Chapter 11

The turbo principle applied to equalization and detection

The invention of turbo codes at the beginning of the 90s totally revolutionized the field of error correcting coding. Codes relatively simple to build and decode, making it possible to approach Shannon's theoretical limit very closely, were at last available. However, the impact of this discovery was not limited to one single coding domain. More generally, it gave birth to a new paradigm for designing digital transmission systems, today commonly known as the "turbo principle". To solve certain very complex *a priori* signal processing problems, we can envisage dividing these problems into a cascade of elementary processing operations, simpler to implement. However, today we know that the one-directional succession of these processing operations leads to a loss of information. To overcome this sub-optimality, the turbo principle advocates establishing an exchange of probabilistic information, "in the two directions", between these different processing operations. All of the information available is thus taken into account in solving the global problem and a consensus can be found between all the elementary processing operations in order to elaborate the final decision.

The application of the turbo principle to a certain number of classical problems in digital transmission has provided impressive gains in performance in comparison to traditional systems. Therefore its use rapidly became popular within the scientific community. This chapter presents the first two systems having historically benefited from the application of the turbo principle to a context other than error correction coding. The first system, called turbo equalization, iterates between the equalization function and a decoding function to improve the processing of the intersymbol interference for data transmission over multipath channels. The second, commonly called turbo CDMA, exploits the turbo principle to improve the discrimination between users in the case of a

radio-mobile communication between several users based on the *Code Division Multiple Access* technique.

11.1 Turbo equalization

Multipath channels have the particularity of transforming a transmitted signal into a linear superposition of several different copies (or echoes) of this signal. Turbo equalization is a digital reception technique that makes it possible to detect data deteriorated by multipath transmission channels. It combines the work of an equalizer and a channel decoder using the turbo principle. Schematically, this digital reception system involves a repetition of the equalization-interleaving-decoding processing chain. First, the equalization performs an initial estimation of the transmitted data. Second, the estimation is transmitted to the decoding module which updates this information. Then the information updated by the decoder is sent to the equalization module. Thus, over the iterations, the equalization and decoding processing operations exchange information in order to reach the performance of a transmission on a channel with a single path.

The purpose of this section is to present the turbo equalization principle and its implementation in two versions: turbo equalization according to the Maximum *A Posteriori* (MAP) criterion and turbo equalization according to the Minimum Mean Square Error (MMSE) criterion. We will describe the algorithms associated with these two techniques, as well as their respective complexity. This will lead us to present the possible architectures and give examples of implementation. Finally, potential and existing applications for these techniques will be shown.

11.1.1 Multipath channels and intersymbol interference

This section is dedicated to transmissions on multipath channels whose particularity is to generate one or several echoes of the signal transmitted. Physically these echoes can, for example, correspond to reflections off a building. The echoes thus produced come and superpose themselves on the signal initially transmitted and thus degrade the reception. The equivalent discrete channel model allows a mathematically simple representation of these physical phenomena in the form of a linear filtering of the transmitted discrete-time symbol sequence. Let x_i be the symbol transmitted at discrete instant i, and y_i be the received symbol at this same instant. The channel output is then given by

$$y_i = \sum_{k=0}^{L-1} h_k(i) x_{i-k} + w_i \tag{11.1}$$

where $h_k(i)$ represents the action of the channel (echo) at instant i on a symbol transmitted at instant $i-k$. The impulse response of the channel at instant i

is then written in the following way in the form of a z-transform:

$$h(z) = \sum_{k=0}^{L-1} h_k(i) z^{-k} \tag{11.2}$$

The impulse response of the channel is assumed to have finite duration (L coefficients), which is a realistic hypothesis in practice in most scenarios.

Equation (11.1) shows that generally, received symbol y_i is a function of the symbols transmitted before, or after (if the channel introduces a propagation delay) information symbol x_i considered at instant i. In accordance with what was introduced in Chapter 2, we then say that the received signal is spoiled by intersymbol interference (ISI). If we now assume that the transmission channel does not vary (or very little) on the duration of a transmitted block of information, model(11.1) can be simplified as follows:

$$y_i = \sum_{k=0}^{L-1} h_k x_{i-k} + w_i \tag{11.3}$$

where we have suppressed the time dependency from the coefficients of the equivalent discrete channel. The representation of the equivalent discrete channel in the form of a digital filter with finite impulse response presented in Figure 11.1 comes directly from (11.3). The coefficients of the filter are precisely those of the impulse response of the channel.

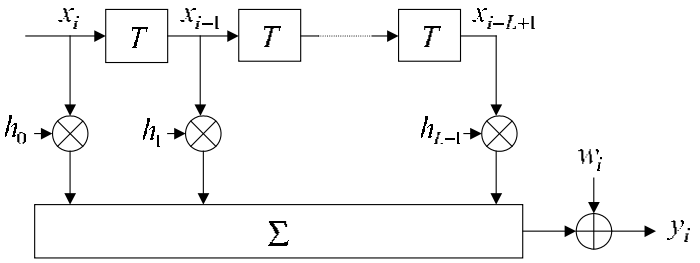

Figure 11.1 – Representation of the equivalent discrete channel in the form of a digital filter.

ISI can be a major obstacle for establishing a good quality digital transmission, even in the presence of very low noise. As an illustration, we have shown in Figure 11.2 the constellation of the symbols received at the output of a channel highly perturbed by ISI, for a signal to noise ratio of 20 dB[1], given that we have transmitted a sequence of discrete symbols with four phase states (QPSK modulation). We thus observe that when the ISI is not processed by an

[1] We recall that a signal to noise ratio of 20 dB corresponds to a power of the transmitted signal 100 times higher than the power of the additive noise on the link.

adequate device, it can lead to great degradation in the error rate at reception, and therefore in the general quality of the transmission.

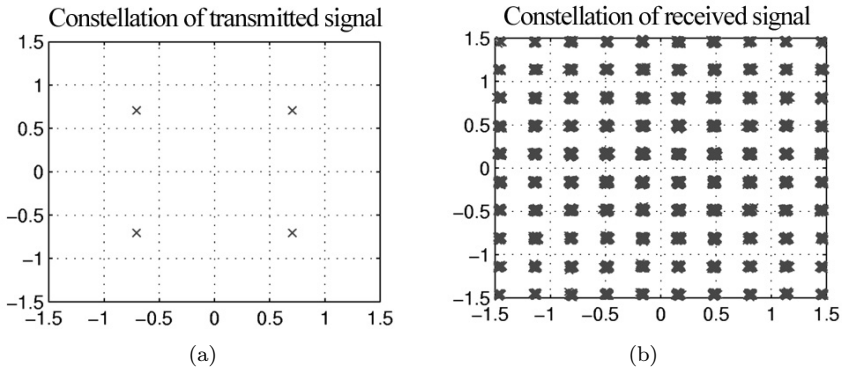

Figure 11.2 – Illustration of the phenomenon of ISI in the case of a 5-path highly frequency-selective channel, for a signal to noise ratio of 20 dB.

We now study the characteristics of a multipath channel in the frequency domain. We show in Figure 11.3 the frequency response of the channel generating the constellation presented in Figure 11.2. The latter is highly perturbed by ISI. We note that the frequencies of the signal will not be attenuated and delayed in the same way over the whole frequency band. Thus, a signal having a band W between 0 and 3 kHz will be distorted by the channel. We then speak of a frequency selective channel in opposition to a flat non-frequency selective channel, for which all the frequencies undergo the same distortion. To resume, when a multipath channel generates intersymbol interference in the time domain, it is then frequency selective in the frequency domain.

We mainly have three different techniques to combat the frequency selectivity of transmission channels: multi-carrier transmissions, spread spectrum and equalization. In this chapter, we deal only with the third solution, applied here to transmissions on a single carrier frequency ("single-carrier" transmissions). Note, however, that some of the concepts tackled here can be transposed relatively easily to systems of the multi-carrier type (*Orthogonal Frequency Division Multiplex*, or OFDM systems).

11.1.2 The equalization function

In its most general form, the purpose of the equalization function is to give an estimation of the transmitted sequence of symbols from the sequence observed at the output of the channel, the latter being perturbed both by intersymbol interference and additive noise, assumed to be Gaussian. We distinguish different equalization strategies. Here we limit ourselves to a succinct overview of the

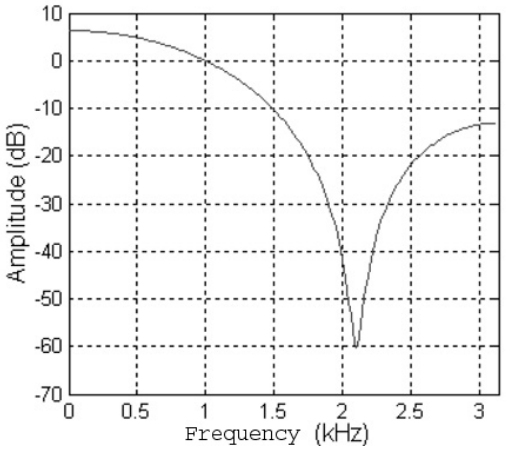

Figure 11.3 – Frequency response of the 5-path discrete-time channel.

main techniques usually implemented in systems. The interested reader can find additional information in Chapters 10 and 11 of [11.44], in articles [11.45] and [11.54] or in book [11.11], for example.

A first solution, called *Maximum Likelihood Sequence Detection*, or MLSD, involves searching for the most probable sequence transmitted relatively to the observation received at the output of the channel. We can show that this criterion amounts to choosing the candidate sequence at the minimum Euclidean distance from the observation, and that it thus minimizes the error probability per sequence, that is to say, the probability of choosing a candidate sequence other than the sequence transmitted. A naive implementation of this criterion involves listing the set of admissible sequences in such a way as to calculate the distance between each sequence and the observation received, then to select the sequence closest to this observation. However, the complexity of this approach increases exponentially with the size of the message transmitted, which turns out to be unacceptable for a practical implementation.

In a famous article dating from 1972 [11.27], Forney noted that a frequency selective channel presents a memory effect whose content characterizes its state at a given instant. More precisely, state s of the channel at instant i is perfectly defined by the knowledge of the $L-1$ previous symbols, which we denote $s = (x_i, \ldots, x_{i-L+2})$. This fact is based on the representation of the channel in the form of a shift register (see Figure 11.1). The evolution of the state of the channel over time can then be represented by trellis diagram having M^{L-1} states, where M denotes the number of discrete symbols in the modulation alphabet. As an illustration, we have represented in Figure 11.4 the trellis associated with

a channel having $L = 3$ coefficients $\mathbf{h} = (h_0, h_1, h_2)$ in the case of a binary phase-shift-keying (BPSK) modulation.

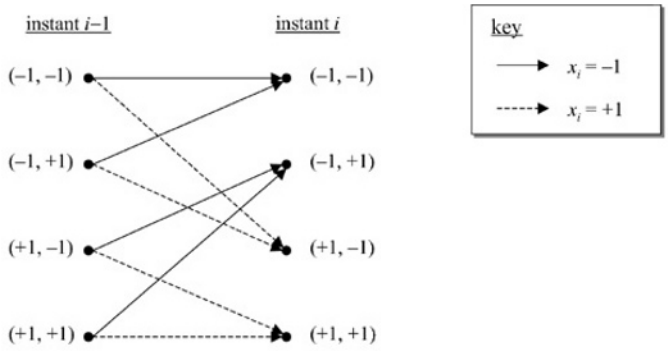

Figure 11.4 – Trellis representation for BPSK transmission on a frequency-selective discrete-time channel with $L = 3$ paths.

Each candidate sequence takes a single path in the trellis. Searching for the sequence with the minimum Euclidean distance from the observation can then be performed recursively, with a linear computation cost depending on the size of the message, by applying the Viterbi algorithm on the trellis of the channel.

The MLSD equalizer offers very good performance. However, the complexity of its implementation increases proportionally with the number of states in the trellis, and therefore exponentially with duration L of the impulse response of the channel and size M of the modulation alphabet. Its practical utilization is consequently limited to transmissions using modulations with a small number of states (2, 4, or 8) on channels with few echoes. On the other hand, it should be noted that this equalizer requires prior estimation of the impulse response of the channel in order to build the trellis. The MLSD solution has been adopted by many manufacturers to perform the equalization operation in mobile telephones for the worldwide GSM (*Global System Mobile*) standard.

In the presence of modulations with a large number of states or on channels whose impulse response length is large, the MLSD equalizer has an unacceptable computation time for real-time applications. An alternative strategy then involves combating the ISI with the help of equalizers having less complexity, implementing digital filters.

In this perspective, the simplest solution involves applying a linear transverse filter at the output of the channel. This filter is optimized so as to compensate ("equalize") the irregularities of the frequency response of the channel, with the aim of converting the frequency selective channel into an equivalent ideally ISI-free (or frequency-flat) channel, perturbed only by additive noise. The transmitted message is then estimated thanks to a simple operation of symbol by symbol decision (threshold detector) at the output of the equalizer, optimal

on an additive white Gaussian noise (AWGN) channel. This equalizer, shown in Figure 11.5, is called a *linear equalizer* or LE.

Figure 11.5 – Linear equalizer.

We distinguish several optimization criteria to define the coefficients of the transverse filter. The optimal criterion involves minimizing the symbol error probability at the output of the filter, but its application leads to a system of equations difficult to solve. In practice, we prefer criteria sub-optimal in terms of performance, but leading to solutions easily implementable, like the *Minimum Mean Square Error* or MMSE criterion [11.44]. The linear MMSE equalizer is an attractive solution due to its simplicity. However, this equalizer suffers from the problem of amplification of the noise level on highly selective channels having strong attenuations at certain points in the frequency spectrum.

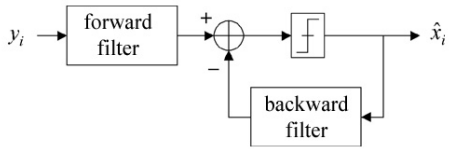

Figure 11.6 – Decision-feedback equalizer.

Examining the diagram of the principle of the linear equalizer, we can note that when we take a decision on symbol x_i at instant i, we have an estimation on the previous symbols $\hat{x}_{i-1}$, $\hat{x}_{i-2}$, ... We can therefore envisage rebuilding the (causal) interference caused by these data and therefore cancel it, in order to improve the decision. The equalizer which results from this reasoning is called a *Decision-Feedback Equalizer* or DFE. The diagram of the principle of the device is illustrated in Figure 11.6. It is made up of a forward filter, in charge of converting the impulse response of the channel into a purely causal response, followed by a decision device and a feedback filter, in charge of estimating the residual interference at the output of the feedback filter in order to cancel it via a feedback loop.

As a general rule, the DFE provides performance higher than that of the linear equalizer, particularly on channels that are highly frequency selective. However, this equalizer is non-linear in nature, due to the presence of the decision device in the feedback loop, which can give rise to an error propagation phenomenon (particularly at low signal to noise ratio) when some of the estimated data are incorrect. In practice, the filter coefficients are generally optimized following the MMSE criterion, by assuming that the estimated data are equal

to the transmitted data, in order to simplify the computations (see Chapter 10 in [11.44], for example).

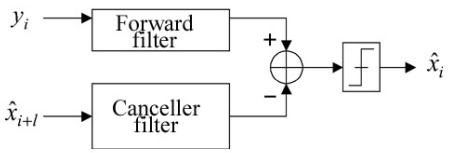

Figure 11.7 – Interference canceller.

If we assume now that we have a estimation $\hat{x}_{i+l}$ on the transmitted data both before ($l < 0$) and after ($l > 0$) the symbol considered at instant i, we can then envisage removing the whole of the ISI at the output of the channel. The equalization structure obtained is called an *interference canceller*, or IC [11-6,11-7]. It is detailed in Figure 11.7. This structure is made up of two digital transverse filters, with finite impulse response: a forward filter (matched to the channel) whose aim is to maximize the signal to noise ratio before the decision, and a canceller filter, in charge of rebuilding the ISI present at the output of the matched filter. Note that by construction, the central coefficient of the canceller filter is necessarily null in order to avoid subtracting the useful signal. With the reserve that the estimated data $\hat{x}_{i+l}$ be equal to the transmitted data, we can show that this equalizer eliminates all the ISI, without any increase in noise level. We thus reach the matched-filter bound, which represents what we can best do with an equalizer on a frequency selective channel. Of course, we never know *a priori* the transmitted data in practice. The difficulty then lies in building an estimation of the data that is sufficiently reliable to keep performance close to optimal.

None of the equalizer structures presented so far take into account the presence of a possible error correcting code on transmission. We shall now see how we can best combine the equalization and decoding functions to improve the global performance of the receiver.

11.1.3 Combining equalization and decoding

Most single-carrier digital transmission systems operating on frequency selective channels incorporate an error correction coding function before the actual modulation step at transmission. The error correcting code is traditionally inserted to combat the errors caused by the additive noise on the link. However, coupled with a carefully built interleaving function, the encoder also offers an additional degree of protection faced with power fading caused by the channel, when the characteristics of the latter vary over time. We saw in the previous section that independently of the nature of the equalizer used, the ISI systematically leads to a loss in performance compared with a non-selective AWGN channel. The

presence of the encoder can then be exploited to reduce this gap in performance, by benefiting from the coding gain at reception.

In the following part of this section, we are going to examine a transmission system shown in Figure 11.8. The modulation and transmission operations on the channel are here represented in equivalent baseband, in order not to have to consider a carrier frequency.

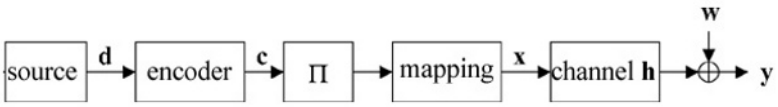

Figure 11.8 – Baseband model of the transmission system considered.

The source sends a sequence of Q information bits, $\mathbf{d} = (d_0, \ldots, d_{Q-1})$. This message is protected by a convolutional code of rate R, to provide a sequence $\mathbf{c} = (c_0, \ldots, c_{P-1})$ of $P = Q/R$ coded bits. The coded bits are then interleaved following a permutation Π, then finally converted in groups of m successive bits into discrete complex symbols chosen in an alphabet with $M = 2^m$ elements, that we will denote $\{X_1, \ldots, X_M\}$. This is the mapping operation. We thus obtain a sequence of $N = P/m$ complex symbols: $\mathbf{x} = (x_0, \ldots, x_{N-1})$. Later, the vector of m coded and interleaved binary elements associated with symbol x_i at instant i will be denoted $(x_{i,1}, \ldots, x_{i,j}, \ldots, x_{i,m})$. The transmitted symbols are discrete random variables with zero mean and variance $\sigma_x^2 = E\{|x_i|^2\}$.

This transmission scheme is called *Bit Interleaved Coded Modulation* or BICM. We still find it today in many systems: the mobile telephony standard GSM, for example. For voice transmission at 13 kbits/s (TCH/FS channel), the specifications of the radio interface indicate that at the output of the speech encoder, the most sensitive bits (class 1A and 1B bits) are protected by a convolutional code with rate $R = 1/2$ and generator polynomials (33,23) in octal [11.1]. The coded message is then interleaved on 8 consecutive packets (or *bursts*), to benefit from time-diversity in the presence of fading, then finally modulated following a waveform of the *Gaussian Minimum Shift Keying* (GMSK) type.

If we now return to the scenario of Figure 11.8, sequence $\mathbf{x}$ is transmitted within a frequency selective channel with L coefficients, with discrete impulse response $\mathbf{h} = (h_0, \ldots, h_{L-1})$. The resulting signal is then perturbed by a vector $\mathbf{w}$ with complex centred AWGN samples and variance $\sigma_w^2 = E\{|w_i|^2\}$. The noisy sequence observed at the output of the channel is finally denoted $\mathbf{y} = (y_0, \ldots, y_{N-1})$, the expression of sample y_i at instant i being given by Equation (11.3).

In this context, the problem that faces the designer of the receiver is the following: how can we best combine equalization and decoding, so that each processing benefits from the result of the other processing?

In reply to this question, estimation theory tells us that to minimize the error probability in this case, the equalization and decoding operations must be performed jointly, following the maximum likelihood criterion. Conceptually, implementing the optimal receiver then amounts to applying the Viterbi algorithm, for example, on a "super-trellis" simultaneously taking into account the constraints imposed by the code, the channel and the interleaver. However, the "super-trellis" has a number of states that increases exponentially with the size of the interleaver, which excludes a practical implementation of the optimal receiver. It is therefore legitimate to question the feasibility of such a receiver in the absence of an interleaver. Historically, this question has been asked in particular in the context of data transmission over twisted-pair telephone cables (voice-band modems). These systems implement error correction coding in Euclidean space (trellis coded modulations), without interleaving, and the telephone channel is a typical example of a frequency-selective, time-invariant channel. Assuming an encoder with S states, a constellation of M points and a discrete channel with L coefficients, the studies undertaken in this context have shown that the corresponding "super-trellis" has exactly $S(M/2)^{L-1}$ states [11.13]. It is then easy to verify that in spite of the absence of an interleaver, the complexity of the optimal receiver again rapidly becomes prohibitive, whenever we wish to transmit a high rate of information (with modulations having a large number of states) or when we are confronted with a channel having large delays.

To counter the unaffordable complexity of the optimal receiver, the solution commonly adopted in practice involves performing the equalization and decoding operations disjointly, sequentially in time. If we again take the example of GSM, the received data are thus first processed by an MLSD equalizer. The estimated sequence provided by the equalizer is then transmitted, after deinterleaving, to a Viterbi decoder. The permutation function then plays a twofold role in this context: not only combating slow fading on the channel, but also dispersing error packets at the input of the decoder. This strategy presents the advantage of simplicity of implementation, since the total complexity is then given by the sum of the individual complexities of the equalizer and the decoder. However, it necessarily leads to loss in performance compared with the optimal receiver since the equalization operation does not exploit all the available information. To be more precise, the estimation sent by the equalizer will not necessarily correspond to a valid coded sequence since the equalizer does not take into account the constraints imposed by the code. The performance of the disjoint solution can be improved when we introduce the passing of weighted (probabilistic) information instead of an exchange of binary data between the equalizer and the decoder. By propagating a reliability measure on the decisions of the equalizer, the decoder thus benefits from additional information to produce its own estimation of the message, and we benefit from a correction gain generally of the order of several dB (see for example [11.28, 11.23] or Chapter 3 in [11.15]). Despite this, the

drawback of this solution remains: only one-way communication between the equalizer and the decoder.

Therefore, does a strategy exist that can somehow produce the best of both worlds, capable of reconciling both good performance of the optimal joint receiver and simplicity in implementation of the sub-optimal disjoint receiver? Today, it is possible to reply in the affirmative, thanks to what we have called "turbo equalization".

11.1.4 Principle of turbo equalization

The concept of turbo equalization first saw the light of day in the laboratories of ENST Bretagne at the beginning of the 90s, under the impulsion of the spectacular results obtained with turbo codes. It was the outcome of a very simple realization: the transmission scheme in Figure 11.8 can be seen as the serial concatenation of two codes (Chapter 6), separated by an interleaver, the second code being formed by cascading the mapping operation with the channel[2]. Seen from this angle, it would then seem natural to apply a decoding strategy of the "turbo" type at reception, that is, a reciprocal, iterative exchange of probabilistic information (extrinsic information) between the equalizer and the decoder. The first turbo equalization scheme was proposed in 1995 by Douillard *et al.* [11.12]. This scheme implements a weighted input and output (*Soft Input Soft Output*, or SISO) Viterbi equalizer according to the *Soft Output Viterbi Algorithm* (SOVA). The principle was then used in 1997 by Bauch *et al.*, substituting the SOVA equalizer by a SISO equalizer that was optimal in the sense of the MAP criterion, using the algorithm developed by Bahl *et al.* (BCJR algorithm [11.7]) .

The simulation results quickly showed that the turbo equalizer was capable of totally removing ISI, under certain conditions. Retrospectively, this excellent performance can be explained by the fact that this transmission scheme brings together two key ingredients which are the force of the turbo principle:

1. The implementation of iterative decoding at reception, introducing an exchange of probabilistic information between the processing operations, about which we today know that, when the signal to noise ratio exceeds a certain "convergence threshold", it converges towards the performance of the optimal joint receiver after a certain number of iterations.

2. The presence of an interleaver at transmission, whose role here mainly involves breaking up the error packets at the output of the equalizer (to avoid the phenomenon of error propagation), and decorrelating as far as

[2] Note that, strictly speaking, transmission on a selective channel does not represent a coding operation in itself, despite its convolutional character, as it does not provide any gain. Indeed, it only degrades performance. Nevertheless, this analogy makes sense from the iterative decoding point of view.

possible the probabilistic data exchanged between the equalizer and the channel decoder. The turbo equalizer is then capable of totally compensating the degradation caused by the ISI, with the reserve that the interleaver be large enough and carefully constructed.

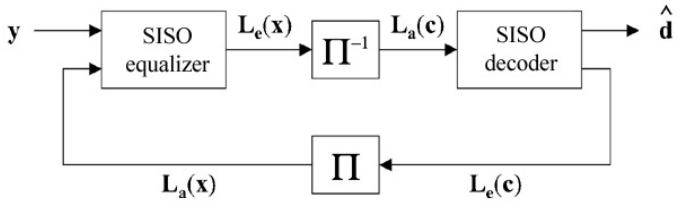

Figure 11.9 – Turbo equalizer for BICM transmission systems.

Generally, the turbo equalizer corresponding to the transmission scenario in Figure 11.8 takes the form shown in Figure 11.9. It is first made up of a SISO equalizer, which at the input takes both vector $\mathbf{y}$ of the data observed at the output of the channel, and *a priori* probabilistic information on the set of coded, interleaved bits $x_{i,j}$, here formally denoted $\mathbf{L}_a(\mathbf{x}) = \{L_a(x_{i,j})\}$. The probabilistic information is propagated in the form of log likelihood ratios (LLRs), the definition of which we recall here for a binary random variable d with values in $\{0, 1\}$:

$$L(d) = \ln\left(\frac{\Pr(d=1)}{\Pr(d=0)}\right) \tag{11.4}$$

The notion of LLR provides twofold information since the sign of the quantity $L(d)$ gives the hard decision on d, while its absolute value $|L(d)|$ measures the reliability this decision can be given.

From the two pieces of information $\mathbf{y}$ and $\mathbf{L}_a(\mathbf{x})$, the SISO equalizer produces extrinsic information denoted $\mathbf{L}_e(\mathbf{x}) = \{L_e(x_{i,j})\}$ on the coded, interleaved binary message. This vector $\mathbf{L}_e(\mathbf{x})$ is then deinterleaved to give a new sequence $\mathbf{L}_a(\mathbf{c})$, which is the *a priori* information on the sequence coded for the SISO decoder. The latter then deduces two pieces of information from this: a hard decision on the information message transmitted, here denoted $\hat{\mathbf{d}}$, and some new extrinsic information on the coded message, denoted $\mathbf{L}_e(\mathbf{c})$. This information is then re-interleaved and sent back to the SISO equalizer where it is exploited as *a priori* information for a new equalization step at the following iteration.

The turbo equalization scheme that we have presented above corresponds to BICM transmitters. It is however important to note that the turbo equalization principle also applies in the case of a system implementing traditional coded modulation, that is to say, a system where the coding and modulation operations are jointly optimized, on condition however that the symbols to be transmitted are interleaved before being modulated and sent on the channel (Figure 11.10).

The main difference with the previous scheme thus lies in the implementation of the SISO equalizer and SISO decoder. Indeed, these latter no longer exchange probabilistic information at binary level but at symbol level, whether in LLR form or directly in probability form. The interested reader can find further details on this subject in [11.8], for example.

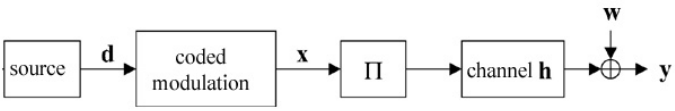

Figure 11.10 – Baseband model of traditional coded modulation systems.

As a general rule, the channel code is a convolutional code and the channel decoder uses a soft-input soft-output decoding algorithm of the MAP type (or its derivatives in the logarithmic domain: Log-MAP and Max-Log-MAP). Again, we will not consider the hardware implementation of the decoder since this subject is dealt with in Chapter 7. Note, however, that unlike classical turbo decoding schemes, the channel decoder here does not provide extrinsic information on the information bits, but instead on the coded bits.

On the other hand, we distinguish different optimization criteria to implement the SISO equalizer, leading to distinct families of turbo equalizers. The first, sometimes called "turbo detection" and what we call MAP turbo equalization here, uses an equalizer that is optimal in the Maximum *A Posteriori* sense. The SISO equalizer is then typically performed thanks to the BCJR-MAP algorithm. As we shall see in the following section, this approach leads to excellent performance, but like the classical MLSD equalizer, it has a very high computation cost which excludes any practical implementation in the case of modulations with a large number of states and for transmissions on channels having large time delays. We must then turn towards alternative solutions, with less complexity but that will necessarily be sub-optimal in nature. One strategy that can be envisaged in this context involves reducing the number of branches to examine at each instant in the trellis. This approach is commonly called "reduced complexity MAP turbo equalization". We know different methods to reach this result, which will be briefly presented in the following section. Another solution is inspired by classical equalization methods and implements an optimized SISO equalizer following the minimum mean square error (MMSE) criterion. We thus obtain an MMSE (filtering-based) turbo equalizer, a scheme described in Section 11.1.6 and that appears as a very promising solution today for high data rates transmissions on highly frequency-selective channels.

11.1.5 MAP turbo equalization

MAP turbo equalization corresponds to the turbo equalization scheme originally introduced by Douillard *et al.* [11.12]. In this section, we first present the equations for implementing the SISO equalizer. The good performance of the MAP turbo equalizer is illustrated by simulation. We also introduce solutions of less complexity derived from the MAP criterion. Finally, we examine the problems encountered during a circuit implementation of the turbo equalizer, as well as potential applications of this reception technique.

Implementation of the BCJR-MAP equalizer

The MAP equalizer shown in Figure 11.11 takes at its input vector $\mathbf{y}$ of the discrete symbols observed at the output of the channel, as well as *a priori* information denoted $\mathbf{L}_a(\mathbf{x})$ on the coded interleaved bits. This information comes from the channel decoder and is produced at the previous iteration. In the particular case of the first iteration, we do not generally have any *a priori* information other than the hypothesis of equiprobability on the bits transmitted, which leads us to put $L_a(x_{i,j}) = 0$.

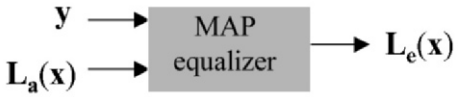

Figure 11.11 – Block diagram of the MAP equalizer.

The purpose of the MAP equalizer is to evaluate the *a posteriori* LLR $L(x_{i,j})$ on each coded interleaved bit $x_{i,j}$, defined as follows:

$$L(x_{i,j}) = \ln\left(\frac{\Pr(x_{i,j} = 1\,|\mathbf{y})}{\Pr(x_{i,j} = 0\,|\mathbf{y})}\right) \tag{11.5}$$

Using conventional results in detection theory, we can show that this equalizer is optimal in the sense of the minimization of the symbol error probability. To calculate the *a posteriori* LLR $L(x_{i,j})$, we will use the trellis representation associated with transmission on the frequency selective channel. Applying Bayes' relation, the previous relation can also be written:

$$L(x_{i,j}) = \ln\left(\frac{\Pr(x_{i,j} = 1, \mathbf{y})}{\Pr(x_{i,j} = 0, \mathbf{y})}\right) \tag{11.6}$$

Among the set of possible sequences transmitted, each candidate sequence traces a single path in the trellis. The joint probability $\Pr(x_{i,j} = 0 \text{ or } 1, \mathbf{y})$ can then be evaluated by summing the probability $\Pr(s', s, \mathbf{y})$ of passing through a particular transition in the trellis linking a state s' at instant $i-1$ to a state s

at instant i, on all of the transitions between instants $i-1$ and i for which the j-th bit making up the symbol associated with these transitions equals 0 or 1. Thus,

$$L(x_{i,j}) = \ln\left(\frac{\sum\limits_{(s',s)/x_{i,j}=1} \Pr(s', s, \mathbf{y})}{\sum\limits_{(s',s)/x_{i,j}=0} \Pr(s', s, \mathbf{y})}\right) \tag{11.7}$$

Adopting a similar approach now to the one presented in the original article by Bahl *et al.* [11.3], we can show that the joint probability $\Pr(s', s, \mathbf{y})$ associated with each transition considered can be decomposed into a product of 3 terms:

$$\Pr(s', s, \mathbf{y}) = \alpha_{i-1}(s')\gamma_{i-1}(s', s)\beta_i(s) \tag{11.8}$$

Figure 11.12 shows the conventions of notation used here.

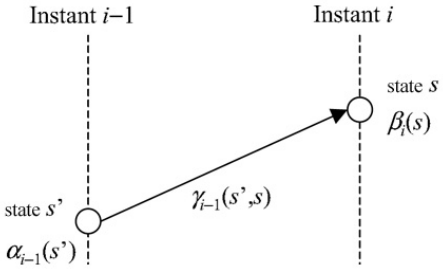

Figure 11.12 – Conventions of notation used to describe the MAP equalizer.

Forward and backward state probabilities $\alpha_{i-1}(s')$ and $\beta_i(s)$ can be calculated recursively for each state and at each instant in the trellis, by applying the following update equations:

$$\alpha_i(s) = \sum_{(s',s)} \alpha_{i-1}(s')\gamma_{i-1}(s', s) \tag{11.9}$$

$$\beta_i(s') = \sum_{(s',s)} \gamma_i(s', s)\beta_{i+1}(s) \tag{11.10}$$

These two steps are called *forward recursion* and *backward recursion*, respectively. Summations are performed over all the couples of states with indices (s', s) for which there is a valid transition between two consecutive instants in the trellis. Forward recursion uses the following initial condition:

$$\alpha_0(0) = 1, \quad \alpha_0(s) = 1 \text{ for } s \neq 0 \tag{11.11}$$

This condition translates the fact that the initial state in the trellis (with index 0, by convention) is perfectly known. Concerning the backward recursion,

we usually assign the same weight to each state at the end of the trellis since the arrival state is generally not known *a priori*:

$$\beta_N(s) = \frac{1}{M^{L-1}} \quad \forall s \tag{11.12}$$

In practice, we see that the dynamic of values $\alpha_{i-1}(s')$ and $\beta_i(s)$ increases during the progression in the trellis. Consequently, these values must be normalized at regular intervals in order to avoid overflow problems in the computations. One natural solution involves dividing these metrics at each instant by constants K_α and K_β chosen in such a way as to satisfy the following normalization condition:

$$\frac{1}{K_\alpha}\sum_s \alpha_i(s) = 1 \quad \text{and} \quad \frac{1}{K_\beta}\sum_s \beta_i(s) = 1 \tag{11.13}$$

the sums here concerning all possible states s of the trellis at instant i.

To complete the description of the algorithm, it remains for us to develop the expression of the term $\gamma_{i-1}(s', s)$. This term can be assimilated to a branch metric. We can show that it is decomposed into a product with two terms:

$$\gamma_{i-1}(s', s) = \Pr(s|s')P(y_i|s', s) \tag{11.14}$$

$\Pr(s|s')$ represents the *a priori* probability of going through the transition between state s and state s', that is to say, the *a priori* probability $P_a(X_l) = \Pr(x_i = X_l)$ of having transmitted at time instant i the constellation symbol X_l labeling the branch considered in the trellis. Owing to the presence of the interleaver at transmission, bits $x_{i,j}$ composing symbol x_i can be assumed statistically independent. Consequently, probability $P_a(X_l)$ has the following decomposition:

$$P_a(X_l) = \Pr(x_i = X_l) = \prod_{j=1}^{m} P_a(X_{l,j}) \tag{11.15}$$

where we have written $P_a(X_{l,j}) = \Pr(x_{i,j} = X_{l,j})$, binary element $X_{l,j}$ taking the value 0 or 1 according to the symbol X_l considered and the mapping rule. Within the turbo equalization iterative process, the *a priori* probabilities $P_a(X_{l,j})$ are deduced from the *a priori* information available at the input of the equalizer. From the initial definition (11.4) of the LLR, we can in particular show that probability $P_a(X_{l,j})$ and corresponding *a priori* LLR $L_a(x_{i,j})$ are linked by the following expression:

$$P_a(X_{l,j}) = K \exp\left(X_{l,j} L_a(x_{i,j})\right) \quad \text{with} \quad X_{l,j} \in \{0, 1\} \tag{11.16}$$

The term K is a normalization constant that we can omit in the following computations without compromising the final result in any way.

Conditional probability $\Pr(s|s')$ is therefore finally given by:

$$\Pr(s\,|s'\,) = \exp\left(\sum_{j=1}^{m} X_{l,j} L_a(x_{i,j})\right) \tag{11.17}$$

As for the second term $P(y_i|s', s)$, it quite simply represents the likelihood $P(y_i|z_i)$ of observation y_i relative to branch label z_i associated with the transition considered. The latter corresponds to the symbol that we would have observed at the output of the channel in the absence of noise:

$$z_i = \sum_{k=0}^{L-1} h_k x_{i-k} \tag{11.18}$$

The sequence of symbols $(x_i, x_{i-1}, \ldots, x_{i-L+1})$ occurring in the computation of z_i is deduced from the knowledge of initial state s' and of information symbol X_l associated with transition $s' \rightarrow s$. In the presence of zero-mean circularly-symmetric complex additive white Gaussian noise with total variance σ_w^2, we obtain:

$$P(y_i\,|s', s\,) = P(y_i\,|z_i\,) = \frac{1}{\pi\sigma_w^2} \exp\left(-\frac{|y_i - z_i|^2}{\sigma_w^2}\right) \tag{11.19}$$

Factor $1/\pi\sigma_w^2$ is common to all the branch metrics and can therefore be omitted in the computations. On the other hand, we see here that calculating branch metrics $\gamma_{i-1}(s',s)$ requires both knowledge of the impulse response of the equivalent discrete channel and knowledge of the noise variance. In other words, in the context of a practical implementation of the system, the MAP equalizer will have to be preceded by a channel estimation procedure.

To summarize, after computing branch metrics $\gamma_{i-1}(s', s)$ then performing the forward and backward recursions, the *a posteriori* LLR $L(x_{i,j})$ is finally given by:

$$L(x_{i,j}) = \ln \frac{\sum\limits_{(s',s)/x_{i,j}=1} \alpha_{i-1}(s')\gamma_{i-1}(s', s)\beta_i(s)}{\sum\limits_{(s',s)/x_{i,j}=0} \alpha_{i-1}(s')\gamma_{i-1}(s', s)\beta_i(s)} \tag{11.20}$$

In reality and in accordance with the turbo principle, it is not this *a posteriori* information that is propagated to the SISO decoder, but rather the extrinsic information. Here, the latter measures the equalizer's own contribution in the global decision process, excluding the information relating to the bit considered coming from the decoder at the previous iteration, that is to say, the *a priori* LLR $L_a(x_{i,j})$. If we develop the expression of branch metric $\gamma_{i-1}(s', s)$ in the

computation of $L(x_{i,j})$, we obtain:

$$L(x_{i,j}) = \ln \left[\frac{\sum\limits_{(s',s)/x_{i,j}=1} \alpha_{i-1}(s') \exp\left(-\frac{|y_i - z_i|^2}{\sigma_w^2} + \sum\limits_{k=1}^{m} X_{l,k} L_a(x_{i,k}) \right) \beta_i(s)}{\sum\limits_{(s',s)/x_{i,j}=0} \alpha_{i-1}(s') \exp\left(-\frac{|y_i - z_i|^2}{\sigma_w^2} + \sum\limits_{k=1}^{m} X_{l,k} L_a(x_{i,k}) \right) \beta_i(s)} \right] \quad (11.21)$$

We can then factorize the *a priori* information term in relation to the bit $x_{i,j}$ considered, both in numerator ($X_{l,j} = 1$) and in denominator ($X_{l,j} = 0$), which gives:

$$L(x_{i,j}) = L_a(x_{i,j}) + \underbrace{\ln \left[\frac{\sum\limits_{(s',s)/x_{i,j}=1} \alpha_{i-1}(s') \exp\left(-\frac{|y_i - z_i|^2}{\sigma_w^2} + \sum\limits_{k \neq j} X_{l,k} L_a(x_{i,k}) \right) \beta_i(s)}{\sum\limits_{(s',s)/x_{i,j}=0} \alpha_{i-1}(s') \exp\left(-\frac{|y_i - z_i|^2}{\sigma_w^2} + \sum\limits_{k \neq j} X_{l,k} L_a(x_{i,k}) \right) \beta_i(s)} \right]}_{L_e(x_{i,j})} \quad (11.22)$$

Finally, we see that the extrinsic information is obtained quite simply by subtracting the *a priori* information from the *a posteriori* LLR calculated by the equalizer:

$$L_e(x_{i,j}) = L(x_{i,j}) - L_a(x_{i,j}) \quad (11.23)$$

This remark concludes the description of the MAP equalizer. As we have presented it, this algorithm proves to be difficult to implement on a circuit due to the presence of numerous multiplication operations. In order to simplify the computations, we can then envisage transposing the whole algorithm into the logarithmic domain (Log-MAP algorithm), the advantage being that the multiplications are then converted into additions, which are simpler to do. If we wish to further reduce the processing complexity, we can also use a simplified (but sub-optimal) version, the Max-Log-MAP (or Sub-MAP) algorithm. These two variants were presented in the context of turbo codes in Chapter 7. The derivation is quite similar in the case of the MAP equalizer. Reference [11.5] presents a comparison in performance between these different algorithms in a MAP turbo equalization scenario. In particular, it turns out that the Max-Log-MAP equalizer offers the best performance/complexity compromise when the estimation of the channel is imperfect.

Example of performance

In order to illustrate the good performance offered by MAP turbo equalization, we chose to simulate the following transmission scenario: a binary source generates messages of 16382 bits of information, which are then protected by a rate

$R = 1/2$ non-recursive non-systematic convolutional code with 4 states, and with generator polynomials (5,7) in octal. Two null bits (tail-bits) are inserted at the end of the message in order to force the termination of the trellis in state 0. Thus we obtain a sequence of 32768 coded bits, which are then randomly interleaved and mapped into a sequence of BPSK symbols. These symbols are transmitted on a 5-path discrete-time channel with impulse response:

$$\mathbf{h} = (0.227, 0.460, 0.688, 0.460, 0.227)$$

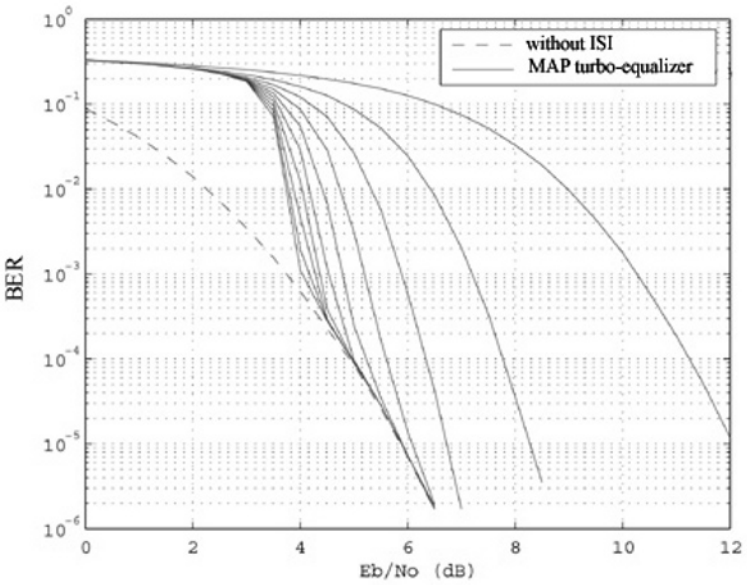

Figure 11.13 – Performance of the MAP turbo equalizer for BPSK transmission on the Proakis C channel, using a rate $R = 1/2$ 4-state non-recursive non-systematic convolutional code and a pseudo-random interleaver of size 32768 bits.

This channel model, called Proakis C, taken from Chapter 10 in [11.44], is relatively difficult to equalize. At reception, we implement 10 iterations of the MAP turbo equalizer described above. The SISO decoder is performed using the BCJR-MAP algorithm. Figure 11.13 presents the bit error rate after decoding, measured at each iteration, as a function of the normalized signal to noise ratio E_b/N_0 on the channel. For reference, we have also shown the performance obtained after decoding on a non-frequency selective AWGN channel. This curve shows the optimal performance of the system. We see that beyond a signal to noise ratio of 5 dB, the turbo equalizer suppresses all the ISI after 5 iterations, and we reach the ideal performance of the AWGN channel. Furthermore, for a

target bit error rate of 10^{-5}, the iterative process provides a gain of the order of 6.2 dB compared with the performance of the conventional receiver performing the equalization and decoding disjointly, given by the curve at the 1^{st} iteration. This performance is very similar to that presented in reference [11.7].

These results give rise to a certain number of remarks, since the example considered here presents the characteristic behaviour of turbo systems. In particular, we see that the gain provided by the iterative process only appears beyond a certain signal to noise ratio (convergence threshold, equal to 3 dB here). Beyond this threshold, we observe a rapid convergence of the turbo equalizer towards the asymptotic performance of the system, given by the error probability after decoding on a non-selective AWGN channel. To improve the global performance of the system, we can envisage using a more powerful error correcting code. Experience shows that we then come up against the necessity of finding a compromise in choosing the code, between rapid convergence of the iterative process and good asymptotic performance of the system (at high signal to noise ratios). The greater the correction capacity of the code, the higher the convergence threshold. On this topic, we point out that today there exist semi-analytical tools such as EXIT (*EXtrinsic Information Transfer*) charts [11.49], enabling the value of the convergence threshold to be predicted precisely, as well as the error rate after decoding for a given transmission scenario, under the hypothesis of ideal interleaving (infinite size). A second solution involves introducing a feedback effect in front of the equivalent discrete-time channel, by inserting an adequate precoding scheme at transmission. Cascading the pre-encoder with the channel produces a new channel model, recursive in nature, leading to a performance gain that is greater, the larger the dimension of the interleaver considered. This phenomenon is known as "interleaving gain" in the literature dedicated to serial turbo codes. Subject to carefully choosing the pre-encoder, we can then exceed the performance of classical non-recursive turbo equalization schemes as has been shown in [11.35] and [11.39].

Complexity of the MAP turbo equalizer and alternative solutions

The complexity of the MAP turbo equalizer is mainly dictated by the complexity of the MAP equalizer. Now, the latter increases proportionally with the number of branches to examine at each instant in the trellis. Considering a modulation with M states and a discrete channel with L coefficients, the total number of branches per section of the trellis rises to $M \times M^{L-1} = M^L$. We therefore see that the processing cost associated with the MAP equalizer increases exponentially with the number of states of the modulation and the length of the impulse response of the channel. As an illustration, EDGE (*Enhanced Data Rate for GSM Evolution*) introduces the use of 8-PSK modulation on channels with 6 coefficients maximum, that is, slightly more than 262000 branches to examine at each instant! MAP turbo equalization is therefore an attractive solution for modulations with a low number of states (typically BPSK and QPSK) on chan-

nels having ISI limited to a few symbol periods. Beyond that, we must turn to less complex, but less efficient, solutions.

There are several ways to deal with this problem. If we limit ourselves to using equalizers derived from the MAP criterion, one idea is to reduce the number of paths examined by the algorithm in the trellis. A first approach performs a truncation of the channel impulse response in order to keep only the $J < L$ first coefficients. The number of states in the trellis will then be decreased. The ISI terms ignored in the definition of the states are then taken into account when calculating the branch metrics, using past decisions obtained from the knowledge of the survivor path in each state. This strategy is called *Delayed Decision Feedback Sequence Estimation* (DDFSE). It offers good performance provided most of the channel's energy be concentrated in its first coefficients which, in practice, requires the implementation of a minimum-phase pre-filtering operation. Applying this technique to turbo equalization has, for example, been studied in [11.2]. A refinement of this algorithm involves grouping some states of the trellis together, in accordance with the set-partitioning rules defined by Ungerboeck [11.52] for designing trellis coded modulations. This improvement, called Reduced State Sequence Estimation (RSSE), includes DDFSE as a particular case [11.19]. In a similar way, we can also envisage retaining more than one survivor path in each state to improve the robustness of the equalizer and if necessary to omit the use of pre-filtering [11.42]. Rather than reduce the number of states of the trellis by truncation, it can also be envisaged to examine only a non-exhaustive list of the most likely paths at each instant. The resulting algorithm is called the "M algorithm", and its extension to SISO equalization was studied in [11.17]. Whatever the case, the search for efficient equalizers with reduced complexity regularly continues to give rise to new contributions.

All the strategies that we have mentioned above enter into the category of MAP *turbo equalizers* with reduced complexity. Generally, these solutions are interesting when the number of states of the modulation is not too high. On the other hand, in the case of high data rate transmissions on channels with long delay spreads, it is preferable to envisage filter-based turbo equalizers of the MMSE type.

Architectures and applications

When systems based on MAP turbo equalization require real time processing with relatively high data rates (of the order of several Mbits/s), a software implementation cannot be envisaged. In this case, we must resort to specific ASIC circuits. The circuit implementation of a MAP turbo equalizer poses problems similar to those encountered in the context of the hardware implementation of a turbo decoder. Two architectural solutions can be envisaged:

- The first uses an implementation of the turbo decoder in the form of a *pipeline* by cascading several elementary modules, each module implementing one detection and one decoding iteration.
- The second uses a single hardware module, implementing the successive iterations sequentially, by looping back on itself.

The first architecture presents a smaller latency, so is better adapted to applications requiring high data rates. On the other hand, the second solution enables an economy in the number of transistors and therefore in the silicon surface. In order to further reduce the surface used, some authors have proposed sophisticated architectures enabling part of the SISO algorithm to be shared between the equalizer and the decoder, despite the different structure of the trellises concerned [11.36]. This approach also enables a reduction in length of the critical path, and therefore in the global latency of the system. This last factor can be a major obstacle to the practical implementation of turbo equalization (and turbo systems more generally) since not all applications may tolerate an increase in the processing delay at reception. Resorting to analogue electronics will perhaps soon enable this obstacle to be overcome. An analogue implementation of a simplified MAP turbo equalizer has thus been reported in [11.24], with promising results.

From the algorithmic point of view, the application of MAP turbo equalization to the GSM system has been the subject of several studies [11.15, 11.43, 11.6, 11.18]. The traditional turbo equalization scheme must thus be revised in order to take into account the specificities of the standard (inter-frame interleaving, different levels of protection applied to the bits at the output of the speech encoder, GMSK modulation, ...). Simulation results show generally moderate gains in performance, in return for a large increase in the complexity of the receiver. This can be partly explained by the fact that the conventional GSM system faces only a limited level of ISI on the majority of the test channels defined in the standard. On the other hand, the introduction of 8-PSK modulation in the context of EDGE greatly increases the level of interference. This scenario therefore seems more appropriate for the use of turbo equalization, and has given rise to several contributions. In particular, the authors of [11.40][3] have studied the implementation of a complete turbo equalization system relying on a SISO equalizer of the DDFSE type with pre-filtering, coupled to a channel estimator. They have obtained gains of the order of several dB, depending on the modulation and coding scheme considered, compared with the performance of the classical receiver. Furthermore, they have also proved the fact that the equalization and iterative decoding principle could be carefully exploited in the context of the ARQ retransmission protocol defined in EDGE (the *Incremental Redundancy* scheme) to improve the global quality of service at reception.

[3] See also the references cited in this article.

11.1.6 MMSE turbo equalization

The increase in data rates, in response to current multimedia service requirements, combined with the infatuation with mobility and wireless infrastructures, present receivers with severe propagation conditions. Thus, if we take the example of the radio interface of the Wireless MAN (*Metropolitan Area Network*) 802.16a standard normalized by IEEE during 2003 and operating in the 2-11 GHz band, the ISI encountered is likely to recover up to 50 symbol durations, or even more. Underwater acoustic communications is another example. The application of turbo equalization to such scenarios involves using low complexity SISO equalizers. MMSE turbo equalization is an attractive solution in this context.

In contrast with the approaches described in the previous section, MMSE turbo equalization mainly involves substituting for the MAP equalizer an equalizer structure based on digital filters, optimized according to the minimum mean square error criterion[4]. This solution presents a certain number of advantages. First of all, simulations show that the MMSE turbo equalizer gives very good performance on average, sometimes very close to the performance offered by MAP turbo equalization. On the other hand, the complexity of the MMSE equalizer increases linearly (and not exponentially) with the length of the channel impulse response, independently of the order of the modulation. Finally, as we shall see in what follows, this approach naturally lends itself well to an implementation in adaptive form, appropriate for tracking the time variations of the channel.

Historically, the first MMSE turbo equalization scheme was proposed by Glavieux *et al.* in 1997 [11.20, 11.32, 11.34]. This original contribution laid down the bases of MMSE turbo equalization, particularly for the design of a filter-based soft-input soft-output equalizer. Indeed, classical equalizers based on digital filters do not naturally lend themselves to handling probabilistic information. This difficulty was overcome by inserting a binary to M-ary conversion operation at the input of the equalizer, in charge of rebuilding a soft estimation of the symbols transmitted using the *a priori* information sent by the decoder. In addition, a SISO demapping module placed at the output of the equalizer converts the equalized data (complex symbols) into extrinsic LLR on the coded bits, which are then sent to the decoder. This initial scheme relied on the implementation of an equalization structure of the interference canceller type, whose coefficients were updated adaptively thanks to the *Least Mean Square* (LMS) algorithm.*Least Mean Square* Remarkable progress was then achieved with the work of Wang and Poor [11.56], taken up by Reynolds and Wang [11.47] then by Tüchler *et al.* [11.51, 11.50]. These contributions have made it possible to

[4] Equalizers optimized according to the Zero Forcing criterion could also be envisaged. However these equalizers usually introduce significant noise enhancement on channels with deep nulls in their frequency response, and thus generally turn out to be less efficient than MMSE equalizers.

establish a theoretical expression for the coefficients of the equalizer, explicitly taking into account the presence of *a priori* information on the transmitted data. This progress has proved to be particularly interesting for packet mode transmissions, in which the coefficients of the equalizer are precalculated once from an estimation of the impulse response of the channel, and applied to the whole received block.

MMSE turbo equalization relies on a soft-input soft-output linear equalization scheme optimized according to the MMSE criterion. This type of equalizer is also sometimes known as a "linear MMSE equalizer with *a priori* information" in the literature. This section describes the principle of this equalizer, assuming that we know the parameters of the channel, which enables the filter coefficients to be calculated directly. Its implementation in adaptive form is also discussed. We next present some examples of MMSE turbo equalizer performance, and we describe the (*Digital Signal Processor*, or DSP) implementation of this solution. This part ends with a reflection on the potential applications of MMSE turbo equalization.

Principle of soft-input soft-ouput linear MMSE equalization

Generally, the linear soft-input soft-output MMSE equalizer can formally be decomposed into three main functions (Figure 11.14).

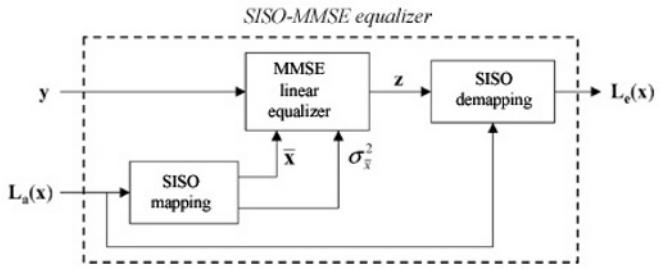

Figure 11.14 – Soft-input soft-output linear equalizer optimized according to the MMSE criterion.

1. The first operation, the SISO mapping, calculates a soft estimate for the transmitted symbols, denoted $\bar{\mathbf{x}} = (\bar{x}_0, \ldots, \bar{x}_{N-1})$, from the *a priori* information $\mathbf{L}_a(\mathbf{x})$ coming from the decoder at the previous iteration.

2. The linear equalizer then uses estimated data $\bar{x}_i$ to rebuild and cancel the ISI affecting the received signal. The resulting signal is filtered in order to eliminate residual interference. The filter coefficients are optimized so as to minimize the mean square error between the equalized data and

the corresponding transmitted data. However, unlike the classical linear MMSE equalizer, the reliability information coming from the decoder is here explicitly taken into account when calculating the coefficients.

3. The equalizer is finally followed by a soft-input soft-output demapping module whose role is to convert the equalized symbols into extrinsic LLRs on the (interleaved) coded bits.

We now examine in greater detail the implementation of each of these three functions.

• *SISO mapping*

This operation involves calculating the soft estimate $\bar{x}_i$, defined as the mathematical expectation of symbol x_i transmitted at instant i:

$$\bar{x}_i = E_a\{x_i\} = \sum_{l=1}^{M} X_l \times P_a(X_l) \tag{11.24}$$

The sum here concerns all of the discrete symbols in the constellation. The term $P_a(X_l)$ denotes the *a priori* probability $\Pr(x_i = X_l)$ of symbol X_l being transmitted at instant i. We have put index a at the level of the expectation term to highlight the fact that these probabilities are deduced from the *a priori* information at the input of the equalizer. Indeed, provided the m bits making up symbol x_i are statistically independent, it is possible to write:

$$P_a(X_l) = \prod_{j=1}^{m} P_a(X_{l,j}) \tag{11.25}$$

where binary element $X_{l,j}$ takes the value 0 or 1 according to the symbol X_l considered and the mapping rule. On the other hand, starting from the general definition (11.4) of the LLR, we can show that the *a priori* probability and the *a priori* LLR are linked by the following relation:

$$P_a(X_{l,j}) = \frac{1}{2}\left(1 + (2X_{l,j} - 1)\tanh\left(\frac{L_a(x_{i,j})}{2}\right)\right) \text{ with } X_{l,j} \in \{0,1\} \tag{11.26}$$

In the particular case of a BPSK modulation, the above computations are greatly simplified. We then obtain the following expression for the soft estimate $\bar{x}_i$:

$$\bar{x}_i = \tanh\left(\frac{L_a(x_i)}{2}\right) \tag{11.27}$$

In the classical situation where we make the hypothesis of equiprobability on the transmitted symbols, we have $L_a(x_{i,j}) = 0$ and $\bar{x}_i = 0$. On the other hand, in the ideal case of perfect *a priori* information, $L_a(x_{i,j}) \to \pm\infty$ and

the soft estimate $\bar{x}_i$ is then strictly equal to the transmitted symbol x_i (perfect estimate). To summarize, the value of the soft estimate $\bar{x}_i$ evolves as a function of the reliability of the *a priori* information provided by the decoder. This explains the name of "soft" (or probabilistic) estimate for $\bar{x}_i$. By construction, the estimated data $\bar{x}_i$ are random variables. In particular, we can show (see [11.33] for example) that they satisfy the following statistical properties:

$$E\{\bar{x}_i\} = 0 \tag{11.28}$$

$$E\left\{\bar{x}_i x_j^*\right\} = E\left\{\bar{x}_i \bar{x}_j^*\right\} = \sigma_{\bar{x}}^2 \delta_{i-j} \tag{11.29}$$

The parameter $\sigma_{\bar{x}}^2$ here denotes the variance of estimated data $\bar{x}_i$. In practice, this quantity can be estimated using the sample variance estimator on a frame of N symbols as follows:

$$\sigma_{\bar{x}}^2 = \frac{1}{N} \sum_{i=0}^{N-1} |\bar{x}_i|^2 \tag{11.30}$$

We easily verify that under the hypothesis of equiprobable *a priori* symbols, $\sigma_{\bar{x}}^2 = 0$. Conversely, we obtain $\sigma_{\bar{x}}^2 = \sigma_x^2$ in the case of perfect *a priori* information on the transmitted symbols. To summarize, the variance of the estimated data offers a measure of the reliability of the estimated data. This parameter plays a major role in the behaviour of the equalizer.

• *Calculating the linear equalizer coefficients*

As explained above, the equalization step can be seen as the cascading of an interference cancellation operation followed by a filtering operation. The filter coefficients are optimized so as to minimize the mean square error $E\{|z_i - x_{i-\Delta}|^2\}$ between the equalized symbol z_i at instant i and symbol $x_{i-\Delta}$ transmitted at instant $i - \Delta$. The introduction of a delay Δ enables the anti-causality of the solution to be taken into account. Here we will use a matrix formalism to derive the optimal form of the equalizer coefficients. Indeed, digital filters always have a finite number of coefficients in practice. The matrix formalism takes this aspect into account and thus enables us to establish the optimal coefficients under the constraint of a finite-length implementation.

Here we consider a filter with F coefficients: $\mathbf{f} = (\mathbf{f}_0, \ldots, \mathbf{f}_{F-1})$. The channel impulse response and the noise variance are assumed to be known, which requires prior estimation of these parameters in practice. Starting from the expression (11.3) and grouping the F last samples observed at the output of the channel up until instant i in the form of a vector column $\mathbf{y}_i$, we can write:

$$\mathbf{y}_i = \mathbf{H}\mathbf{x}_i + \mathbf{w}_i \tag{11.31}$$

with $\mathbf{y}_i = (y_i, \ldots, y_{i-F+1})^{\mathbf{T}}$, $\mathbf{x}_i = (x_i, \ldots, x_{i-F-L+2})^{\mathbf{T}}$ and $\mathbf{w}_i = (w_i, \ldots, w_{i-F+1})^{\mathbf{T}}$. Matrix $\mathbf{H}$, of dimensions $F \times (F + L - 1)$, is a Toeplitz matrix[5]

[5] The coefficients of the matrix are constant along each of the diagonals.

describing the convolution by the channel:

$$\mathbf{H} = \begin{pmatrix} h_0 & \cdots & h_{L-1} & 0 & \cdots & 0 \\ 0 & h_0 & & h_{L-1} & & \vdots \\ \vdots & & \ddots & & \ddots & 0 \\ 0 & \cdots & 0 & h_0 & \cdots & h_{L-1} \end{pmatrix} \tag{11.32}$$

With these notations, the interference cancellation step from the estimated signal $\bar{\mathbf{x}}$ can then be written formally:

$$\tilde{\mathbf{y}}_i = \mathbf{y}_i - \mathbf{H}\tilde{\mathbf{x}}_i \tag{11.33}$$

where the vector $\tilde{\mathbf{x}}_i = (\bar{x}_i, \ldots, \bar{x}_{i-\Delta+1}, 0, \bar{x}_{i-\Delta-1}, \ldots, \bar{x}_{i-F-L+2})^{\mathbf{T}}$ is of dimension $F + L - 1$. The component related to symbol $x_{i-\Delta}$ is set to zero in order to cancel only the ISI and not the signal of interest. At the output of the forward filter, the expression of the equalized sample at instant i is given by:

$$z_i = \mathbf{f}^{\mathbf{T}}\tilde{\mathbf{y}}_i = \mathbf{f}^{\mathbf{T}}\left[\mathbf{y}_i - \mathbf{H}\tilde{\mathbf{x}}_i\right] \tag{11.34}$$

It remains to determine the theoretical expression of the coefficients of the filter $\mathbf{f}$ minimizing the mean square error $E\{|z_i - x_{i-\Delta}|^2\}$. In the most general case, these coefficients vary in time. The corresponding solution, developed in detail by Tüchler *et al.* [11.51, 11.50], leads to an equalizer whose coefficients must be recalculated for each received symbol. This equalizer represents what can best be done currently for MMSE equalization in the presence of *a priori* information. On the other hand, the computation load associated with updating the coefficients symbol by symbol increases quadratically with the number F of coefficients, which again turns out to be too complex for real time implementations. The equalizer that we present here can be seen as a simplified, and therefore sub-optimal, version of the solution cited above. The coefficients of filter $\mathbf{f}$ are calculated only once per frame (at each iteration) and then applied to the whole block, which considerably decreases the implementation cost. On the other hand, and despite this reduction in complexity, this equalizer retains performance close to the optimal one[6], which makes it an excellent candidate for practical realizations. This solution was derived independently by several authors, including [11.51] and [11.33].

With these hypotheses, the optimal form of the set of coefficients $\mathbf{f}$ is obtained using the projection theorem, which stipulates that the estimation error must be orthogonal to the observations[7] :

$$E\left\{(z_i - x_{i-\Delta})\tilde{\mathbf{y}}_i^{\mathbf{H}}\right\} = \mathbf{0} \tag{11.35}$$

[6] The degradation measured experimentally in comparison with Tüchler's original time-varying solution is at most 1 dB, depending on the channel model considered.

[7] We recall here that the notation $\mathbf{A}^{\mathbf{H}}$ denotes the *Hermitian* (conjugate) transpose of matrix $\mathbf{A}$.

We then obtain the following solution:

$$\mathbf{f}^* = E\{\tilde{\mathbf{y}}_i \tilde{\mathbf{y}}_i^{\mathbf{H}}\}^{-1} E\{x_{i-\Delta}^* \tilde{\mathbf{y}}_i\} \tag{11.36}$$

Using the statistical properties of the estimated data $\bar{x}_i$, we note that:

$$E\{x_{i-\Delta}^* \tilde{\mathbf{y}}_i\} = E\{x_{i-\Delta}^* \mathbf{H}(\mathbf{x}_i - \tilde{\mathbf{x}}_i)\} = \mathbf{H}\mathbf{e}_{\boldsymbol{\Delta}} \sigma_x^2 \tag{11.37}$$

where we have introduced the unit vector $\mathbf{e}_\Delta$ with dimension $F + L - 1$ that has a 1 in coordinate Δ and 0 elsewhere. Denoting by $\mathbf{h}_\Delta$ the Δ-th column Δ of matrix $\mathbf{H}$, the previous expression can also be written:

$$E\left\{x_{i-\Delta}^* \tilde{\mathbf{y}}_i\right\} = \mathbf{h}_{\boldsymbol{\Delta}} \sigma_x^2 \tag{11.38}$$

In addition,

$$\begin{aligned} E\left\{\tilde{\mathbf{y}}_i \tilde{\mathbf{y}}_i^{\mathbf{H}}\right\} &= \mathbf{H} E\left\{(\mathbf{x}_i - \tilde{\mathbf{x}}_i)(\mathbf{x}_i - \tilde{\mathbf{x}}_i)^{\mathbf{H}}\right\} \mathbf{H}^{\mathbf{H}} + \sigma_w^2 \mathbf{I} \\ &= (\sigma_x^2 - \sigma_{\bar{x}}^2)\mathbf{H}\mathbf{H}^{\mathbf{H}} + \sigma_{\bar{x}}^2 \mathbf{h}_{\boldsymbol{\Delta}} \mathbf{h}_{\boldsymbol{\Delta}}^{\mathbf{H}} + \sigma_w^2 \mathbf{I} \end{aligned} \tag{11.39}$$

To summarize, the optimal form of the equalizer coefficients can finally be written:

$$\mathbf{f}^* = \left[(\sigma_x^2 - \sigma_{\bar{x}}^2)\mathbf{H}\mathbf{H}^{\mathbf{H}} + \sigma_{\bar{x}}^2 \mathbf{h}_{\boldsymbol{\Delta}} \mathbf{h}_{\boldsymbol{\Delta}}^{\mathbf{H}} + \sigma_w^2 \mathbf{I}\right]^{-1} \mathbf{h}_{\boldsymbol{\Delta}} \sigma_x^2 \tag{11.40}$$

By bringing into play a simplified form of the matrix inversion lemma[8], the previous solution can then be written:

$$\mathbf{f}^* = \frac{\sigma_x^2}{1 + \beta \sigma_{\bar{x}}^2} \tilde{\mathbf{f}}^* \tag{11.41}$$

where we have introduced vector $\tilde{\mathbf{f}}$ and scalar quantity β defined as follows:

$$\tilde{\mathbf{f}}^* = \left[(\sigma_x^2 - \sigma_{\bar{x}}^2)\mathbf{H}\mathbf{H}^{\mathbf{H}} + \sigma_w^2 \mathbf{I}\right]^{-1} \mathbf{h}_{\boldsymbol{\Delta}} \quad \text{and} \quad \beta = \tilde{\mathbf{f}}^{\mathbf{T}} \mathbf{h}_{\boldsymbol{\Delta}} \tag{11.42}$$

By means of this new expression, we note that the computation of the coefficients of the equalizer is mainly based on the inversion of the matrix $(\sigma_x^2 - \sigma_{\bar{x}}^2)\mathbf{H}\mathbf{H}^{\mathbf{H}} + \sigma_w^2 \mathbf{I}$, with dimensions $F \times F$. This matrix has a rich structure since it is a Toeplitz matrix with Hermitian symmetry. Consequently, matrix inversion can be performed efficiently with the help of dedicated algorithms, with a computation cost in $O(F^2)$ (see Chapter 4 in [11.21], for example). In order to reduce even further the complexity of determining the coefficients, the authors of [11.33] have proposed a sub-optimal, but nevertheless efficient, method using the *Fast Fourier Transform*, (or FFT), with a cost in $O(F \log_2(F))$. However, the number of coefficients F must be a power of 2.

[8] $\left[\mathbf{A} + \mathbf{u}\mathbf{u}^{\mathbf{H}}\right]^{-1} = \mathbf{A}^{-1} - \frac{\mathbf{A}^{-1}\mathbf{u}\mathbf{u}^{\mathbf{H}}\mathbf{A}^{-1}}{1 + \mathbf{u}^{\mathbf{H}}\mathbf{A}^{-1}\mathbf{u}}$

It is particularly instructive to study the limiting form taken by the equalizer in the classical case where the transmitted symbols are assumed to be equiprobable (which corresponds to the 1st iteration of the turbo equalizer). In this case, $\sigma_{\bar{x}}^2 = 0$ and the equalizer coefficients can be written:

$$\mathbf{f}^* = \left[\sigma_x^2 \mathbf{H}\mathbf{H}^{\mathbf{H}} + \sigma_w^2 \mathbf{I}\right]^{-1} \mathbf{h}_{\mathbf{\Delta}} \sigma_x^2 \tag{11.43}$$

Here we can recognize the form of a classical linear MMSE equalizer with finite length. Inversely, under the hypothesis of perfect *a priori* information on the transmitted symbols, we have $\sigma_{\bar{x}}^2 = \sigma_x^2$. The equalizer then takes the following form:

$$\mathbf{f} = \frac{\sigma_x^2}{\sigma_x^2 \|\mathbf{h}\|^2 + \sigma_w^2} \mathbf{h}_{\mathbf{\Delta}}^* \quad \text{with} \quad \|\mathbf{h}\|^2 = \mathbf{h}_{\mathbf{\Delta}}^{\mathbf{H}} \mathbf{h}_{\mathbf{\Delta}} = \sum_{k=0}^{L-1} |h_k|^2 \tag{11.44}$$

and the equalized signal z_i can be written:

$$z_i = \frac{\sigma_x^2 \|\mathbf{h}\|^2}{\sigma_x^2 \|\mathbf{h}\|^2 + \sigma_w^2} \left(x_{i-\Delta} + \mathbf{h}_{\mathbf{\Delta}}^{\mathbf{H}} \mathbf{w}_i\right) \tag{11.45}$$

We recognize here the output of a classical MMSE interference canceller, fed by a perfect estimation of the transmitted data. The equalized signal can be decomposed as the sum of the useful signal $x_{i-\Delta}$, up to a scale factor that is characteristic of the MMSE criterion, and a coloured noise term. In other words, the equalizer suppresses all the ISI without raising the noise level and thus reaches the theoretical matched-filter bound corresponding to ISI-free transmission.

To summarize, we see that the SISO MMSE linear equalizer adapts the equalization strategy according to the reliability of the estimated data, measured here by parameter $\sigma_{\bar{x}}^2$.

To conclude this description of the equalizer, we point out that the interference cancellation operation defined formally by Equation (11.33) has no physical reality in the sense that it cannot be performed directly in this way using transverse linear filters. In practice, we prefer to use one of the two architectures presented in Figure 11.15, strictly equivalent from a theoretical point of view. The coefficient g_Δ appearing in implementation (1) is the central coefficient $g_\Delta = \mathbf{f}^T \mathbf{h}_\Delta$ of the global filter formed by the cascade of the channel with filter $\mathbf{f}$. In the case of implementation (2), we again find the classical structure of an interference canceller type equalizer, operating here on the estimated signal $\bar{\mathbf{x}}$. Filter $\mathbf{g} = \mathbf{f}^T \mathbf{H}$ is given by the convolution of filter $\mathbf{f}$ with the impulse response of the channel, the central coefficient g_Δ then being forced to zero.

•*SISO demapping*

The role of this module is to convert the equalized data z_i into extrinsic LLRs on the interleaved coded bits, which will be then transmitted to the channel

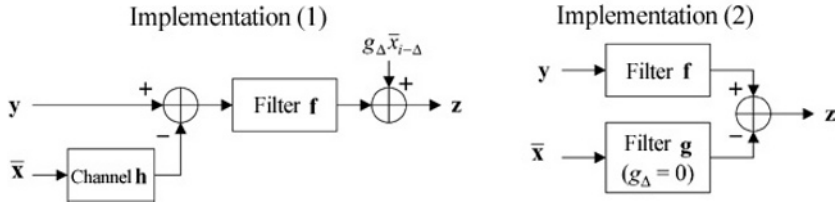

Figure 11.15 – Practical implementation of the equalizer using transverse filters.

decoder. Generally, we can always decompose the expression of z_i as the sum of two quantities:

$$z_i = g_\Delta x_{i-\Delta} + \nu_i \tag{11.46}$$

The term $g_\Delta x_{i-\Delta}$ represents the useful signal up to a constant factor g_Δ. We recall that this factor quite simply corresponds to the central coefficient of the cascading of the channel with the equalizer. The term ν_i accounts for both residual interference and noise at the output of the equalizer. In order to perform the demapping operation, we make the hypothesis[9] that interference term ν_i follows a complex Gaussian distribution, with zero mean and total variance σ_ν^2. Parameters g_Δ and σ_ν^2 are easy to deduce from the knowledge of the set of equalizer coefficients. We can thus show ([11.51, 11.33]) that we have:

$$g_\Delta = \mathbf{f^T h_\Delta} \text{ and } \sigma_\nu^2 = E\left\{ |z_i - g_\Delta x_{i-\Delta}|^2 \right\} = \sigma_x^2 g_\Delta (1 - g_\Delta) \tag{11.47}$$

Starting from these hypotheses, the demapping module calculates the *a posteriori* LLR on the coded interleaved bits, denoted $L(x_{i,j})$ and defined as follows:

$$L(x_{i,j}) = \ln \left(\frac{\Pr(x_{i,j} = 1 \,|z_i)}{\Pr(x_{i,j} = 0 \,|z_i)} \right) \tag{11.48}$$

The values present in the numerator and denominator can be evaluated by summing the *a posteriori* probability $\Pr(x_i = X_l | z_i)$ of having transmitted a particular symbol X_l of the constellation on all the symbols for which the j-th bit making up this symbol takes the value 0 or 1 respectively. Thus, we can write:

[9] This hypothesis rigorously only holds on condition that the equalizer suppresses all the ISI, which assumes perfect knowledge of the transmitted data. Nevertheless, it is a good approximation in practice, particularly in a turbo equalization context where the reliability of the decisions at the output of the decoder increases along the iterations which, in its turn, improves the equalization operation.

$$\begin{aligned} L(x_{i,j}) &= \ln\left(\frac{\sum\limits_{x_l/X_{l,j}=1} \Pr(x_i=X_l|z_i)}{\sum\limits_{x_l/X_{l,j}=0} \Pr(x_i=X_l|z_i)}\right) \\ &= \ln\left(\frac{\sum\limits_{x_l/X_{l,j}=1} P(z_i|x_i=X_l)P_a(X_l)}{\sum\limits_{x_l/X_{l,j}=0} P(z_n|x_i=X_l)P_a(X_l)}\right) \end{aligned} \tag{11.49}$$

The second equality results from applying Bayes' relation. It shows the *a priori* probability $P_a(X_l) = \Pr(x_i = X_l)$ of having transmitted a given symbol X_l of the modulation alphabet. This probability is calculated from the *a priori* information available at the input of the equalizer (relations (11.25) and (11.26)). By exploiting the above hypotheses, the likelihood of observation z_i conditionally to the hypothesis of having transmitted the symbol X_l at instant i can be written:

$$P(z_i\,|x_i\,=X_l) = \frac{1}{\pi\sigma_\nu^2}\exp\left(-\frac{|z_i - g_\Delta X_l|^2}{\sigma_\nu^2}\right) \tag{11.50}$$

After simplification, the *a posteriori* LLR calculated by the demapping operation becomes:

$$L(x_{i,j}) = \ln\left(\frac{\sum\limits_{X_l/X_{l,j}=1} \exp\left(-\frac{|z_i-g_\Delta X_l|^2}{\sigma_\nu^2} + \sum\limits_{k=1}^{m} X_{l,k}L_a(x_{i,k})\right)}{\sum\limits_{X_l/X_{l,j}=0} \exp\left(-\frac{|z_i-g_\Delta X_l|^2}{\sigma_\nu^2} + \sum\limits_{k=1}^{m} X_{l,k}L_a(x_{i,k})\right)}\right) \tag{11.51}$$

Like in the case of the BCJR-MAP equalizer, we can factorize in the numerator and denominator the *a priori* information term in relation to the considered bit, in order to obtain the extrinsic information that is then provided to the decoder:

$$L(x_{i,j}) = L_a(x_{i,j}) + \underbrace{\ln\left(\frac{\sum\limits_{X_l/X_{l,j}=1} \exp\left(-\frac{|z_i-g_\Delta X_l|^2}{\sigma_\nu^2} + \sum\limits_{k\neq j} X_{l,k}L_a(x_{i,k})\right)}{\sum\limits_{X_j/X_{j,i}=0} \exp\left(-\frac{|z_i-g_\Delta X_l|^2}{\sigma_\nu^2} + \sum\limits_{k\neq j} X_{l,k}L_a(x_{i,k})\right)}\right)}_{L_e(x_{i,j})} \tag{11.52}$$

Finally, the extrinsic information is obtained quite simply by subtracting the *a priori* information from the *a posteriori* LLR calculated by the equalizer:

$$L_e(x_{i,j}) = L(x_{i,j}) - L_a(x_{i,j}) \tag{11.53}$$

In the particular case of BPSK modulation, the SISO demapping equations are simplified to give the following expression of the extrinsic LLR:

$$L(x_i) = \frac{4}{1-g_\Delta} Re\,\{z_i\} \tag{11.54}$$

When Gray mapping rules are used, experience shows that we can reduce the complexity of the demapping by ignoring the *a priori* information coming from the decoder in the equations above[10], without really affecting the performance of the device. On the other hand, this simplification no longer applies when we consider other mapping rules, like the *Set Partitioning* rule used in coded modulation schemes. This point has been particularly well highlighted in [11.14] and [11.30].

This completes the description of the soft-input soft-output linear MMSE equalizer. Finally, we can note that unlike the BCJR-MAP equalizer, the complexity of the SISO mapping and demapping operations increases linearly (and not exponentially) as a function of size M of the constellation and of the number L of taps in the impulse response of the discrete-time equivalent channel model.

Adaptive implementation of the equalizer

Historically, the first MMSE turbo equalizer was proposed in 1997, directly in adaptive form [11.20, 11.32]. The closed-form expression (11.40) enabling the computation of the equalizer coefficients from the knowledge of the channel impulse response was not known at that time. The chosen solution thus involved determining the filter coefficients in an adaptive manner, using stochastic gradient descent algorithms aiming at minimizing the mean square error between the transmitted data and the equalizer output. As we shall see in the following, when evaluating performance, the adaptive MMSE turbo equalizer remains a very interesting solution for time-invariant or slowly time-varying channels. The purpose of this section is to show that, for such channels, the adaptive MMSE equalizer and the MMSE equalizer proposed in (11.40) have similar performance and characteristics.

The structure of the considered equalizer is shown in Figure 11.15 (implementation (2)). An adaptive procedure is used to obtain the filters' coefficients. This adaptive algorithm is composed of two distinct phases: the training phase and the tracking phase. The training phase makes use of sequences known by the receiver (training sequences) to initialize the equalizer coefficients. Next, during the tracking period, the coefficients are continuously updated in a decision-directed manner, based on the receiver estimate of the transmitted sequence.

Adaptive algorithms involve determining, for each symbol entering the equalizer, output z_i from the following relation:

$$z_i = \mathbf{f}_i^{\mathbf{T}} \mathbf{y}_i - \mathbf{g}_i^{\mathbf{T}} \tilde{\mathbf{x}}_i \tag{11.55}$$

where $\mathbf{y}_i = (y_{i+F}, \ldots, y_{i-F})^T$ is the vector of channel output samples and $\tilde{\mathbf{x}}_i = (\bar{x}_{i+G}, \ldots, \bar{x}_{i-\Delta+1}, 0, \bar{x}_{i-\Delta-1}, \ldots, \bar{x}_{i-G})^T$ is the vector of estimated symbols, with respective lengths $2F+1$ and $2G+1$. Note that the coordinate relative

[10] This amounts to assuming the transmitted symbols to be equiprobable, i.e. to putting $P_a(X_l) = 1/M$ whatever the symbol and the iteration considered.

to the soft estimate $\bar{x}_{i-\Delta}$ in $\tilde{\mathbf{x}}_i$ is set to zero in order not to cancel the signal of interest. Vectors $\mathbf{f}_i = (f_{i,F}, \ldots, f_{i,-F})^T$ and $\mathbf{g}_i = (g_{i,G}, \ldots, g_{i,-G})^T$ represent the coefficients of filters $\mathbf{f}$ and $\mathbf{g}$, respectively. Both vectors are a function of time since they are updated at each new received symbol.

The relations used to update the vectors of the coefficients can be obtained from a least-mean square (LMS) gradient algorithm:

$$\begin{array}{rcl} \mathbf{f}_{i+1} & = & \mathbf{f}_i - \mu\,(z_i - x_{i-\Delta})\,\mathbf{y}_i^* \\ \mathbf{g}_{i+1} & = & \mathbf{g}_i - \mu\,(z_i - x_{i-\Delta})\,\tilde{\mathbf{x}}_i^* \end{array} \tag{11.56}$$

where μ is a small, positive, step-size that controls the convergence properties of the algorithm.

During the first iteration of the turbo equalizer, $\tilde{\mathbf{x}}_i$ is a vector all the components of which are null; the result is that the coefficients vector $\mathbf{g}_i$ is also null. The MMSE equalizer then converges adaptively towards an MMSE transversal equalizer. When the estimated data are very reliable and close to the transmitted data, the MMSE equalizer converges towards an ideal (genie) interference canceller, then having the performance of a transmission without intersymbol interference. The limiting forms of the adaptive equalizer are therefore totally identical to those obtained in (11.43) and (11.44), on condition of course that the adaptive algorithm can converge towards a local minimum close to the optimal solution.

Note, however, that for intermediate iterations where the estimated information symbols $\bar{x}_i$ are neither null nor perfect, filter $\mathbf{g}_i$ must not be fed directly with the transmitted symbols otherwise the equalizer will converge towards the solution of the genie interference canceller, which is not the aim searched for. To enable the equalizer to converge towards the targeted solution, the idea here involves providing filter $\mathbf{g}_i$ with soft estimates built from the transmitted symbols, during the training periods:

$$(\bar{x}_i)_{its} = \sigma_{\bar{x}} x_i + \sqrt{1 - \sigma_{\bar{x}}^2}\,\eta_i \tag{11.57}$$

where $\sigma_{\bar{x}}^2$ corresponds to the variance of the soft estimates $\bar{x}_i$ obtained from (11.24) and η_i a zero-mean complex circularly-symmetric additive white Gaussian noise with unit variance.

In the tracking period and in order to enable the equalizer to follow the variations of the channel, it is possible to replace the transmitted symbols x_i in relations (11.56) by the decisions $\hat{x}_i$ at the output of the equalizer, or by the decisions on the estimated symbols $\bar{x}_i$.

When the SISO MMSE equalizer is realized in adaptive form, we do not explicitly have access to the channel impulse response, and the updating relation of $\mathbf{g}_i$ does not enable g_Δ to be obtained since component $g_{i,\Delta}$ is constrained to be zero. To perform the SISO demapping operation, we must however estimate both bias g_Δ on the data z_n provided by the equalizer and variance σ_ν^2 of the

residual interference at the output of the equalizer. As we will see, these two parameters can be estimated from the output of the equalizer. From relation (11.46) again, we can show the general following result:

$$E\left(|z_i|^2\right) = g_\Delta \sigma_x^2 \tag{11.58}$$

Assuming that the variance of the transmitted data is normalized to unity, an estimate of g_Δ is given by:

$$\hat{g}_\Delta = \frac{1}{N} \sum_{i=0}^{N-1} |z_i|^2 \tag{11.59}$$

Once we have estimated g_Δ, we immediately deduce the value of σ_ν^2 thanks to relation (11.47):

$$\sigma_\nu^2 = \sigma_x^2 \hat{g}_\Delta (1 - \hat{g}_\Delta) \tag{11.60}$$

One particularity of adaptive MMSE turbo equalization concerns the determination of the estimated symbols. Indeed, in accordance with the remarks of [11.20] and [11.55], using *a posteriori* information instead of extrinsic information at the output of the channel decoder in (11.27) can yield significant performance improvement.

We have therefore defined an adaptive MMSE turbo equalizer whose coefficients are obtained from a low complexity stochastic gradient descent algorithm, making it possible to track the slow time variations of the transmission channel. A drawback of this technique lies in the necessity to transmit training sequences, which lower the spectral efficiency. The size of training sequences can be significantly reduced by considering self-learning or blind algorithms. In particular, the equalizer in the first iteration can be advantageously replaced by a self-learning equalizer called *Self Adaptive Decision-Feedback Equalizer* (SADFE) [11.32] that requires a very small transmission overhead. The work of Hélard *et al.* [11.26] has shown that such a turbo equalizer can reach performance virtually identical to that of the adaptive MMSE turbo equalizer with learning sequence, while operating at a higher spectral efficiency. On the other hand, a higher number of iterations is then required.

Examples of performance

For comparison purposes, the performance of the MMSE turbo equalizer has been simulated by considering the same transmission scenario as for the turbo MAP equalizer.

First, the parameters of the channel are assumed to be perfectly estimated. The coefficients are calculated once per frame by matrix inversion, by considering a digital filter with $F = 15$ coefficients and a designed delay $\Delta = 9$. The simulation results, obtained after 10 iterations, are presented in Figure 11.16.

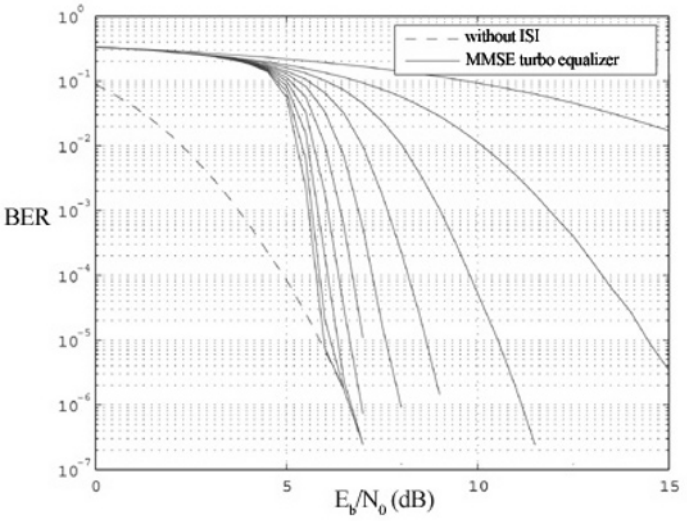

Figure 11.16 – Performance of the MMSE turbo equalizer for BPSK transmission on the Proakis C channel, with a 4-state rate $R = 1/2$ non-recursive non-systematic convolutional code and a 16384 bit pseudo-random interleaver.

Convergence of the iterative process occurs here at threshold signal to noise ratio of 4 dB, and the turbo equalizer suppresses all the ISI beyond a signal to noise ratio of 6 dB (after 10 iterations). Compared to the results obtained with the MAP turbo equalizer (Figure 11.13), we can therefore make the following remarks:

1. The convergence occurs later with MMSE turbo equalization (of the order of 1 dB here, compared to MAP turbo equalization).

2. The MMSE turbo equalizer requires more iterations than the MAP turbo equalizer to reach comparable error rates.

However, the MMSE turbo equalizer here shows its capacity to suppress all the ISI when the signal to noise ratio is high enough, even on a channel that is known to be difficult to equalize. It is therefore a serious alternative solution to the MAP turbo equalizer when the latter cannot be used for reasons of complexity.

Second, the hypothesis of perfect knowledge of the channel parameters has been removed and the turbo equalizer is simulated in the adaptive form, keeping the same transmission parameters. The communication begins with the transmission of an initial training sequence of 16384 symbols assumed to be perfectly known by the receiver. Then, frames composed of 1000 training symbols followed by 16384 information symbols are periodically sent into the channel.

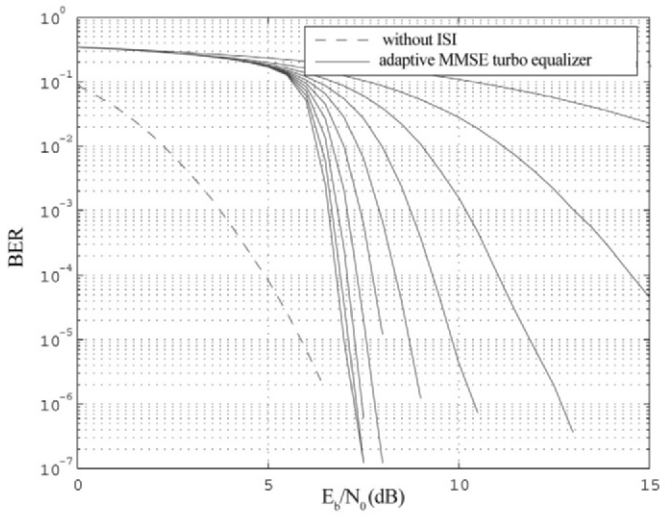

Figure 11.17 – Performance of the adaptive MMSE turbo equalizer for BPSK transmission on the Proakis C channel, with a 4-state rate $R = 1/2$ non-recursive non-systematic convolutional code and a 16384 bit pseudo-random interleaver.

During the processing of the 16384 information symbols, the turbo equalizer operates in a decision-directed manner. The equalizer filters each have 21 coefficients ($F = G = 10$). The coefficients are updated using the LMS algorithm. The step size is set to $\mu = 0,0005$ during the training period, and then to $\mu = 0,000005$ during the tracking period. Simulation results are given in Figure 11.17, considering 10 iterations at reception. We observe a degradation of the order of only 1 dB compared to the ideal situation where the channel is assumed to be perfectly known. We note that when the channel is estimated and used for the direct computation of the coefficients of the MMSE equalizer, losses in performance will also appear, which reduces the degradation in comparison to the ideal situation of Figure 11.16. Note also that, to track the performance of Figure 11.17, we have not taken into account the loss in the signal to noise ratio caused by the use of training sequences.

In the light of these results, we note that the major difference between adaptive MMSE turbo equalization and that which uses direct computation of the coefficients from the estimate of the channel lies in the way the filter coefficients are determined, since the structure and the optimization criterion of the equalizers are identical.

To finish, we point out that, in the same way as for the turbo MAP equalizer, we can use EXIT charts to predict the theoretical convergence threshold of the

MMSE turbo equalizer, under the hypothesis of ideal interleaving. The reader will find further information on this subject in [11.8] or [11.50], for example.

Example of implementation and applications

The implementation of an MMSE turbo equalizer on a signal processor was reported in [11.9]. The target was the TMS320VC5509 processor by Texas Instruments. This is a 16-bit fixed-point DSP with low power consumption, which makes it an ideal candidate for mobile receivers. The considered transmission scheme included a 4-state rate 1/2 convolutional encoder and a 1024 bit interleaver followed by a QPSK modulator. The whole turbo equalizer was implemented in C language on the DSP, with the exception of some processing optimized in assembly (filtering and FFT) provided by a specialized library. The equalizer included 32 coefficients. The decoding was performed using the Max-Log-MAP algorithm. The simulation results showed that, subject to carefully choosing the representation in fixed decimal points of the data handled (within the limit of 16 bits maximum granted by the DSP), data quantization did not cause any loss in performance in comparison with the corresponding floating-point receiver. The final data rate obtained was of the order of 42 kbits/s after 5 iterations, which shows the feasibility of such receivers using current technology. The challenge now involves defining appropriate circuit architectures, capable of operating at several Mbits/s, in order to respond to emerging demands for high data rate services.

MMSE turbo equalization is a relatively recent technology. Therefore, at the moment of writing this book, there have been few studies on the potential applications of this technique at reception. Generally, resorting to MMSE turbo equalization is an effective solution in the context of high data rate transmissions on highly frequency-selective channels. In particular, this system has shown excellent performance on the ionospheric channel typically used in the context of HF military communications. Indeed, the long echoes produced by this channel prevent the use of MAP equalizers. On the other hand, conventional linear equalization schemes do not make it possible to reach a transmission quality acceptable when high-order modulations (*e.g.* 8-PSK or 16-QAM) are considered. MMSE turbo equalization is thus an attractive solution to the problem of increasing the data rate of military transmissions. In the context of HF communications, the interest of MMSE turbo equalization for high spectral efficiency modulations has been validated by the work of Langlais [11.29] and Otnes [11.41], which shows that this technique can offer gains up to 5 dB compared to conventional receivers. To our knowledge, MMSE turbo equalization has not yet been implemented in standardized modems. However, it is important to note that this reception technique enables the transmission performance to be notably improved while keeping standardized transmitters.

11.2 Multi-user turbo detection and its application to CDMA systems

11.2.1 Introduction and some notations

In a Code Division Multiple Access (CDMA) system, such as the one shown in Figure 11.18, user k ($1 \leqslant k \leqslant K$) transmits a sequence of binary elements $\{d_k\} = \pm 1$ with an amplitude A_k. For each of the users, a channel encoder (CC_k) is used, followed by an external interleaver (π_k) before the spreading operation (multiplication by size N, normalized spreading code $\mathbf{s}_k$,) which provides binary symbols called *chips*. This code can vary at each symbol time.

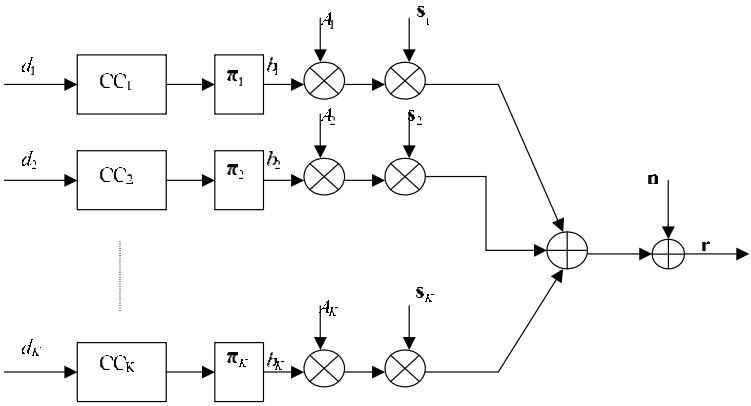

Figure 11.18 – CDMA transmitter.

The received signal, $\mathbf{r}$, can be written in matrix form by:

$$\mathbf{r} = \mathbf{SAb} + \mathbf{n} \tag{11.61}$$

where:

- $\mathbf{S}$ is the $N \times K$ matrix formed by the normalized codes of each user (the k-th column represents the k-th code $\mathbf{s}_k$ whose norm is equal to unity),
- $\mathbf{A}$ is a diagonal $K \times K$ matrix made up of the amplitudes A_k of each user.
- $\mathbf{b}$ is the vector of dimension K made up of the elements transmitted after coding channel by the K users.
- $\mathbf{n}$ is the N-dimensional centred Gaussian vector with covariance matrix $\sigma^2 \mathbf{I}_N$.

The source data rates of the different users can be different. The size of the spreading code is such that the chip data rate (after spreading) is the same for all

users. The received signal $\mathbf{r}$ is given by the contribution of all the K users plus a centred AWGN with variance σ^2. From observation $\mathbf{r}$, we wish to recover the information bits d_k of each user. Figure 11.19 gives the diagram of the receiver using a turbo CDMA type technique to jointly process the multi-detection and the channel decoding:

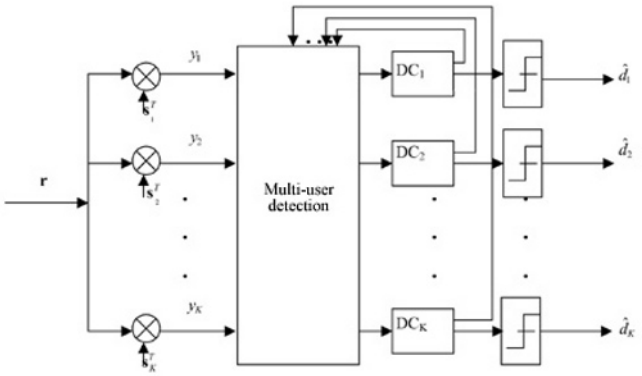

Figure 11.19 – Turbo CDMA receiver.

11.2.2 Multi-user detection

This section presents the main multi-user detection methods. In order to simplify the description of these methods, only the case of synchronous transmissions over Gaussian channels is considered.

Standard receiver

The simplest (conventional or standard) detector is the one which operates as if each user was alone on the channel. The receiver is quite simply made up of the filter adapted to the signature of the user concerned (this operation is also called despreading), see Figure 11.20.

At the output of the adapted filter bank the signal can be written in the form:

$$\mathbf{y} = \mathbf{S}^T\mathbf{r} = \mathbf{RAb} + \mathbf{S}^T\mathbf{n} \tag{11.62}$$

We note that the vector of the additive noise at the output of the adapted filter bank, is made up of correlated components. Its covariance matrix depends directly on the intercorrelation matrix of the spreading sequences used, $\mathbf{R} = \mathbf{S}^T\mathbf{S}$. We have $\mathbf{S}^T\mathbf{n} \sim N(0, \sigma^2\mathbf{R})$.

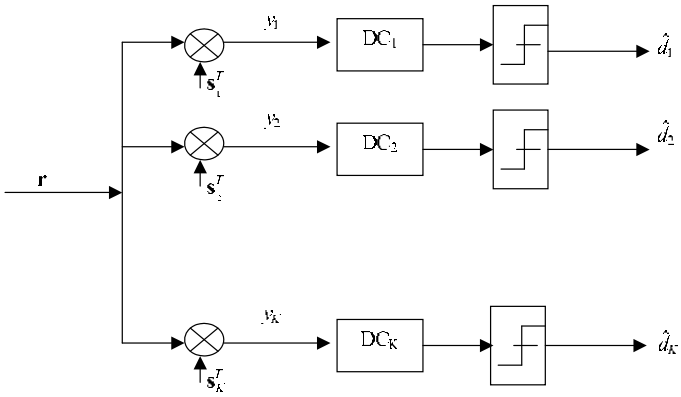

Figure 11.20 – Standard detector.

We can show that the error probability (before channel decoding) for the k-th user can be written in the form:

$$P_{e,k} = P\left(\hat{b}_k \neq b_k\right) = \frac{1}{2^{K-1}} \sum_{\mathbf{b}_{-k} \in \{-1,+1\}^{K-1}} Q\left(\frac{A_k}{\sigma} + \sum_{j \neq k} b_j \frac{A_j}{\sigma} \rho_{jk}\right) \quad (11.63)$$

where $\rho_{jk} = \mathbf{s}_j^T \mathbf{s}_k$ measures the intercorrelation between the codes of users j and k, with $\mathbf{b}_{-k} = (b_1, b_2, \cdots, b_{k-1}, b_{k+1}, \cdots, b_K)$.

Assuming that the spreading codes used are such that the intercorrelation coefficients are constant and equal to 0.2, Figure 11.21 gives the performance of the standard receiver, in terms of error probability of the first user as a function of the signal to noise ratio, for a number of users varying from 1 to 6. The messages of all the users are assumed to be received with the same power. We note of course that the higher the number of users, the worse the performance. This error probability can even tend towards 1/2, while the signal to noise ratio increases if the following condition (*Near Far Effect*) is not satisfied:

$$A_k > \sum_{j \neq k} A_j \left|\rho_{jk}\right|$$

.

Optimal joint detection

Optimal joint detection involves maximizing the *a posteriori* probability (probability of vector $\mathbf{b}$ conditionally to observation $\mathbf{y}$). If we assume that the binary elements transmitted are equiprobable and given that $\mathbf{y} = \mathbf{S}^T\mathbf{r} = \mathbf{RAb} + \mathbf{S}^T\mathbf{n}$ with $\mathbf{S}^T\mathbf{n} \sim N(0, \sigma^2\mathbf{R})$, we can deduce that the optimal joint detection is given

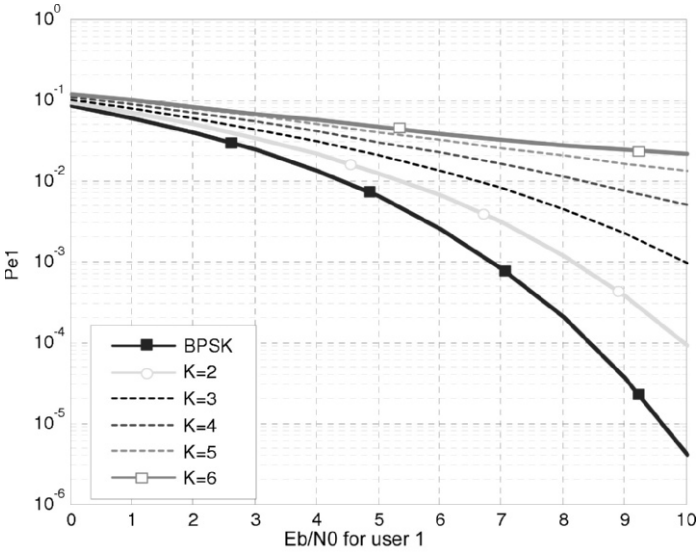

Figure 11.21 – Error probability of the first user as a function of the signal to noise ratio $Eb/N0$ for constant intercorrelation coefficients $\rho = 0.2$, for K=6 users sharing the resource.

by the equivalences:

$$\text{Max}_{\mathbf{b}}\left(f_{\mathbf{y}}^{\mathbf{b}}(\mathbf{y})\right) \Leftrightarrow \text{Min}_{\mathbf{b}}\left(\|\mathbf{y}-\mathbf{RAb}\|^2_{\mathbf{R}^{-1}}\right) \Leftrightarrow \text{Max}_{\mathbf{b}}\left(2\mathbf{b}^T\mathbf{Ay}-\mathbf{b}^T\mathbf{ARAb}\right) \tag{11.64}$$

An exhaustive search for the optimal solution is relatively complex.

Decorrelator detector

Decorrelator detectors involves multiplying observation $\mathbf{y}$ by the inverse of the intercorrelation matrix of the codes: $\mathbf{R}^{-1}\mathbf{y} = \mathbf{Ab} + \mathbf{R}^{-1}\mathbf{S}^T\mathbf{n}$. This equation shows that the decorrelator enables the multiple access interference to be cancelled completely, which makes it robust in relation to the *Near Far Effect*. On the other hand, the resulting additive Gaussian noise has greater variance. Indeed, we have: $\mathbf{R}^{-1}\mathbf{S}^T\mathbf{n} \sim N(0, \sigma^2\mathbf{R}^{-1})$. The error probability of the k-th user can then be written in the form:

$$P_{e,k} = P\left(\hat{b}_k \neq b_k\right) = Q\left(\frac{A_k}{\sigma\sqrt{(\mathbf{R}^{-1})_{kk}}}\right) \tag{11.65}$$

Linear MMSE detector

The MMSE detector involves finding the transformation $\mathbf{M}$ that minimizes the mean squared error: $\text{Min}_{\mathbf{M}\in R^{K\times K}} E\left(\|\mathbf{b}-\mathbf{My}\|^2\right)$. This transformation is no

other than:

$$\mathbf{M} = \mathbf{A}^{-1}\left(\mathbf{R} + \sigma^2\mathbf{A}^{-2}\right)^{-1} \tag{11.66}$$

and consequently, the error probability of the k-th user can be written in the form:

$$P_{e,k} = \frac{1}{2^{K-1}} \sum_{\mathbf{b}_{-k}\in\{-1,+1\}^{K-1}} Q\left(\frac{A_k}{\sigma}\frac{(\mathbf{MR}_{k,k})}{\sqrt{(\mathbf{MRM})_{k,k}}}\left(1+\sum_{j\neq k}\frac{(\mathbf{MR}_{k,j})A_j b_j}{(\mathbf{MR}_{k,k})A_k}\right)\right) \tag{11.67}$$

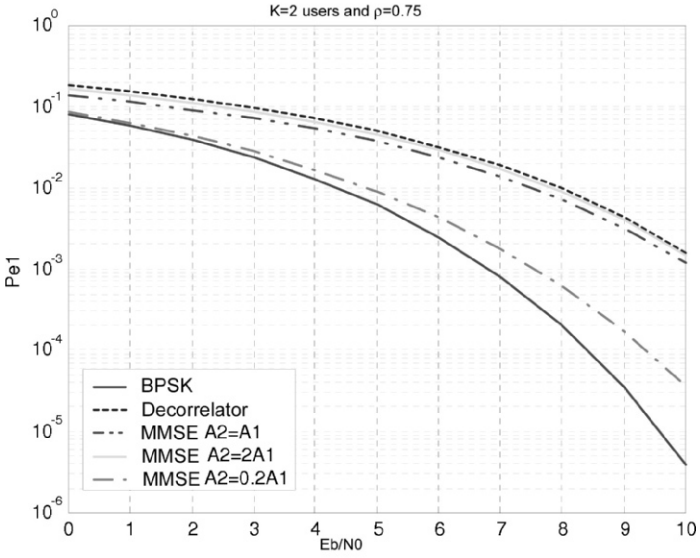

Figure 11.22 – Comparison between the decorrelator method and MMSE.

In order to compare the two techniques, the decorrelator detector and MMSE, the two receivers were simulated with 2 users whose spreading codes are highly correlated (intercorrelation coefficients equal to 0,75). Figure 11.22 shows the curves of the BER for the first user and parametred by the power of the 2nd user (or the user's amplitude). The performance of the MMSE receiver is always better than that of the decorrelator. For low power of the 2nd user, performance is close to that of the single-user. However, for a high power of the 2nd user, the MMSE performance will tend towards that of the decorrelator.

Iterative detector

Decorrelator receivers or MMSE receivers can be implemented with iterative matrix inversion (Jacobi, Gauss-Siedel, or relaxation) methods. The Jacobi method

leads to the *Parallel Interference Cancellation*(PIC) method. The Gauss-Siedel method leads to the *Successive Interference Cancellation* (SIC) method. Figure 11.23 gives the diagram for implementing the *SIC* (with $K = 4$ users and $M = 3$ iterations) where $ICU_{m,k}$ is the *interference cancellation unit* (or ICU) for the k-th user at iteration m (see Figure 11.24). The binary elements are initialized to zero: at iteration $m = 0 \quad b_{0,k} = 0$ for $k = 1, \cdots, K$.

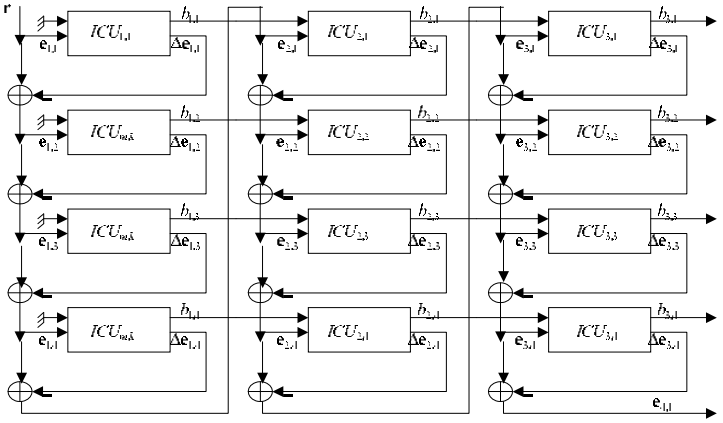

Figure 11.23 – Iterative SIC (*Successive Interference Cancellation*) detector, with $K = 4$ users and $M = 3$ iterations.

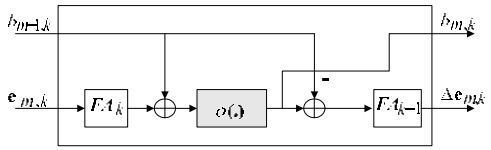

Figure 11.24 – Interference cancellation unit $ICU_{m,k}$ for user k, at iteration m.

Function FA_k (respectively FA_k^{-1}) is the despreading (respectively spreading) of the k-th user. Function $\phi(.)$ can be chosen as a non-linear function or quite simply as being equal to the identity function (see also the choice of $\phi(.)$ in the case of turbo CDMA). If we choose the identity function, the ICU unit of Figure 11.24 can of course be simplified and is easy to define. In this case, we can verify that, for user k and at iteration m, the output of the receiver can be written as the result of linear filtering:

$$b_{m,k} = \mathbf{s}_k^T \prod_{j=1}^{k-1} \left(\mathbf{I} - \mathbf{s}_j \mathbf{s}_j^T\right) \sum_{p=0}^{m-1} \mathbf{\Phi}_K^p \mathbf{r} = \mathbf{g}_{m,k}^T \mathbf{r} \tag{11.68}$$

with:

$$\mathbf{\Phi}_K = \prod_{j=1}^{K} \left(I - \mathbf{s}_j \mathbf{s}_j^T\right) \tag{11.69}$$

We can show that the error probability for the k-th user at iteration m can be written in the following form, where $\mathbf{S}$ is the matrix of the codes and $\mathbf{A}$ is the diagonal matrix of the amplitudes:

$$P_e(m,k) = \frac{1}{2^{K-1}} \sum_{\mathbf{b}/b_k=+1} Q\left(\frac{\mathbf{g}_{m,k}^T \mathbf{SAb}}{\sigma\sqrt{\mathbf{g}_{m,k}^T \mathbf{g}_{m,k}}}\right) \tag{11.70}$$

Figure 11.25 gives an example of simulations of the SIC method with 5 users, a spreading factor of 20 and an intercorrelation matrix given by:

$$\mathbf{R} = \begin{pmatrix} 1 & 0,3 & 0 & 0 & 0 \\ 0,3 & 1 & 0,3 & 0,3 & 0,1 \\ 0 & 0,3 & 1 & 0 & -0,2 \\ 0 & 0,3 & 0 & 1 & 0 \\ 0 & 0,1 & -0,2 & 0 & 1 \end{pmatrix}$$

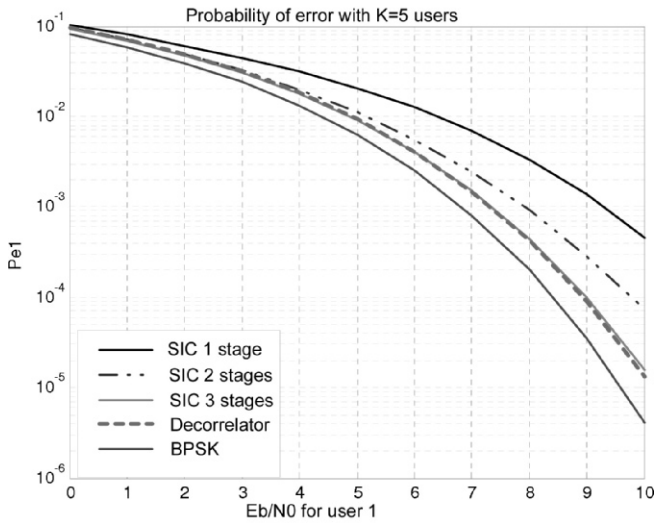

Figure 11.25 – Simulation of a SIC receiver.

We note that after 3 iterations, the SIC converges towards the result obtained with the decorrelator (we can prove mathematically that the SIC converges towards this result when the number of iterations M tends towards infinity).

11.2.3 Turbo CDMA

Several turbo CDMA type techniques have been proposed to jointly process multi-detection and channel decoding:

- Varanasi and Guess [11.53] have proposed (hard estimation) decoding and immediately recoding each user before subtracting this contribution from the received signal. The same operation is performed on the residual signal to decode the information of the second user, and so on, until the final user.
- Reed and Alexander [11.46] have proposed to use an adapted filter bank followed (in parallel) by different decoders before subtracting, for each user, the multiple access interference linked to the $K-1$ other users.
- Wang and Poor [11.56] have proposed a multi-user detector that involves implementing in parallel the MMSE filters associated with each user, followed by the corresponding channel decoders. These two elements exchange their extrinsic information iteratively.
- Tarable *et al.* [11.48] have proposed a simplification of the method presented in [11.56]. For the first iterations, an MMSE type multi-user detector is used, followed by channel decoders placed in parallel. For the final iterations, the MMSE filter is replaced by an adapted filter bank.

Turbo SIC detector

In this section, channel decoding is introduced into a new successive interference cancellation (SIC) structure. Figure 11.23 remains valid, only units $ICU_{m,k}$ change. Each interference cancellation unit $ICU_{m,k}$, relative to the k-th user and at iteration m, is given in Figure 11.26. The originality lies in the way in which this unit is designed: the residual error signal $\mathbf{e}_{m,k}$ is despread (by $\mathbf{s}_k$) then deinterleaved (π_k^{-1}) before adding the weighted estimation of the $b_{m-1,k}$ data of the same user calculated at the previous iteration. The signal thus obtained, $y_{m,k}$, passes through the channel decoder that provides the *a posteriori* log likelihood ratio, conditionally to the whole observation, of all the binary elements (both for the information bits and the parity bits):

$$\mathrm{LLR}(b_k/y_{m,k}) = \log\left(\frac{P\left[b_k=+1/y_{m,k}\right]}{P\left[b_k=-1/y_{m,k}\right]}\right) \tag{11.71}$$

This ratio is then transformed into a weighted estimation of the binary elements:

$$\tilde{b}_{m,k} = E\left[b_k/y_{m,k}\right] = \tanh\left(\frac{1}{2}\mathrm{LLR}\left(b_k/y_{m,k}\right)\right) \tag{11.72}$$

The soft estimation of user k at iteration m is given by $b_{m,k} = A_k\tilde{b}_{m,k}$. The difference $(b_{m,k} - b_{m-1,k})$ is interleaved by π_k before spreading by $\mathbf{s}_k$. The result

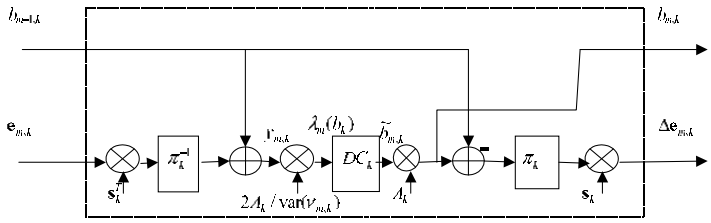

Figure 11.26 – Interference cancellation unit for the turbo SIC decoder in CDMA for the k-th user and at iteration m.

thus obtained $\Delta\mathbf{e}_{m,k}$ is subtracted from residual signal $\mathbf{e}_{m,k}$ to obtain the new residual signal $\mathbf{e}_{m,k+1}$ of the following user (if $k < K$) or to obtain the new residual signal $\mathbf{e}_{m+1,1}$ for the first user at the following iteration ($\mathbf{e}_{m,K+1} = \mathbf{e}_{m+1,1}$). Here, $y_{m,k}$ is written in the form $y_{m,k} = A_k b_k + \nu_{m,k}$ where $\nu_{m,k}$ (residual multiple access interference plus the additive noise) is approximated by a centred Gaussian random variable whose variance is given by:

$$\mathrm{var}\,(\nu_{m,k}) = \sum_{i<k} A_i^2 \rho_{i,k}^2 \left(1 - \tilde{b}_{m,i}^2\right) + \sum_{i>k} A_i^2 \rho_{i,k}^2 \left(1 - \tilde{b}_{m-1,i}^2\right) + \sigma^2 \qquad (11.73)$$

We show that the extrinsic information of user k at iteration m is given by:

$$\lambda_m(b_k) = \log\left(\frac{P\left[y_{m,k}/b_k = +1\right]}{P\left[y_{m,k}/b_k = -1\right]}\right) = \frac{2y_{m,k}A_k}{\mathrm{var}(\nu_{m,k})} \qquad (11.74)$$

This extrinsic information serves as the input at the decoder associated with the k-th user.

Some simulations

To give an idea of the performance of the turbo SIC decoder, Gold sequences of size 31 are generated. The channel turbo encoder (rate $R = 1/3$) normalized for UMTS [11.1] is used. We consider frames of 640 bits per user. The external interleavers of the different users are produced randomly. The BER and PERs are averaged over all the users. For the channel turbo encoder, the Max-Log-MAP algorithm is used, 8 being the number of iterations internal to the turbo decoder. Figure 11.27(a) gives the performance of the turbo SIC decoder for one, two and three iterations with $K = 31$ users (that is, 100% load rate) having the same power. The performance of the single-user detector and of the conventional detector are also indicated. Figure 11.27(b) shows performance in terms of PER.

Turbo SIC/RAKE detector

In the case where the propagation channel of the k-th user has an impulse response with multiple paths $c_k(t)$, it suffices to replace the despreading function

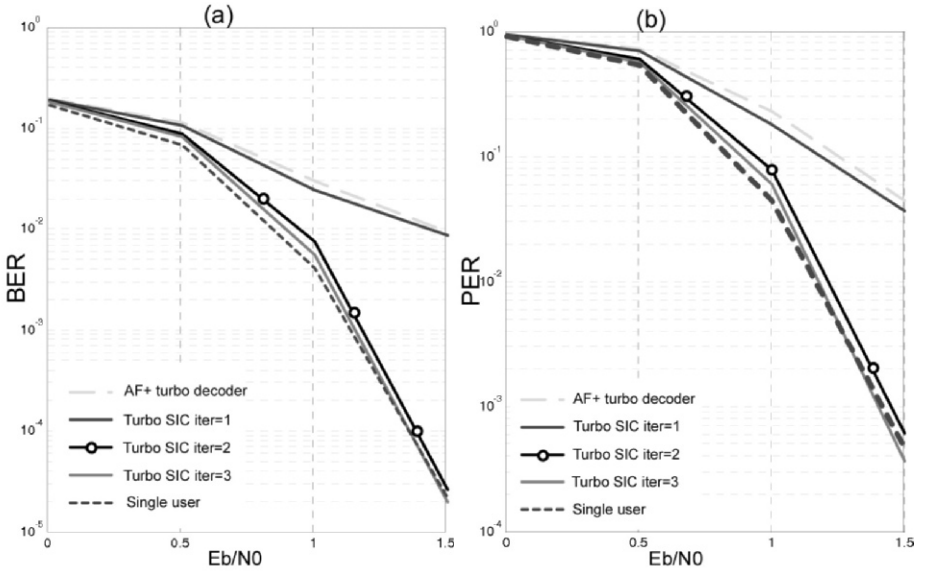

Figure 11.27 – Performance of the turbo SIC decoder: (a) mean Binary Error Rates (BER) (b) mean Packet Error Rates (PER). $K = 31$ users, spreading factor of 31, with frame size 640 bits.

by a RAKE filter (filter adapted to the spreading sequence convolved with the transfer function $c_k(t)$ in the $ICU_{m,k}$) unit, and to replace the spreading function by the spreading function convolved by $c_k(t)$. This new structure is called a turbo SIC/RAKE decoder.

The turbo *SIC/RAKE* decoder is used particularly in the context of the uplink in the UMTS-FDD system.

11.3 Conclusions

In this chapter, we have presented the first two systems to have benefited from applying the turbo principle to a context other than error correction coding. In the first part, we have described the principle of turbo equalization, which relies on an iterative exchange of probabilistic information between a SISO equalizer and a SISO decoder. The SISO equalizer can take different forms according to the chosen optimization criterion. We have presented two types of SISO equalizers: the BCJR-MAP equalizer, operating on the trellis representation of the ISI channel, and the MMSE equalizer, which uses linear filtering. The MAP turbo equalizer leads to excellent performance compared to the conventional receiver. However, this approach is often avoided in practice since it leads to a very high computation cost. We have discussed several solutions for reducing

the complexity of the BCJR-MAP equalizer. As for the MMSE turbo equalizer, it offers a good compromise between performance and complexity. For many transmission configurations it leads to performance close to that offered by the BCJR-MAP turbo equalizer, with reasonable complexity. In addition, unlike the BCJR-MAP turbo equalizer, the MMSE turbo equalizer can be realized in adaptive form, thereby jointly performing equalization and tracking of the channel time variations.

In the second part, we have dealt with the application of the turbo principle to the domain of multi-user communications in code-division multiple access systems. We have presented a survey of conventional multi-user detection techniques. In particular, the PIC and SIC methods for cancellation of multi-user interference have been described. Their particular structures lead to a relatively simple exploitation of the turbo principle in a multi-user transmission context. Like for turbo equalization, different detectors can be implemented based on MMSE filters or matched-filter banks, for example.

In this chapter, we have deliberately limited ourselves to the presentation of two particular systems exploiting the turbo principle. However, more generally, any problem of detection or parameter estimation may benefit from the turbo principle. Thus, the range of solutions dealing with interference caused by a multi-antenna system at transmission and at reception (MIMO) has been enriched by iterative techniques such as the turbo BLAST (*Bell Labs layered space time*) [11.25]. The challenge involves proposing SISO detectors of reasonable complexity, without sacrificing data rates and/or the high performance of such systems.

We can also mention the efforts dedicated to receiver synchronization. Indeed, the gains in power provided by the turbo principle lead to moving the systems' operation point towards low signal to noise ratios. Now, conventional synchronization devices were not initially intended to operate in such difficult conditions [11.31]. One possible solution is to integrate the synchronization into the turbo process. A state of the art of turbo methods for timing synchronization was presented in [11.4]. More generally, when the choice of turbo processing at reception is performed, it seems interesting, or even necessary, to add a system to the receiver to iteratively estimate the transmission parameters, like channel turbo estimation or turbo synchronization.

Among other applications, the uplink of future radio-mobile communications systems will require higher and higher data rates, with an ever-increasing number of users. This is the one of the favourite applications of the turbo principle, the generalization of which will be essential in order to respond to the never-ending technological challenge posed by the evolution of telecommunications.

Understanding the turbo principle has led to the introduction of novel theoretical tools and concepts, like EXIT charts or factor graphs. While the former enable accurate prediction of the convergence threshold of iterative decoding schemes, the latter offer a graphical framework for representing complex detec-

tion/estimation problems and then deriving efficient turbo-like iterative algorithms for solving them. The interested reader will find good overviews of factor graphs and their applications in [11.37] and [11.38]. The use of factor graphs in a turbo equalization context has been considered in particular in [11.22] et[11.16] and an in-depth study of multi-user detection from a factor graph perspective has been presented in [11.10].

Bibliography

[11.1] Etsi digital cellular telecommunication system (phase 2+). GSM 05 Series, Rel. 1999.

[11.2] B. S. Ünal A. O. Berthet and R. Visoz. Iterative decoding of convolutionally encoded signals over multipath rayleigh fading channels. *IEEE Journal of Selected Areas in Communications*, 19(9):1729–1743, Sept. 2001.

[11.3] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv. Optimal decoding of linear codes for minimizing symbol error rate. *IEEE Transactions on Information Theory*, IT-20:284–287, March 1974.

[11.4] J. R. Barry, A. Kavcic, S. W. McLaughlin, A. Nayak, and W. Zeng. Iterative timing recovery. *IEEE Signal Processing Magazine*, 21(1):89–102, Jan. 2004.

[11.5] G. Bauch and V. Franz. A comparison of soft-in/soft-out algorithms for turbo detection. *Proceedings of International Conference on Telecommunications (ICT'98)*, pages 259–263, June 1998.

[11.6] G. Bauch and V. Franz. Iterative equalization and decoding for the gsm system. In *Proceedings of IEEE Vehicular Technology Conference (VTC'98)*, pages 2262–2266, Ottawa, Canada, May 1998.

[11.7] G. Bauch, H. Khorram, and J. Hagenauer. Iterative equalization and decoding in mobile communication systems. In *Proceedings of 2nd European Personal Mobile Commununications Conference (EPMCC'97)*, pages 307–312, Bonn, Germany, Sept.-Oct. 1997.

[11.8] R. Le Bidan. *Turbo Equalization for Bandwidth-Efficient Digital Communication over Frequency-Selective Channels.* PhD thesis, INSA de Rennes, Nov. 2003.

[11.9] R. Le Bidan, C. Laot, and D. Leroux. Real-time mmse turbo equalization on the tms320c5509 fixed-point dsp. In *Proceedings of IEEE International Conference on Accoustics, Speech and Signal Processing ICCASP 2004*, volume 5, pages 325–328, Montreal, Canada, May 2004.

[11.10] J. Boutros and G. Caire. Iterative multiuser joint decoding: Unified framework and asymptotic analysis. *IEEE Transactions on Information Theory*, 48(7):1772–1793, July 2002.

[11.11] J.-M. Brossier. *Signal et Communication Numérique – Equalisation et Synchronisation.* Collection Traitement du Signal. Hermès, Paris, 1997.

[11.12] C. Berrou C. Douillard, M. Jézequel, A. Picart, P. Didier, and A. Glavieux. Iterative correction of intersymbol interference: Turbo equalization. *European Transactions on Telecommununication*, 6(5):507–511, Sept.-Oct. 1995.

[11.13] P. R. Chevillat and E. Eleftheriou. Decoding of trellis-encoded signals in the presence of intersymbol interference and noise. *IEEE Transactions on Commununications*, 37(7):669–676, July 1989.

[11.14] A. Dejonghe and L. Vandendorpe. Turbo equalization for multilevel modulation: An efficient low-complexity scheme. In *Proceedings of IEEE International Conference on Communications (ICC'2002)*, volume 3, pages 1863–1867, New-York City, NY, 28 Apr.-2 May 2002.

[11.15] P. Didier. *La Turbo-Egalisation et son Application aux Communications Radiomobiles.* PhD thesis, Université de Bretagne Occidentale, Déc. 1996.

[11.16] R. J. Drost and A. C. Singer. Factor-graph algorithms for equalization. *IEEE Transactions on Signal Processing*, 55(5):2052–2065, May 2007.

[11.17] C. Fragouli, N. Al-Dhahir, S. N. Diggavi, and W. Turin. Prefiltered space-time m-bcjr equalizer for frequency-selective channels. *IEEE Transactions on Commununications*, 50(5):742–753, May 2002.

[11.18] V. Franz. *Turbo Detection for GSM Systems – Channel Estimation, Equalization and Decoding.* PhD thesis, Lehrstuhl für Nachrichten Technik, Nov. 2000.

[11.19] G. Ferrari G. Colavolpe and R. Raheli. Reduced-state bcjr-type algorithms. *IEEE Journal of Selected Areas in Communications*, 19(5):849–859, May 2001.

[11.20] A. Glavieux, C. Laot, and J. Labat. Turbo equalization over a frequency selective channel. In *Proceedings of International Symposium on Turbo Codes & Related Topics*, pages 96–102, Brets, France, Sept. 1997.

[11.21] G. H. Golub and C. F. Van Loan. *Matrix Computations.* The Johns Hopkins University Press, Baltimore, 3rd edition, 1996.

[11.22] Q. Guo, L. Ping, and H.-A. Loeliger. Turbo equalization based on factor graphs. In *Proceedings of IEEE International Symposium on Information Theory (ISIT'05)*, pages 2021–2025, Sept. 2005.

[11.23] J. Hagenauer. Soft-in / soft-out – the benefits of using soft decisions in all stages of digital receivers. In *Proceedings of 3rd International Symposium on DSP Techniques applied to Space Communications*, Noordwijk, The Netherlands, Sept. 1992.

[11.24] J. Hagenauer, E. Offer, C. Measson, and M. Mörz. Decoding and equalization with analog non-linear networks. *European Transactions on Telecommunications*, pages 107–128, Oct. 1999.

[11.25] S. Haykin, M. Sellathurai, Y. de Jong, and T. Willink. Turbo mimo for wireless communications. *IEEE Communications Magazine*, pages 48–53, Oct. 2004.

[11.26] M. Hélard, P.J. Bouvet, C. Langlais, Y.M. Morgan, and I. Siaud. On the performance of a turbo equalizer including blind equalizer over time and frequency selective channel. comparison with an ofdm system. In *Proceedings of International Symposium on Turbo Codes & Related Topics*, pages 419–422, Brest, France, Sept. 2003.

[11.27] G. D. Forney Jr. Maximum-likelihood sequence estimation of digital sequences in the presence of intersymbol interference. *IEEE Transactions on Information Theory*, IT-18(3):363–378, May 1972.

[11.28] W. Koch and A. Baier. Optimum and sub-optimum detection of coded data disturbed by time-varying intersymbol interference. In *Proceedings of IEEE Global Telecommununication Conference (GLOBECOM'90)*, volume 3, pages 1679–1684, San Diego, CA, 2-5 Dec. 1990.

[11.29] C. Langlais. *Etude et amélioration d'une technique de réception numérique itérative: Turbo Egalisation.* PhD thesis, INSA de Rennes, Nov. 2002.

[11.30] C. Langlais and M. Hélard. Mapping optimization for turbo equalization improved by iterative demapping. *Electronics Letters*, 38(2), Oct. 2002.

[11.31] C. Langlais, M. Hélard, and M. Lanoiselée. Synchronization in the carrier recovery of a satellite link using turbo codes with the help of tentative decisions. In *Proceedings of IEE Colloqium on Turbo Codes in Digital Broadcasting*, Nov. 1999.

[11.32] C. Laot. *Egalisation Auto-didacte et Turbo-Egalisation – Application aux Canaux Sélectifs en Fréquence.* PhD thesis, Université de Rennes I, Rennes, France, July 1997.

[11.33] C. Laot, R. Le Bidan, and D. Leroux. Low-complexity mmse turbo equalization: A possible solution for edge. *IEEE Transactions on Wireless Communications*, 4(3), May 2005.

[11.34] C. Laot, A. Glavieux, and J. Labat. Turbo equalization: Adaptive equalization and channel decoding jointly optimized. *IEEE Journal of Selected Areas in communications*, 19(9):1744–1752, Sept. 2001.

[11.35] I. Lee. The effects of a precoder on serially concatenated coding systems with an isi channel. *IEEE Transactions on Communications*, 49(7):1168–1175, July 2001.

[11.36] S.-J. Lee, N. R. Shanbhag, and A. C. Singer. Area-efficient high-throughput vlsi architecture for map-based turbo equalizer. In *Proceedings of IEEE Workshop on Signal Processing Systems SIPS 2003*, pages 87–92, Seoul, Korea, Aug. 2003.

[11.37] H.-A. Loeliger. An introduction to factor graphs. *IEEE Signal Processing Magazine*, 21(1):28–41, Jan. 2004.

[11.38] H.-A. Loeliger, J. Dauwels, J. Hu, S. Korl, L. Ping, and F. R. Kschischang. The factor graph approach to model-based signal processing. *Proceedings of IEEE*, 95(6):1295–1322, June 2007.

[11.39] K. Narayanan. Effects of precoding on the convergence of turbo equalization for partial response channels. *IEEE Journal of Selected Areas in Communications*, 19(4):686–698, Apr. 2001.

[11.40] N. Nefedov, M. Pukkila, R. Visoz, and A. O. Berthet. Iterative receiver concept for tdma packet data systems. *European Transactions on Telecommunications*, 14(5):457–469, Sept.-Oct. 2003.

[11.41] R. Otnes. *Improved receivers for digital High Frequency communications: iterative channel estimation, equalization, and decoding (adaptive turbo equalization)*. PhD thesis, Norwegian University of Science and Technology, 2002.

[11.42] B. Penther, D. Castelain, and H. Kubo. A modified turbo detector for long delay spread channels. In *Proceedings of International Symposium on Turbo Codes & Related Topics*, pages 295–298, Brest, France, Sept. 2000.

[11.43] A. Picart, P. Didier, and A. Glavieux. Turbo detection: A new approach to combat channel frequency selectivity. In *Proceedings of IEEE International Conference on Communications (ICC'97)*, pages 1498–1502, Montreal, Canada, June 1997.

[11.44] J. G. Proakis. *Digital Communications*. McGraw-Hill, New-York, 4th edition, 2000.

[11.45] S. U. H. Qureshi. Adaptive equalization. *Proceedings of the IEEE*, 73(9):1349–1387, Sept. 1985.

[11.46] M.C. Reed and P.D. Alexander. Iterative multiuser detection using antenna arrays and fec on multipath channels. *IEEE Journal of Selected Areas in communications*, 17(12):2082–2089, Dec. 1999.

[11.47] D. Reynolds and X. Wang. Low-complexity turbo equalization for diversity channels. *Signal Processing*, 81(5):989–995, May 2001.

[11.48] A. Tarable, G. Montorsi, and S. Benedetto. A linear front end for iterative soft interference cancellation and decoding in coded cdma. In *Proceedings of IEEE International Conference on Communications (ICC'01)*, 11-14 June 2001.

[11.49] S. ten Brink. Designing iterative decoding schemes with the extrinsic information transfer chart. *AEÜ International Journal of Electronics Communications*, 54(6):389–398, Nov. 2000.

[11.50] M. Tüchler, R. Kötter, and A. C. Singer. Turbo equalization: Principles and new results. *IEEE Transactions on Communications*, 50(5):754–767, May 2002.

[11.51] M. Tüchler, A. C. Singer, and R. Kötte. Minimum mean-squared error equalization using a priori information. *IEEE Transactions on Signal Processing*, 50(3):673–683, March 2002.

[11.52] G. Ungerboeck. Channel coding with mutilevel/phase signals. *IEEE Trans. Info. Theory.*, IT-28(1):55–67, Jan. 1982.

[11.53] M.K. Varanasi and T. Guess. Optimum decision feedback multiuser equalization with successive decoding achieves the total capacity of the gaussian multiple-access channel. In *Conference Record of the Thirty-First Asilomar Conference on Signals, Systems & Computers*, volume 2, pages 1405–1409, 2-5 Nov. 1997.

[11.54] G. M. Vitetta, B. D. Hart, A. Mämmelä, and D. P. Taylor. Equalization techniques for single-carrier unspread digital modulations. In A. F. Molish, editor, *Wideband Wireless Digital Communications*, pages 155–308. Prentice-Hall, Upper Saddle River, NJ, 2001.

[11.55] F. Volgelbruch and S. Haar. Improved soft isi cancellation for turbo equalization using full soft output channel decoder's information. In *Proceedings of IEEE Global Telecommununication Conference (Globecom'2003)*, pages 1736–1740, San Francisco, USA, Dec. 2003.

[11.56] X. Wang and H. V. Poor. Iterative (turbo) soft interference cancellation and decoding for coded cdma. *IEEE Transactions on Communications*, 47(7):1046–1061, July 1999.