, up to the limit given by the ideal CSI tracker. On the other hand, when the value of R is very large, all curves tend to the value related to the system with CSI, irrespectively of the amount of memory. This also was expected, since the stronger the impulsive interferers, the easier to detect them even when the state tracking is not helped by the memory. It is also interesting to notice that, for a given value of γ, the theoretical power efficiency exhibits

[2]The information rate of systems with CSI equals the statistical average of the information rates over the channel states [31]. In this case, the value of γ is obviously irrelevant.

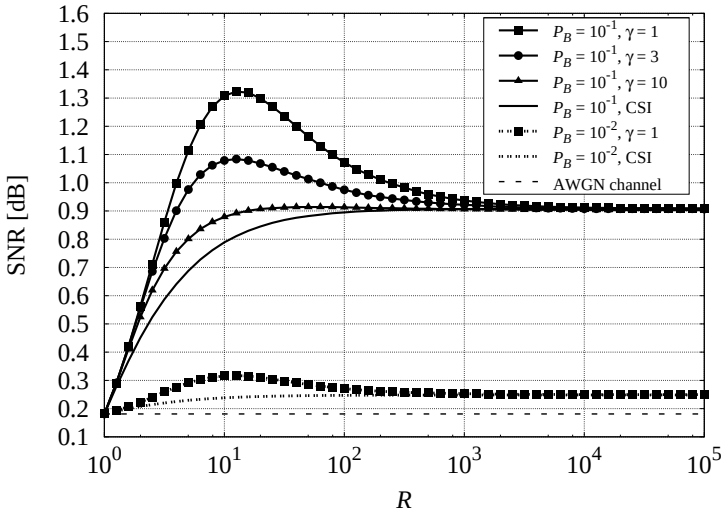

Figure 4.3: Signal-to-noise ratio required to achieve an information rate equal to one bit per channel use when a QPSK modulation is adopted.

a non-monotonic behavior with respect to R. In particular, there exists a threshold value for R depending on γ such that, above that value, the larger the value of R the better the performance. Such a behavior, which tends to vanish as the value of γ increases and vanishes at all for the system with CSI, is somehow surprising since it implies that the system can take advantage of a larger power of the impulsive interferers. The key point is, again, that the stronger the interferers, the easier to detect their presence irrespectively of the amount of memory.

Although similar considerations *qualitatively* hold for all values of P_B and all values of the target information rate, the performance gain provided by the presence of memory *quantitatively* depends on such values. This is proved by comparing the simulation results reported in Fig. 4.2 and those reported in Fig. 4.3, which refer to QPSK transmissions with a target information rate equal to one bit per channel use. Namely, in the former case the memory can provide performance gains in the order of 3 dB, while in the latter case the beneficial effect of the memory is very limited when $P_B = 10^{-1}$ and practically negligible when $P_B = 10^{-2}$. Finally, in Fig. 4.4, simulation results are reported which confirm the same behavior also for 16-QAM transmissions with a target

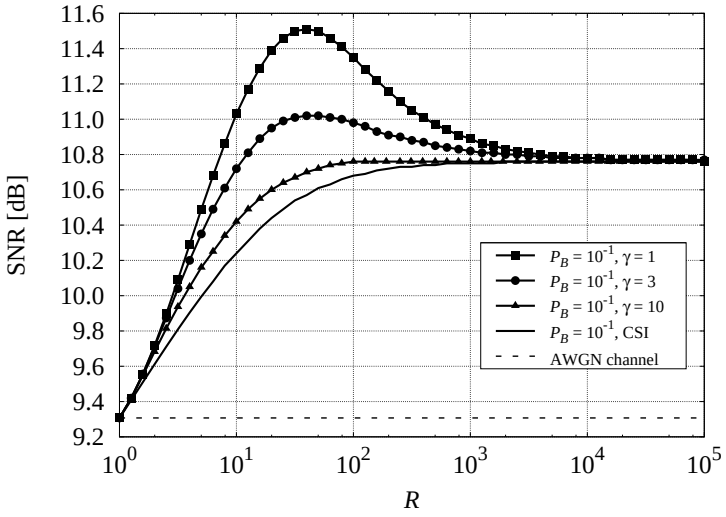

Figure 4.4: Signal-to-noise ratio required to achieve an information rate equal to 3 bits per channel use when a 16-QAM is adopted.

information rate equal to 3 bits per channel use.

A different viewpoint for characterizing the impact of the memory consists of evaluating the achievable information rate for different values of γ when all other channel parameters are fixed. In this case, the information rate increases with the value of γ, taking values in the range whose lower limit is given by the memoryless channel with $\gamma = 1$ and whose upper limit is given by the corresponding system with CSI. The span of such ranges is reported in Fig. 4.5 for QPSK transmissions and different channel parameterizations. These results definitely show that the impact of the memory on the achievable information rate is limited, and tends to vanish when the value of P_B is very low, that is when the presence of impulsive interferers is very infrequent. Hence, our analysis suggests that the presence of memory can provide a significant gain in terms of SNR when we fix the target information rate (see Fig. 4.2), but it does not provide a significant gain in terms of information rate when we fix the target SNR (see Fig. 4.5). The implications of this behavior are discussed in the following, relatively to the example of QPSK transmissions over a channel characterized by $P_B = 10^{-1}$, $R = 10^2$, and $\gamma = 10$, with a target information rate of 1.75 bits per channel use. If the communication system

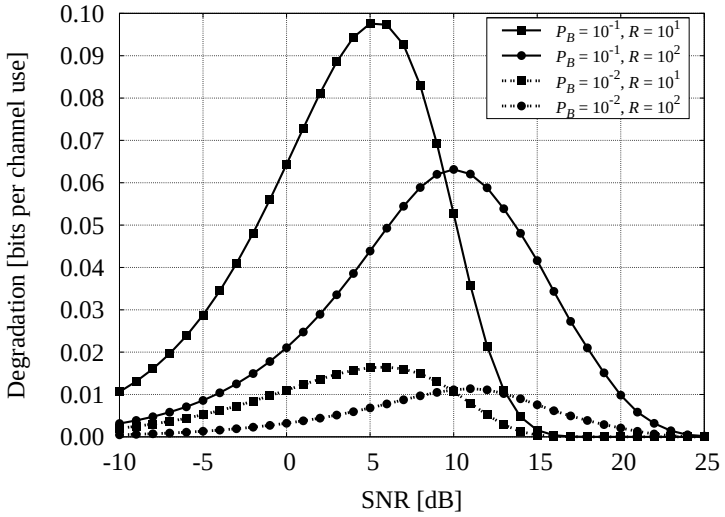

Figure 4.5: Degradation in information rate between systems with CSI and the corresponding channels with $\gamma = 1$ when a QPSK modulation is adopted.

is really constrained to guarantee the target information rate, the search for a scheme that can exploit the channel memory is strongly motivated, since it is possible to gain almost 3 dB according to Fig. 4.2. On the other hand, if the target information rate is not strictly mandatory, a simple alternative option consists of neglecting the channel memory.[3] In fact, without any penalty in terms of SNR, this option only implies a degradation of about 0.06 bits per channel use according to Fig. 4.5, and can be convenient thanks to its lower complexity—details on the complexity of the receivers able to achieve the reported theoretical limits are given in Section 4.3.

In conclusion, the impact of the channel memory on the ultimate performance limits greatly depends on the system requirements, namely on which is the main design target between the information rate and the power efficiency.

[3] In this case, irrespectively of the actual amount of channel memory, the ultimate performance limit is that related to a memoryless channel with all other parameters equal to the actual ones [30].

4.3 Practical Communication Schemes

We are interested in designing communication systems that, under the assumption of ideal knowledge of the noise parameters, can approach the ultimate performance limits evaluated in Section 4.2. Hence, we focus on systems that employ powerful channel codes, such as LDPC codes or turbo-like codes. According to the discussions carried out in Section 2.3, the iterative receivers based on the exchange of soft information between the SISO decoder, the SISO mapper/demapper, and the MAP SISO detector are expected to provide the best performance. In this section, such schemes are compared with simpler schemes, to find out those providing the most convenient performance/complexity tradeoff. The following treatment assumes that the channel has memory, that is $\gamma > 1$, since all channels of practical interest are such by physical nature of the bursty interferers.

First, we consider a scheme based on intentional mismatched decoding (see Section 2.2.1), where the channel memory is neglected. This approach, which is the most common in the literature, leads to a very simple receiver, since, once assumed the absence of memory (that is once assumed $\gamma = 1$), the FBA degenerates into a symbol-by-symbol algorithm that outputs, for each time epoch k and for each trial value of c_k, the extrinsic-information message

$$O_k(c_k) = p(n_k = y_k - c_k | s_k = G)P_G + p(n_k = y_k - c_k | s_k = B)P_B \; . \quad (4.16)$$

In this case, since the SISO detector does not exploit any *a priori* information on the modulation symbols according to (4.16), the detection algorithm can be executed only once per received frame. Hence, this approach provides great improvements in terms of latency with respect to the reference iterative scheme based on the FBA, since it does not involve recursions to be serially executed nor requires iterations with the SISO decoder. On the other hand, the information-theoretical results reported in Section 4.2 show that there exist various scenarios where the potential performance gain motivates the search for alternative schemes able to exploit the channel memory. A possible solution, aimed at reducing the latency that characterizes the reference scheme without neglecting the channel memory, consists of executing the FBA only once per received frame, without further iterations with the SISO decoder. Unfortunately, this approach only reduces the latency due to the iterations, but maintains the major cause of latency, that is the serial recursions. A detection scheme that drops even this cause of latency is proposed in the

following. The idea is basically to adopt a parallel message-passing schedule even if the FG underlying the detection problem would call for a serial schedule (see [77] for algorithms based on a similar rationale). Formally, for each time epoch k and for each state s_k, the forward metric $\alpha_k(s_k)$ is initialized as if the k-th step were the first in the graph, and the backward metric $\beta_k(s_k)$ is initialized as if the k-th step were the last in the graph, that is (see Section 2.1)

$$\alpha_k(s_k) = \mathbb{P}(s_k) \tag{4.17}$$
$$\beta_k(s_k) = 1 \tag{4.18}$$

and all steps of the recursions (2.4) and (2.5) are then executed simultaneously. The final completion is instead unchanged, since it does not involve any operation to be serially executed. Hence, any latency that depends on the length of the transmitted sequence is dropped out.

4.3.1 Simulation Results

The performance of the considered schemes was assessed by means of computer simulations in terms of BER versus SNR. Some of these results are reported in Fig. 4.6, for the case of Gray-mapped QPSK transmissions over a Markovian-Gaussian channel characterized by $P_B = 10^{-1}$, $\gamma = 10$, and $R = 10^2$. A rate-7/8 LDPC code with codeword length of $100,000$ bits is used, which we designed following the general framework in [44].[4] A detection instance is executed before each iteration of the SISO decoder when the FBA with iterative detection is employed, while is executed only once per received frame in all other cases. The process stops if, by checking the code syndrome, a valid codeword is found, or after 100 decoding iterations. No interleaver is used because of the random nature of the LDPC code. To ensure better numerical stability, all algorithms are implemented in the logarithmic domain [28]. We remark that the FBA with iterative detection provides a performance gain of more than 3 dB with respect to the conventional memoryless scheme. Interestingly, if the FBA is applied without iterative detection, the performance is practically the same as that of the iterative scheme. This fact is discussed later. Moreover, we point out that the parallel version of the FBA exhibits a degradation of about 1 dB with respect to the serial one, but still outper-

[4] An alternative design framework, specifically addressed to an ensemble of channels including the considered one, can be found in [42].

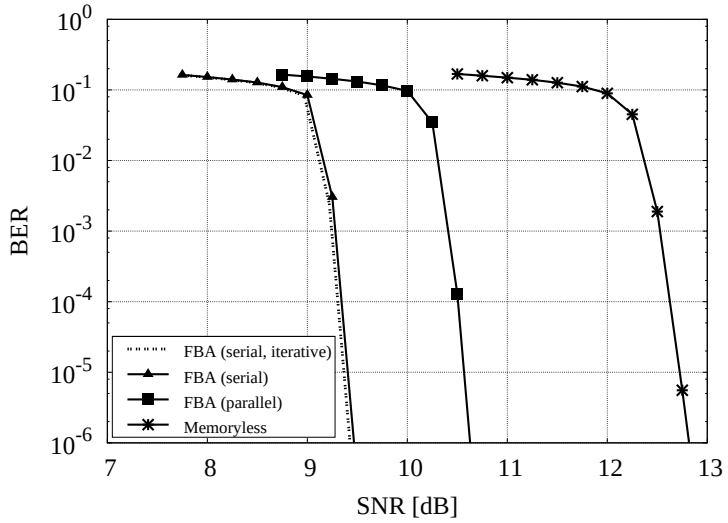

Figure 4.6: Performance of various detection schemes over a channel characterized by $P_B = 10^{-1}$, $\gamma = 10$, and $R = 10^2$. A system employing a rate-7/8 LDPC code and a QPSK modulation is considered.

forms the memoryless scheme of more than 2 dB, and thus results the most convenient solution when low-latency receivers are mandatory.

Other simulation results are reported in Fig. 4.7, for the case of Gray-mapped QPSK transmissions over a Markovian-Gaussian channel characterized by $P_B = 10^{-1}$, $\gamma = 10$, and $R = 10$. The coding scheme and the receivers are the same as in the previous case, but the code has rate 1/2. We point out that these results, yet qualitatively similar to those in Fig. 4.6, show much lower performance differences. This fact is in excellent agreement with the ultimate performance limits discussed in Section 4.2, as explained in the following.

We notice that the results in Fig. 4.6 can be directly compared with the ultimate limits reported in Fig. 4.2, since the use of a rate-7/8 binary code combined with a QPSK modulation implies an actual information rate equal to 1.75 bits per channel use. Similarly, the results in Fig. 4.7 can be directly compared with the ultimate limits reported in Fig. 4.3. In both cases, the gain predicted by the theoretical results for the case $\gamma = 10$ with respect to the memoryless case is very similar to that characterizing the considered practical

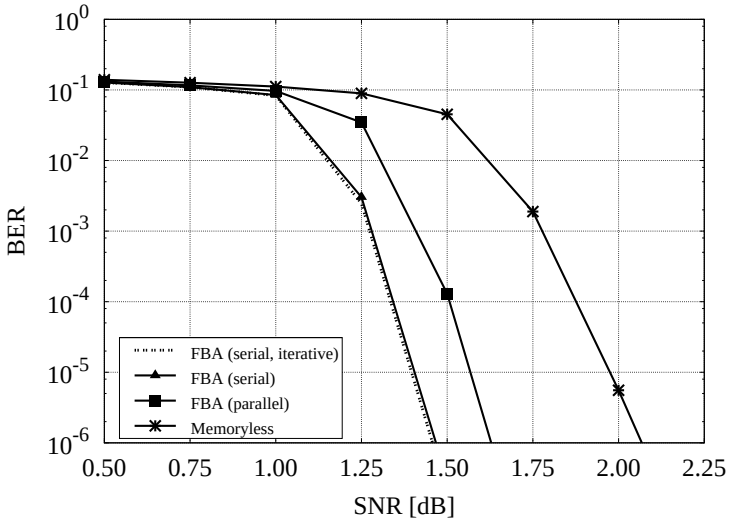

Figure 4.7: Performance of various detection schemes over a channel characterized by $P_B = 10^{-1}$, $\gamma = 10$, and $R = 10$. A system employing a rate-1/2 LDPC code and a QPSK modulation is considered.

schemes, namely about 3 dB for the former case and about 0.6 dB for the latter case. Hence, given the excellent agreement between the theoretical results and the performance of the practical systems, we can definitely confirm that there exist various scenarios where the conventional approach of neglecting the channel memory, which is actually present in any real channel due to the bursty nature of the interferers, can be significantly outperformed by resorting to the proposed schemes.

Finally, we discuss the fact that iterative detection does not significantly improve the performance of the FBA, in both scenarios considered above. This suggests that the FBA cannot take advantage of exploiting the soft information produced by the LDPC decoder. It is interesting to investigate if this behavior is related to particular scenarios or if it is a general property of the algorithm, irrespectively of the noise parameterization. A simple way for characterizing how much a detection algorithm can take advantage of iterating with the LDPC decoder consists of evaluating its EXIT chart [44]. Based on the framework in [44], we carried out extensive simulations for various parameterizations of the Markovian-Gaussian channels, and found that, in all of

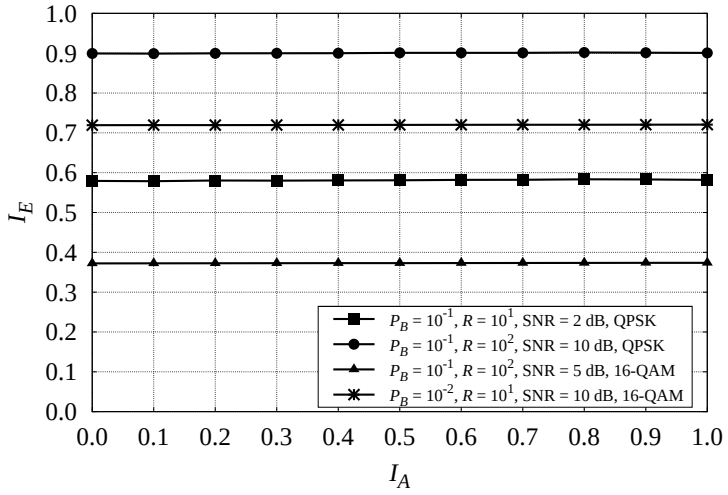

Figure 4.8: EXIT charts related to the FBA for different channel parameterizations. In all cases, we set $\gamma = 10$.

them, the detection algorithm is expected not to significantly gain from iterating with the LDPC decoder—in the context of [44], it means that the EXIT chart related to the detector is nearly flat. Some of these EXIT charts are reported in Fig. 4.8 for different channel parameterizations and modulation formats. Let us notice their flatness, which definitely proves that there is no need for iterative detection.

4.4 Mismatched Decoding

So far, matched decoding has been assumed for both information-theoretical analyses and design of practical systems. Namely, we have assumed that the receiver is provided with exact information on the statistical properties of the channel. In the remainder of this chapter, we investigate the problem of providing reliable communications over channels impaired by impulse noise in conditions of mismatched decoding (see Section 2.2.1). We presented some of the results that follow in [78–80].

According to the model given in Section 4.1, four parameters are required for completely describing the channel, namely N_0, P_B, R, and γ. These pa-

rameters are actually unknown to the receivers, and thus the results obtained under the assumption of matched decoding should be thought of as (optimistic) reference benchmarks. A natural approach consists of estimating the noise parameters (see [71,81,82] for possible estimation algorithms), and running the detection algorithm as if the estimated values were exact. Unfortunately, the estimation algorithms, beside increasing the complexity of the receiver, cannot properly cope with channels whose statistics are significantly time-varying, as most channels of practical interest [71,81–83]. Hence, a blind approach, based on detection algorithms that do not require the knowledge of the statistical properties of the impulse noise nor their estimation, is of great interest. Possible solutions are presented and compared in the following. Since the estimation of the power of the background Gaussian noise is not critical and since our aim is to focus on the impact of the impulsive interferers, a perfect knowledge of the value of N_0 is assumed hereafter, as commonly done in the literature. Also, for simplicity, we will no longer exploit the memory characterizing the bursty impulse noise, so that the first cause of mismatch is the assumption that the channel is memoryless (that is $\gamma = 1$)—let us recall that the impact of this mismatch has been already investigated in Section 4.2 and Section 4.3. Under these assumptions, the optimal detection scheme works symbol-by-symbol and outputs, for each time epoch k and each trial value of c_k, the extrinsic-information message (hereafter referred to simply as metric)

$$O_k(c_k) = \exp\left\{-\frac{|y_k - c_k|^2}{2N_0}\right\} + \frac{P_B}{(1-P_B)R}\exp\left\{-\frac{|y_k - c_k|^2}{2RN_0}\right\} \tag{4.19}$$

which is easily derived from (4.4), (4.5), and (4.16) by dropping irrelevant factors. We point out that the first term in the right-hand side of (4.19) is exactly the optimal metric for AWGN channels, that is

$$O_k^{\mathrm{AWGN}}(c_k) = \exp\left\{-\frac{|y_k - c_k|^2}{2N_0}\right\} \tag{4.20}$$

whereas the second term can be seen as a correction term that accounts for the presence of impulse noise. In the next section, alternative metrics that do not require an explicit estimation of the values of P_B and R are presented.

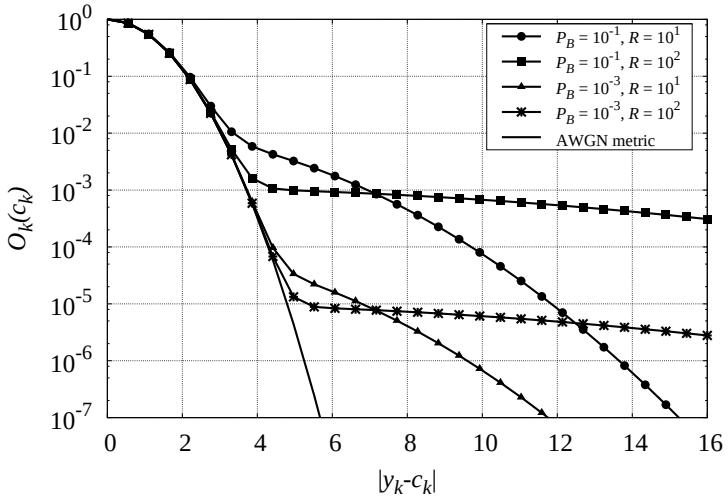

Figure 4.9: Optimal metric for different parameterizations of the impulse noise, normalized according to $N_0 = 1$.

4.4.1 Robust Detection Metrics

The behavior of the optimal metric (4.19) is depicted in Fig. 4.9 for different parameterizations of the impulse noise. We point out that all curves, irrespectively of the values of P_B and R, match the AWGN metric at the left-hand side of the figure, and thus that the knowledge of the statistics of the impulse noise is required only for describing the "tails" of the metric—this is due to the fact that, under the assumption (4.10), the first term in the right-hand side of (4.19) is dominant when the distance $|y_k - c_k|$ is low. This particular behavior can be exploited for deriving detection metrics that do not require an explicit estimation of the values of P_B and R.

First, we mention the simplest solution, which consists of neglecting the presence of impulse noise at all. In this case, the detection metric simply equals the AWGN metric.

Then, we mention the classical *soft limiting* (SL) metric [84]

$$O_k^{\mathrm{SL}}(c_k) = \exp\left\{-\frac{|z_k - c_k|^2}{2N_0}\right\} \tag{4.21}$$

where, at each time epoch k, the sample z_k is obtained by properly cutting

the amplitude of the received sample y_k. Formally, given a suitable threshold value $V_T > 0$, we can write

$$z_k = f(\mathrm{Re}\{y_k\}) + jf(\mathrm{Im}\{y_k\}) \tag{4.22}$$

where the non-linear function $f(\cdot)$ is defined as

$$f(x) = \begin{cases} V_T & \text{if } x > V_T \\ -V_T & \text{if } x < -V_T \\ x & \text{otherwise} \end{cases} . \tag{4.23}$$

The rationale of the SL metric is the following: since the AWGN metric is a very good approximation of the optimal metric over a suitable range, we can just adopt the AWGN metric after cutting the received sample such that it belongs to this suitable range.

We propose the following threshold approximation of the optimal metric

$$O_k(c_k) = \max\left\{O_k^{\mathrm{AWGN}}(c_k), \Delta\right\} \tag{4.24}$$

where $\Delta \in [0, 1]$ is a design parameter discussed later. The rationale of the approximation (4.24), which is just a saturation of the AWGN metric (not of the received sample, unlike the SL metric) to a constant threshold and thus is very simple from a computational viewpoint, is explained in the following. First, when the distance $|y_k - c_k|$ is low, the proposed metric matches the AWGN metric, as an effective metric should definitely do according to Fig. 4.9. On the other hand, when the received sample is far away from the constellation, that is when the distance $|y_k - c_k|$ is very large for all possible values of the modulation symbol c_k, the proposed metric is saturated to the minimum threshold Δ irrespectively of the value of c_k, so that the detector produces the so-called *erasure* decision. In this case, the presence of impulsive contributions in addition to the background noise is very likely, and a receiver that does not know the statistics of the impulse noise cannot produce any more reliable decision than an erasure.

Other metrics are presented in [83], but they are not considered here since they are not suitable for receivers employing iterative decoding [78]. On the other hand, we notice that the metrics in [83] are effective when simpler coding schemes, such as convolutional codes, are considered [78].

To compare the considered metrics, it is useful to consider the results reported in Fig. 4.10, which refer to a BPSK modulation with alphabet $\{1, -1\}$

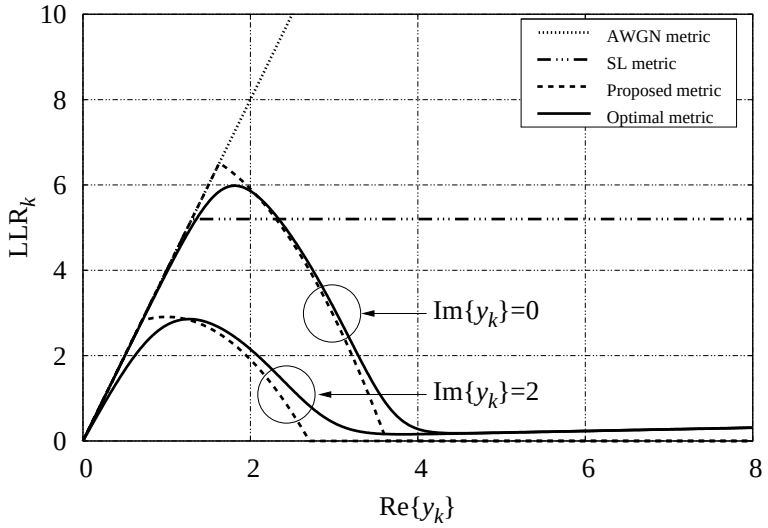

Figure 4.10: Log-likelihood ratios produced by different detection metrics over a channel characterized by $P_B = 10^{-1}$, $R = 10^2$, and SNR = 0 dB.

and a channel characterized by $P_B = 10^{-1}$, $R = 10^2$, and SNR = 0 dB. In Fig. 4.10, it is shown how the LLR, that is the natural logarithm of the ratio between the metric corresponding to the hypothesis $c_k = 1$ and that corresponding to the hypothesis $c_k = -1$, varies with respect to the received sample y_k when different metrics are considered. The behavior of the optimal metric exhibits two key points, namely the need for producing low-magnitude LLRs when the received sample is far away from the constellation, that is when the presence of interfering impulses is very likely, and the need for exploiting the imaginary component of y_k—unlike the AWGN channels, the real and imaginary components of the impulse noise are not independent. Fig. 4.10 definitely proves that the proposed metric, here implemented with $\Delta = 10^{-3}$, approximates this behavior much better than the considered alternatives. In particular, the AWGN metric and the SL metric, here implemented with $V_T = 1.3$, cannot exploit the imaginary component of y_k (thus not reported in Fig. 4.10) and dramatically fail in producing low-magnitude LLRs when the presence of interfering impulses is very likely. In conclusion, it is easy to predict that the proposed metric will significantly outperform them when systems requiring high-quality soft-output detection are considered. On the other hand,

the generation of erasure decisions (that is null LLRs in Fig. 4.10) makes the proposed metric less suitable for hard-output detection.

We now discuss the choice of the threshold parameter Δ, which is crucial for the performance of the proposed metric. Following the arguments we presented in [78], one can derive a rule of thumb for choosing a suitable value of Δ when some rough *a priori* information on the statistics of the impulse noise is available. On the other hand, extensive computer simulations, some of which are reported in Section 4.4.2, show that the proposed metric is very robust, such that no *a priori* information is required. In particular, values of Δ in the order of 10^{-3} result effective irrespectively of the actual statistics of the impulse noise, provided that a powerful channel code is adopted. Such a robustness is due to the fact that values of Δ in the order of 10^{-3} provide the key features required for good soft-output detection, namely the generation of low-magnitude LLRs when the presence of interfering impulses is very likely and the capability of exploiting the dependence between the real and imaginary components of the received samples, and thus result suitable irrespectively of the actual channel parameterization.

4.4.2 Simulation Results

In this section, the performance of the described metrics is assessed by means of computer simulations. The reported results refer to Gray-mapped QPSK transmissions over channels characterized by different statistics of the impulse noise. A $(3, 6)$-regular LDPC code of rate $1/2$ and codeword length of $4,000$ bits is used. The LDPC decoder processes the detection metrics executing 40 self-iterations, but can also stop if, checking the code syndrome, it founds a valid codeword before the 40^{th} iteration. To ensure better numerical stability, all algorithms are implemented in the logarithmic domain [28].

We first consider a channel with impulse noise characterized by $P_B = 10^{-1}$ and $R = 10^2$. Fig. 4.11 shows the performance of the system when different metrics are used, in terms of BER versus SNR. As a comparison, the performance over an AWGN channel is also reported. Although, according to (4.14), the impulse noise increases the power of the overall noise of about 10 dB with respect to the AWGN channel, the optimal metric provides a performance degradation lower than 1.5 dB, canceling out the greatest part of the impulse noise thanks to the powerful coding scheme. We notice that the proposed metric, here implemented with $\Delta = 10^{-3}$, ensures the same performance as the optimal one. Let us point out that the saturation to the minimum thresh-

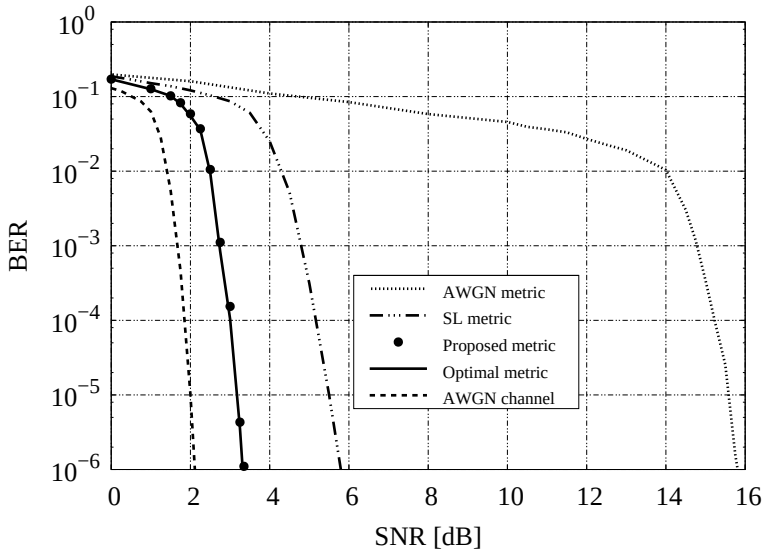

Figure 4.11: Performance over a channel with impulse noise characterized by $P = 10^{-1}$ and $R = 10^2$.

old Δ, which actually is the only difference between the proposed metric and the AWGN metric, provides a gain of more than 12 dB at the expense of a practically null increase in complexity. On the other hand, the simulation results also confirm the conjectures carried out in Section 4.4.1 on the ineffectiveness of the SL metric when employed in systems that require high-quality soft-output detection. In fact, although the threshold value V_T has been optimized for each value of the SNR, we notice that the SL metric exhibits a performance degradation larger than 2 dB with respect to the proposed one.

Fig. 4.12 shows the values of the SNR corresponding to a BER equal to about 10^{-5} when different statistics of the impulse noise are considered. We compare the performance of the optimal metric and that of the proposed one working with $\Delta = 10^{-3}$. Let us remark the robustness of the proposed metric: the performance degradation with respect to the ideal benchmark is negligible irrespectively of the statistics of the impulse noise. In practice, when the proposed metric is adopted, we can just set $\Delta = 10^{-3}$ and there is no need to know the values of P_B and R. This fact, together with the very low computational complexity, definitely makes the proposed solution the most convenient one. Moreover, it is worth to notice that the results

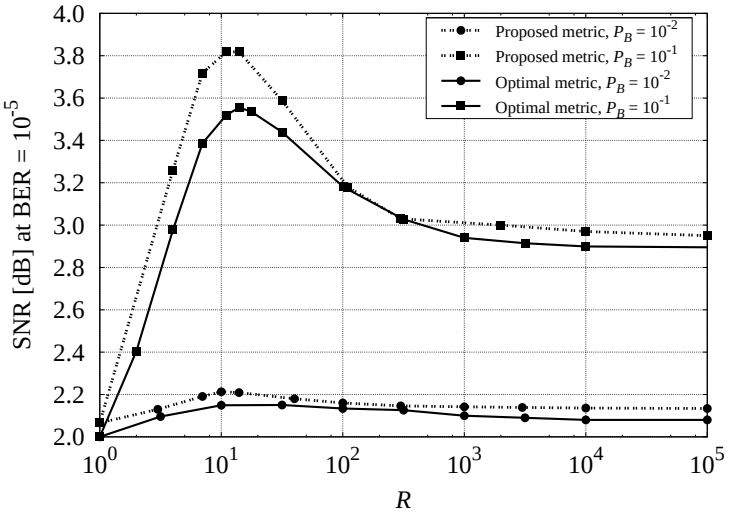

Figure 4.12: Performance over channels with different parameterizations of the impulse noise.

reported in Fig. 4.12 greatly agree with the theoretical results reported in Fig. 4.3, which refer to the same modulation format and the same information rate. In particular, it is confirmed that the impulse noise, irrespectively of its power, can be practically canceled out when the value of P_B is small enough (with the same LDPC code, over an AWGN channel, a BER equal to 10^{-5} is achieved when the value of the SNR is about 2 dB), and that there exists a value R_0 such that, for $R > R_0$, the system can take advantage of a larger power of the impulse noise.

4.4.3 Channel-Model Mismatches

This section introduces a different model for describing the impulse noise, widely known as *class-A* model [72], and discusses the effectiveness of the metric (4.24) when applied to class-A channels.

First, we recall the class-A model. The received samples can still be written as in (4.2), but the noise samples $\{n_k\}$ are IID and have statistical properties that differ from those described in Section 4.1. Namely, at each time epoch k, the statistical properties of the sample n_k are completely defined by the channel state s_k, which belongs to the set of the non-negative integers $\mathbb{N}$, and takes

the value $i \in \mathbb{N}$ with probability [72]

$$P_i = \frac{e^{-A} A^i}{i!} \tag{4.25}$$

where A is a positive parameter characterizing the channel, generally referred to as *impulsive index*. In particular, the sample n_k is a zero-mean circularly-symmetric Gaussian random variable with variance depending on s_k, so that the PDF of n_k conditioned to s_k can be written as [72]

$$p(n_k|s_k = i) = \frac{1}{2\pi N_i} \exp\left\{-\frac{|n_k|^2}{2N_i}\right\} \quad , \quad i \in \mathbb{N} \tag{4.26}$$

where N_i, that is the variance per component of the noise samples when $s_k = i$, can be written as [72]

$$N_i = \Big(1 + \frac{i}{A\Gamma}\Big) N_0 \quad , \quad i \in \mathbb{N} \tag{4.27}$$

Γ being a positive parameter describing the power of the impulse noise. Hence, the PDF of the noise sample n_k results

$$p(n_k) = \sum_{i=0}^{\infty} P_i \, p(n_k|s_k = i) = \sum_{i=0}^{\infty} \frac{P_i}{2\pi N_i} \exp\left\{-\frac{|n_k|^2}{2N_i}\right\} \, . \tag{4.28}$$

which points out that the class-A model is not finite-state.

By properly setting the values of the parameters A and Γ, a large variety of channels with different statistical properties can be described [72, 85]. As in Section 4.1, we focus on scenarios where the presence of impulsive noise, that is the event $\{s_k > 0\}$, is relatively infrequent with respect to the presence of background noise only, that is the event $\{s_k = 0\}$. Hence, we assume that $P_0 > 1/2$, or, equivalently, that the parameter A satisfies the inequality

$$A < \log_e(2) \simeq 0.693 \, . \tag{4.29}$$

On the other hand, no particular restriction on the value of Γ is assumed.

After few manipulations, it turns out that the optimal detection scheme works symbol-by-symbol and outputs, for each time epoch k and each trial value of c_k, the metric [79, 80]

$$O_k(c_k) = \sum_{i=0}^{\infty} \frac{P_i}{2\pi N_i} \exp\left\{-\frac{|y_k - c_k|^2}{2N_i}\right\} \, . \tag{4.30}$$

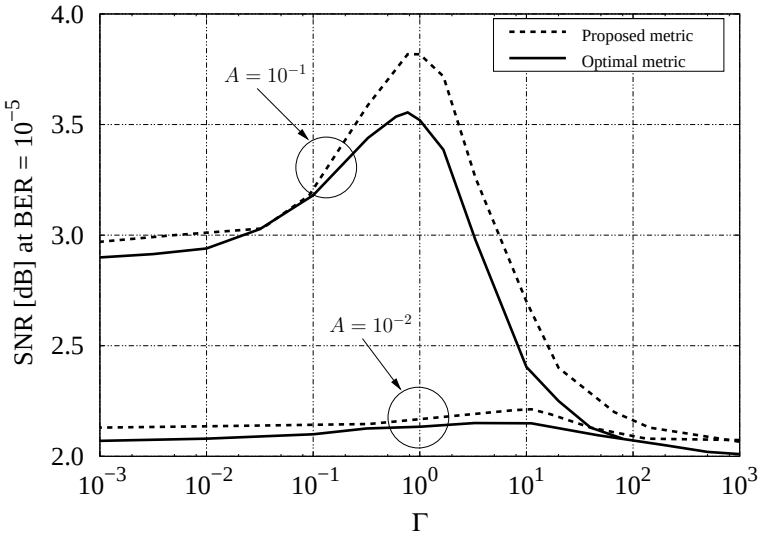

Figure 4.13: Performance over channels with different parameterizations of the class-A impulse noise.

Since the summation in (4.30) involves an infinite number of terms, it is not suitable for practical uses—some useful approximations can be found in [79, 80, 86, 87]. Here, we are interested in evaluating the performance of a receiver that adopts the detection metric (4.24), which, although derived on the basis of a different channel model, is expected to be very robust according to the discussions carried out in Section 4.4.1. Extensive simulation results, some of which are reported in Fig. 4.13, confirm the impressive robustness of the proposed metric, even in these conditions of channel-model mismatch. The reported results refer to Gray-mapped QPSK transmissions over channels characterized by different statistics of the class-A impulse noise, when the same coding scheme and the same receiver as in the scenarios related to Fig. 4.11 and Fig. 4.12 are adopted. Fig. 4.13 compares the value of the SNR, still defined as in (4.15), that provides a BER equal to about 10^{-5} when the optimal metric (4.30) and the proposed metric (4.24), working with $\Delta = 10^{-3}$, are employed. Again, the performance degradation due to the suboptimal metric is limited, irrespectively of the statistics of the impulse noise, such that there is basically no need to know the values of A and Γ.

Chapter 5

Conclusions

This chapter gives concluding remarks on the main results presented in the thesis. We have considered the design of receivers addressed to channels with memory. In particular, the work has focused on finite-state Markovian channels, with emphasis on channels impaired by intersymbol interference and channels impaired by impulse noise. Throughout the thesis, our aim has been to design communication systems able to provide a convenient performance/complexity tradeoff.

First, we have introduced some general frameworks addressed to the analysis and the design of systems matching the finite-state Markovian model. Namely, we have described the reference algorithm for MAP symbol detection, we have explained how to compute the ultimate performance limits imposed by the considered channel model, and we have introduced the general scheme characterizing all systems considered in the thesis, that is a BICM scheme that combines powerful channel codes, such as LDPC codes or turbo codes, with receivers that employ iterative detection/decoding.

The first considered finite-state Markovian channels have been the ISI channels. We have pointed out that, in most channels of practical interest, the complexity of the algorithm for optimal MAP symbol detection is unmanageable, and thus that the search for suboptimal algorithms able to provide a convenient performance/complexity tradeoff is strongly motivated. After an overview of the reduced-complexity algorithms available in the literature, we have presented three different approaches for designing reduced-complexity algorithms. The first approach, which results particularly effective on minimum-phase channels, is based on intentional mismatched decoding. We have shown

how to approximate the actual channel model by means of a simpler (that is with a lower number of states) channel model, and we have proved, by both information-theoretical arguments and simulation results, that effective reduced-complexity receivers can be designed by exploiting this approximation. The second approach is based on reduced search techniques over the full-complexity trellis describing the channel. We have described an alternative formulation of the algorithm for MAP symbol detection that, although equivalent when implemented by full-complexity receivers, results much more suitable for designing low-complexity receivers addressed to mixed-phase and maximum-phase channels. Several simulation results have been reported which show that there exist various scenarios where the proposed algorithms noticeably outperform the existing ones with similar complexity. The third approach stems from the application of the SPA to a proper FG, based on the Ungerboeck model for the ISI channels. Since this FG contains cycles, we have investigated different message-passing schedules, each of them leading to a different detection algorithm. In particular, we have proposed two algorithms that have been shown to provide impressive performance/complexity tradeoffs over different ISI channels.

Finally, we have addressed channels impaired by impulse noise. A two-state Markovian model has been proposed which, unlike the memoryless models generally considered in the literature, allows to describe the typical bursty nature of the impulse noise. We have then proved, by means of information-theoretical arguments, that significant performance improvements can be achieved by properly exploiting the channel memory, and we have described a couple of practical communication schemes able to approach such theoretical limits. In particular, we have proposed a receiver that, although characterized by a similar complexity, noticeably outperforms the conventional memoryless receivers. Finally, the problem of designing robust detection schemes that do not require the knowledge of the statistical properties of the impulse noise, nor their estimation, has been addressed by proposing a novel detection metric. This detection scheme has been shown to perform nearly as the reference one, provided with exact information on the statistics of the noise, and much better than the existing schemes.

List of Acronyms

APP	A Posteriori Probability
AWGN	Additive White Gaussian Noise
BCJR	Bahl, Cocke, Jelinek, Raviv
BER	Bit-Error Rate
BICM	Bit-Interleaved Coded Modulation
BPSK	Binary Phase-Shift Keying
BSPs	Backward-Selected-Paths
BT	Backward Trellis
CSI	Channel-State Information
DE	Density Evolution
DT	Double Trellis
EXIT	EXtrinsic Information Transfer
FBA	Forward-Backward Algorithm
FG	Factor Graph
FSPs	Forward-Selected-Paths
FT	Forward Trellis
FTN	Faster-Than-Nyquist
IID	Independent and Identically Distributed
ISI	InterSymbol Interference
LDPC	Low-Density Parity-Check
LLR	Log-Likelihood Ratio

MAP	Maximum A-Posteriori
ML	Maximum Likelihood
NZ	Non-Zero
PDF	Probability Density Function
PMF	Probability Mass Function
PS-SPA	Parallel-Schedule Sum-Product Algorithm
PSD	Power Spectral Density
PSK	Phase-Shift Keying
QAM	Quadrature Amplitude Modulation
QoS	Quality of Service
QPSK	Quaternary Phase-Shift Keying
RRC	Root Raised Cosine
RS-BCJR	Reduced-State BCJR
RSSD	Reduced-State Sequence Detection
SISO	Soft-Input Soft-Output
SL	Soft Limiting
SNR	Signal-to-Noise Ratio
SOVA	Soft-Output Viterbi Algorithm
SPA	Sum-Product Algorithm
SS-SPA	Serial-Schedule Sum-Product Algorithm

Bibliography

[1] C. Shannon, "A mathematical theory of communication," *Bell System Tech. J.*, pp. 379–423, Jul. 1948.

[2] E. R. Berlekamp, R. J. McEliece, and H. C. A. Van Tilborg, "On the inherent intractability of certain coding problems," *IEEE Trans. Inform. Theory*, vol. 24, no. 3, pp. 384–386, May 1978.

[3] C. Berrou, A. Glavieux, and P. Thitmajshima, "Near Shannon limit error-correcting coding and decoding: turbo-codes," in *Proc. IEEE Intern. Conf. Commun.*, Geneva, Switzerland, May 1993, pp. 1064–1070.

[4] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "Serial concatenation of interleaved codes: performance analysis, design, and iterative decoding," *IEEE Trans. Inform. Theory*, vol. 44, no. 3, pp. 909–926, May 1998.

[5] D. J. C. MacKay and R. M. Neal, "Near Shannon limit performance of low density parity check codes," *IEE Electronics Letters*, vol. 32, no. 18, pp. 1645–1646, Aug. 1996.

[6] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. A. Spielman, "Improved low-density parity-check codes using irregular graphs," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 585–598, February 2001.

[7] D. Divsalar, H. Jin, and R. J. McEliece, "Coding theorems for 'turbo-like' codes," in *Proc. Annu. Allerton Conf. Communications, Control and Computing*, Urbana, IL, Sep. 1998, pp. 201–210.

[8] H. Jin, A. Khandekar, and R. J. McEliece, "Irregular repeat-accumulate codes," in *Proc. Intern. Symp. on Turbo Codes & Relat. Topics*, Brest, France, Sep. 2000.

[9] R. G. Gallager, *Low-Density Parity-Check Codes*. Cambridge, MA: MIT Press, 1963.

[10] G. D. Forney, Jr., *Concatenated Codes*. Cambridge, MA: MIT Press, 1966.

[11] J. Hagenauer, "The turbo principle: tutorial introduction and state of the art," in *Proc. Intern. Symp. on Turbo Codes & Relat. Topics*, Brest, France, Sep. 1997, pp. 1–11.

[12] N. Wiberg, "Codes and decoding on general graphs," Ph.D. dissertation, Linköping University (Sweden), 1996.

[13] R. M. Tanner, "A recursive approach to low complexity codes," *IEEE Trans. Inform. Theory*, vol. 27, pp. 533–547, Sep. 1981.

[14] R. J. McEliece, D. J. C. MacKay, and J. F. Cheng, "Turbo decoding as an instance of Pearl's "belief propagation" algorithm," *IEEE J. Select. Areas Commun.*, vol. 16, pp. 140–152, February 1998.

[15] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.

[16] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inform. Theory*, vol. 47, pp. 498–519, Feb. 2001.

[17] L. Perez, J. Seghers, and D. J. Costello, "A distance spectrum interpretation of turbo codes," *IEEE Trans. Inform. Theory*, vol. 42, pp. 1698–1709, Nov. 1996.

[18] D. J. C. MacKay, "Good error correcting codes based on very sparse matrices," *IEEE Trans. Inform. Theory*, vol. 45, pp. 399–431, Feb. 1999.

[19] T. Richardson and R. Urbanke, "The capacity of low density parity check codes under message passing decoding," *IEEE Trans. Inform. Theory*, vol. 47, pp. 599–618, Feb. 2001.

[20] J. G. Proakis, *Digital Communications*, 4th ed. New York: McGraw-Hill, 2001.

[21] G. Caire, G. Taricco, and E. Biglieri, "Bit-interleaved coded modulation," *IEEE Trans. Inform. Theory*, vol. 44, no. 3, pp. 927–946, May 1999.

[22] A. Papoulis, *Probability, Random Variables and Sthocastic Processes.* New York, NY: McGraw-Hill, 1991.

[23] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Trans. Inform. Theory*, vol. 13, pp. 259–260, Apr. 1967.

[24] G. D. Forney, Jr., "The Viterbi algorithm," *Proc. IEEE*, vol. 61, pp. 268–278, Mar. 1973.

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

[26] L. Baum and T. Petrie, "Statistical inference for probabilistic functions of finite state Markov chains," *Ann. Math. Stat.*, vol. 37, pp. 1554–1563, 1966.

[27] R. W. Chang and J. C. Hancock, "On receiver structures for channels having memory," *IEEE Trans. Inform. Theory*, vol. 12, pp. 463–468, Oct. 1966.

[28] P. Roberston, E. Villebrun, and P. Hoeher, "Optimal and sub-optimal maximum a posteriori algorithms suitable for turbo decoding," *European Trans. Telecommun.*, vol. 8, no. 2, pp. 119–125, March/April 1997.

[29] T. M. Cover and J. A. Thomas, *Elements of Information Theory.* New York: John Wiley & Sons, Inc., 1991.

[30] M. Mushkin and I. Bar-David, "Capacity and coding for the Gilbert-Elliott channels," *IEEE Trans. Inform. Theory*, vol. 35, no. 6, pp. 1277–1290, Nov. 1989.

[31] A. Goldsmith and P. Varaiya, "Capacity, mutual information and coding for finite-state Markov channels," *IEEE Trans. Inform. Theory*, vol. 42, pp. 868–886, May 1996.

[32] V. Sharma and S. K. Singh, "Entropy and channel capacity in the regenerative setup with application to Markov channels," in *Proc. IEEE International Symposium on Information Theory*, Washington, DC, Jun. 2001, p. 283.

[33] H. D. Pfister, J. B. Soriaga, and P. H. Siegel, "On the achievable information rates of finite-state ISI channels," in *Proc. IEEE Global Telecommun. Conf.*, San Antonio, TX, 2001, pp. 2992–2996.

[34] D. M. Arnold, H.-A. Loeliger, P. O. Vontobel, A. Kavčič, and W. Zeng, "Simulation-based computation of information rates for channels with memory," *IEEE Trans. Inform. Theory*, vol. 52, no. 8, pp. 3498–3508, Aug. 2006.

[35] N. Merhav, G. Kaplan, A. Lapidoth, and S. S. Shitz, "On information rates for mismatched decoders," *IEEE Trans. Inform. Theory*, vol. 40, no. 6, pp. 1953–1967, Nov. 1994.

[36] G. Colavolpe, G. Ferrari, and R. Raheli, "Extrinsic information in iterative decoding: a unified view," *IEEE Trans. Commun.*, vol. 49, pp. 2088–2094, Dec. 2001.

[37] C. Douillard, M. Jezequel, C. Berrou, A. Picart, P. Didier, and A. Glavieux, "Iterative correction of intersymbol interference: turbo-equalization," *European Trans. Telecommun.*, vol. 6, no. 5, pp. 507–511, September/October 1995.

[38] M. Tüchler, R. Koetter, and A. C. Singer, "Turbo equalization: Principles and new results," *IEEE Trans. Commun.*, vol. 55, pp. 754–767, May 2002.

[39] X. Li and J. A. Ritcey, "Bit-interleaved coded modulation with iterative decoding using soft feedback," *Electronics Letters*, vol. 34, no. 10, pp. 942–943, May 1998.

[40] J. Boutros and G. Caire, "Iterative multiuser joint decoding: unified framework and asymptotic analysis," *IEEE Trans. Inform. Theory*, vol. 48, no. 7, pp. 1772–1793, July 2002.

[41] S.-Y. Chung, T. Richardson, and R. Urbanke, "Analysis of sum-product decoding of low-density parity-check codes using Gaussian approximation," *IEEE Trans. Inform. Theory*, vol. 47, pp. 657–670, Feb. 2001.

[42] A. W. Eckford, F. R. Kschischang, and S. Pasupathy, "On designing good LDPC codes for Markov channels," *IEEE Trans. Inform. Theory*, vol. 53, no. 1, pp. 5–21, Jan. 2007.

[43] S. ten Brink, "Convergence behavior of iteratively decoded parallel concatenated codes," *IEEE Trans. Commun.*, vol. 49, no. 10, pp. 1727–1737, Oct. 2001.

[44] S. ten Brink, G. Kramer, and A. Ashikhmin, "Design of low-density parity-check codes for modulation and detection," *IEEE Trans. Commun.*, vol. 52, pp. 670–678, Apr. 2004.

[45] G. D. Forney, Jr., "Maximum-likelihood sequence estimation of digital sequences in the presence of intersymbol interference," *IEEE Trans. Inform. Theory*, vol. 18, pp. 284–287, May 1972.

[46] G. Ungerboeck, "Adaptive maximum likelihood receiver for carrier-modulated data-transmission systems," *IEEE Trans. Commun.*, vol. com-22, pp. 624–636, May 1974.

[47] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*. Englewood Cliffs, New Jersey: Prentice-Hall, 1989.

[48] J. E. Mazo, "Faster-than-Nyquist signaling," *Bell System Tech. J.*, vol. 54, pp. 1450–1462, Oct. 1975.

[49] F. Rusek and J. B. Anderson, "On information rates of faster than Nyquist signaling," in *Proc. IEEE Global Telecommun. Conf.*, San Francisco, CA, U.S.A., Nov. 2006.

[50] J. G. Proakis and D. G. Manolakis, *Digital signal processing (3rd ed.): principles, algorithms, and applications*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1996.

[51] G. Colavolpe and A. Barbieri, "On MAP symbol detection for ISI channels using the Ungerboeck observation model," *IEEE Commun. Letters*, vol. 9, no. 8, pp. 720–722, Aug. 2005.

[52] W. Xiang and S. S. Pietrobon, "On the capacity and normalization of ISI channels," *IEEE Trans. Inform. Theory*, vol. 49, no. 9, pp. 2263–2268, Sep. 2003.

[53] M. V. Eyuboğlu and S. U. Qureshi, "Reduced-state sequence estimation with set partitioning and decision feedback," *IEEE Trans. Commun.*, vol. 38, pp. 13–20, Jan. 1988.

[54] A. Duel-Hallen and C. Heegard, "Delayed decision feedback estimation," *IEEE Trans. Commun.*, vol. 37, pp. 428–436, May 1989.

[55] P. R. Chevillat and E. Eleftheriou, "Decoding of trellis-encoded signals in the presence of intersymbol interference and noise," *IEEE Trans. Commun.*, vol. 36, pp. 669–676, Jul. 1989.

[56] G. Colavolpe, G. Ferrari, and R. Raheli, "Reduced-state BCJR-type algorithms," *IEEE J. Select. Areas Commun.*, vol. 19, no. 5, pp. 848–859, May 2001.

[57] P. Thiennviboon, G. Ferrari, and K. M. Chugg, "Generalized trellis-based reduced-state soft-input/soft-output algorithms," in *Proc. IEEE Intern. Conf. Commun.*, New York, U.S.A., Apr. 2002, pp. 1667–1671.

[58] V. Franz and J. B. Anderson, "Concatenated decoding with a reduced-search BCJR algorithm," *IEEE J. Select. Areas Commun.*, vol. 16, no. 2, pp. 186–195, Feb. 1998.

[59] M. Sikora and D. J. Costello, Jr., "A new SISO algorithm with application to turbo equalization," in *Proc. IEEE International Symposium on Information Theory*, Adelaide, Australia, Sep. 2005, pp. 2031–2035.

[60] D. Bokolamulla, A. Hansson, and T. Aulin, "Low-complexity iterative detection based on bi-directional trellis search," in *Proc. IEEE International Symposium on Information Theory*, Yokohama, Japan, June-July 2003, p. 396.

[61] C. Fragouli, N. Seshadri, and W. Turin, "On the reduced trellis equalization using the M-BCJR algorithm," AT&T Labs-Research, NJ, Tech. Rep. TR 99.15.1, Nov. 1999.

[62] B. Frey and F. Kschischang, "Early detection and trellis splicing: Reduced complexity iterative decoding," *IEEE J. Select. Areas Commun.*, vol. 16, pp. 153–159, Feb. 1998.

[63] G. Colavolpe and G. Germi, "On the application of factor graphs and the sum-product algorithm to ISI channels," *IEEE Trans. Commun.*, vol. 53, pp. 818–825, May 2005.

[64] D. Fertonani, A. Barbieri, and G. Colavolpe, "Reduced-complexity BCJR algorithm for turbo equalization," in *Proc. IEEE Intern. Conf. Commun.*, Istanbul, Turkey, June 2006, pp. 1237–1242.

[65] ——, "Reduced-complexity BCJR algorithm for turbo equalization," *IEEE Trans. Commun.*, vol. 55, no. 12, pp. 2279–2287, Dec. 2007.

[66] F. Rusek, M. Loncar, and A. Prlja, "A comparison of Ungerboeck and Forney models for reduced-complexity ISI equalization," in *Proc. IEEE Global Telecommun. Conf.*, Washington, DC, U.S.A., Nov. 2007.

[67] N. Nangare, F. Neeser, and K. Narayanan, "A novel reduced complexity forward-backward equalization algorithm," in *Proc. Annu. Allerton Conf. Communications, Control and Computing*, Monticello, IL, Oct. 2004.

[68] L. Papke, P. Robertson, and E. Villebrun, "Improved decoding with the SOVA in a parallel concatenated (turbo-code) scheme," in *Proc. IEEE Intern. Conf. Commun.*, Dallas, Texas, U.S.A., Jun. 1996, pp. 102–106.

[69] J. Hagenauer and P. Hoeher, "A Viterbi algorithm with soft-decision outputs and its applications," in *Proc. IEEE Global Telecommun. Conf.*, Dallas, TX, U.S.A., Nov. 1989, pp. 1680–1686.

[70] J. Zhang and M. P. C. Fossorier, "Shuffled iterative decoding," *IEEE Trans. Commun.*, vol. 2, no. 53, pp. 209–213, Feb. 2005.

[71] M. Zimmermann and K. Dostert, "Analysis and modeling of impulse noise in broad-band powerline communications," *IEEE Trans. Electromagn. Compat.*, vol. 44, no. 1, pp. 249–258, Feb. 2002.

[72] D. Middleton, "Statistical-physical model of electromagnetic interference," *IEEE Trans. Electromagn. Compat.*, vol. 19, no. 3, pp. 106–126, Aug. 1977.

[73] M. Ghosh, "Analysis of the effect of impulse noise on multicarrier and single carrier QAM systems," *IEEE Trans. Commun.*, vol. 44, no. 2, pp. 145–147, Feb. 1996.

[74] D. Fertonani and G. Colavolpe, "Theoretical limits and practical detection schemes for Markovian-Gaussian channels," in *Proc. IEEE Intern. Conf. Commun.*, Beijing, China, May 2008.

[75] ——, "On reliable communications over channels impaired by bursty impulse noise," in *Proc. IEEE Intl. Symp. on Power Line Commun. and Its Appl.*, Jeju Island, Korea, Apr. 2008.

[76] ——, "On reliable communications over channels impaired by bursty impulse noise," submitted to *IEEE Trans. Commun.*, 2008.

[77] B. M. Kurkoski, P. H. Siegel, and J. K. Wolf, "Joint message-passing decoding of LDPC codes and partial-response channels," *IEEE Trans. Inform. Theory*, vol. 48, no. 6, pp. 1410–1422, June 2002.

[78] D. Fertonani and G. Colavolpe, "A simplified metric for soft-output detection in the presence of impulse noise," in *Proc. IEEE Intl. Symp. on Power Line Commun. and Its Appl.*, Pisa, Italy, Mar. 2007, pp. 121–126.

[79] ——, "Theoretical limits and practical detection schemes for channels affected by class-A impulse noise," in *Proc. IEEE Global Telecommun. Conf.*, Washington, DC, USA, Nov. 2007, pp. 146–150.

[80] ——, "A robust metric for soft-output detection in the presence of class-A noise," submitted to *IEEE Trans. Commun.*, 2008.

[81] D. Middleton, "Procedures for determining the parameters of the first-order canonical models of class A and class B electromagnetic interference," *IEEE Trans. Electromagn. Compat.*, vol. EMC-21, pp. 190–208, Aug. 1979.

[82] S. M. Zabin and H. V. Poor, "Efficient estimation of class A noise parameter via the EM algorithm," *IEEE Trans. Inform. Theory*, vol. 37, no. 1, pp. 60–72, Jan. 1991.

[83] J. Mitra and L. Lampe, "Robust decoding for channels with impulse noise," in *Proc. IEEE Global Telecommun. Conf.*, San Francisco, CA, Nov./Dec. 2006.

[84] G. L. Stüber, "Soft-limiter receivers for coded DS/DPSK systems," *IEEE Trans. Commun.*, vol. 38, no. 1, pp. 46–53, Jan. 1990.

[85] A. D. Spaulding and D. Middleton, "Optimum reception in an impulsive interference environment–Part I: Coherent detection," *IEEE Trans. Commun.*, vol. 25, no. 9, pp. 910–923, Sep. 1977.

[86] D. Umehara, H. Yamaguchi, and Y. Morihiro, "Turbo decoding in impulsive noise environment," in *Proc. IEEE Global Telecommun. Conf.*, Dallas, Texas, Nov./Dec. 2004.

[87] H. Nakagawa, D. Umehara, S. Denno, and Y. Morihiro, "A decoding for low density parity check codes over impulsive noise channels," in *Proc. IEEE Intl. Symp. on Power Line Commun. and Its Appl.*, Vancouver, Canada, Apr. 2005.