Effect of Mismatched SNR on the Performance of Log-MAP Turbo Detector

Mohammad Ali Khalighi

***Abstract*—The effect of signal-to-noise ratio (SNR) mismatch on the bit error rate (BER) performance of Log-MAP turbo detector is studied. It is shown that the sensitivity to the mismatched SNR depends on the channel loss and the encoder memory length. In particular, the sensitivity is more important for a "difficult-to-equalize" channel or, in other words, a high-loss channel. In fact, by affecting the equalizer and the decoder, a mismatched SNR (with an offset with respect to the true SNR) affects the convergence of the turbo detector. Using the asymptotic analysis tool of extrinsic information transfer (EXIT) charts, the effect of a positive or a negative SNR offset on the convergence of the Log-MAP turbo detector is studied. When no information is available at the receiver on SNR, an online estimation of SNR is necessary in order to not lose the advantage of the Log-MAP turbo detector over other *suboptimal* turbo detectors. In this view, a relatively simple SNR estimation method is proposed, which offers satisfying results.**

***Index Terms*—Convergence analysis, extrinsic information transfer (EXIT) charts, turbo detection, iterative detection, log-maximum a posteriori (MAP) algorithm, signal-to-noise ratio (SNR) estimation, signal-to-noise ratio (SNR) mismatch, soft-output Viterbi algorithm (SOVA) .**

I. Introduction

After the invention of turbo codes in 1993 by Berrou *et al.* [1], which consists of applying the "turbo principle" to the decoding of parallel concatenated convolutional codes, the turbo principle was applied to the equalization and decoding in 1995 by Douillard *et al.* [2]. In turbo detection (or turbo equalization), the delay-dispersive channel that induces intersymbol interference (ISI) at receiver is regarded as an encoder serially concatenated to the channel encoder, usually a convolutional code. The same principle of turbo decoding is used at receiver to "decode" the channel by a soft-input–soft-output (SISO) equalizer and the channel code by a SISO decoder in an iterative manner. This technique is referred to as *turbo detection* [3].[1]

For turbo detection, the optimal SISO algorithm [in the sense of minimum bit error rate (BER)] to be used for equalization/decoding is the symbol-maximum *a posteriori* (MAP), also known as the BCJR algorithm [6].[2] It derives its advantage over other algorithms for greater encoder memory lengths and difficult-to-equalize channels, especially in low signal-to-noise ratio (SNR) [7], [8]. In practice, this algorithm is implemented in the logarithmic domain in order to reduce the numerical computation problems. The resulting algorithm is called Log-MAP.

Theoretically, the knowledge of SNR is necessary in the implementation of the Log-MAP turbo detector,[3] while turbo detectors based on some other SISO algorithms such as Max-Log-MAP and (a special implementation of) soft-output Viterbi algorithm (SOVA)[4] do not need this information (see Section III-C). The misknowledge of SNR may result in a degradation of the BER performance of the Log-MAP turbo detector and it can even lose its advantage over those "suboptimal" turbo detectors.

Recently, the problem of misestimation of SNR on the BER performance of turbo *decoders* is studied in several works [9]–[12]. It is shown that the Log-MAP turbo decoder is more sensitive to an underestimation of SNR than to an overestimation. On the other hand, the Max-Log-MAP turbo decoder is shown to be insensitive to an SNR mismatch. However, it seems that there is no previous work in the literature on the performance evaluation in the case of turbo detection.

This paper is organized as follows. After a brief introduction of turbo detection in Section II, we will formulate the Log-MAP equalization algorithm in Section III and will show where SNR is required in this algorithm. Also, it is explained why Max-Log-MAP and SOVA turbo detectors do not require SNR knowledge. Next, in Section IV, we present some simulation results studying the impact of an SNR mismatch on the BER performance of the Log-MAP turbo detector. In order to better understand the effect of SNR mismatch, in Section V we present an asymptotic analysis based on extrinsic information transfer (EXIT) charts. From these results, we will see the necessity of an online SNR estimation, for which we propose a relatively simple method in Section VI and will study the obtained performance. Conclusions are given in Section VII.

Binary phase-shift keying (BPSK) signaling is considered throughout the paper and it is assumed that the channel coefficients are known at receiver.

II. Principle of Turbo Detection

At the transmitter, information bits a_n are first encoded, usually using an recursive systematic convolutional (RSC) code. Before being transmitted through the channel, the encoded bits u_n are interleaved via an interleaver with size N.

Fig. 1 shows the block diagram of the turbo detector employed at the receiver, after the demodulation and sampling of the received signal (one sample per symbol duration). First, a SISO equalizer tries to remove the ISI; the *soft* outputs of this equalizer L^{eq}, which are in the form of log likelihood ratio (LLR; also called L-value) are then used to feed a SISO decoder, which decodes the RSC channel coding made at transmitter. Notice that we have used different indexes (n and k) to take into account the interleaving/deinterleaving. In the next iteration, profiting from the information at the decoder output L^{dec}, the equalizer tries to equalize the initial received symbols *better*. In this way, the equalizer, in fact, can profit from the redundancy added in the transmitted data by channel encoding. The procedure of equalization and decoding is repeated for several iterations. Each time, the *extrinsic* LLRs at the output of a SISO block (denoted by the subscript $\cdot_{\mathrm{ext}}$) are delivered to the other one, which uses them as *a priori* information. Passing extrinsic information between equalizer and decoder ensures that the *a priori* information used by them is decorrelated with the other information (observations) that they receive.

There would be no need to present the formulation of the MAP decoding, since it is given in several works in detail, especially in [6], [13].

Manuscript received October 22, 2002; revised May 6, 2003.

The author is with the Signal Processing (TSI) Department, Ecole National Supérieure de Télécommunications (ENST), Paris, France (e-mail: Khalighi@tsi.enst.fr).

Digital Object Identifier 10.1109/TVT.2003.816644

[1]Sometimes a distinction is made between *turbo detection* and *turbo equalization*; the latter is usually used when a simple equalizer is employed [4], [5].

[2]To be precise, this is true under the condition of white noise.

[3]When we speak of "Log-MAP turbo detector," for example, we mean that the Log-MAP algorithm is used for both equalization and decoding. The same is used for turbo detectors based on Max-Log-MAP and soft-output Viterbi algorithm (SOVA) algorithms.

[4]In this paper, by SOVA algorithm we mean a special implementation for SOVA, as explained in Section III-C.

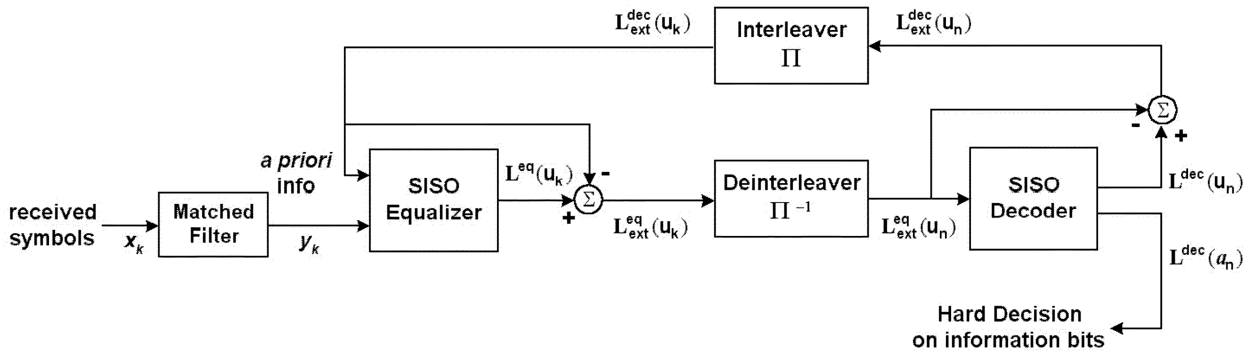

Fig. 1. Block diagram of the turbo detector.

It is, however, interesting to present the formulation of forward–backward Log-MAP equalizer that we provide in the following section. The differences with the case of MAP decoding are in metric calculation and in working in the logarithmic domain. We will see how SNR knowledge is required in the metric calculation of the Log-MAP equalizer.

We will also discuss the necessity of SNR knowledge in the case of Max-Log-MAP and SOVA turbo detectors.

III. SISO Equalization Algorithms and the Need for SNR Knowledge

A. Log-MAP Equalizer

The output of the symbol-by-symbol Log-MAP equalizer is defined as the *a posteriori* LLR on the transmitted symbols. Regarding the trellis of the channel, for a received symbol sequence (corrupted by noise and ISI) $\boldsymbol{X}_1^N$ (from x_1 to x_N, with N the length of the sequence), the LLR of the kth bit u_k, corresponding to the kth transmitted symbol d_k, is obtained by calculating the transition probabilities between all possible channel states between times k and $(k-1)$. For a BPSK signaling scheme with symbols $d_k \in \{+1, -1\}$, the LLR on $u_k \in \{0, 1\}$ is given by

$$\begin{aligned} L^{\text{eq}}(u_k) &= \text{LLR}\left(u_k \,\middle|\, \boldsymbol{X}_1^N\right) \\ &= \log \frac{p\left(u_k = 1 \,\middle|\boldsymbol{X}_1^N\right)}{p\left(u_k = 0 \,\middle|\, \boldsymbol{X}_1^N\right)} \\ &= \log \frac{\sum_{\substack{(m',m)\\ d_k=+1}} p\left(S_{k-1} = m', S_k = m, \boldsymbol{X}_1^N\right)}{\sum_{\substack{(m',m)\\ d_k=-1}} p\left(S_{k-1} = m', S_k = m, \boldsymbol{X}_1^N\right)} \end{aligned} \tag{1}$$

where S_k denotes the channel state at time k. Each term in the numerator and denominator indicates the probability of the transition from the state m' at time $(k-1)$ to the state m at time k.

In computing the sum of exponential of LLRs a *Jacobian logarithm* function is usually considered, which is defined as [14]

$$\log\left(e^{L_1} + e^{L_2}\right) = \text{Jac.log}(L_1, L_2) \tag{2}$$

$$\begin{aligned} \text{Jac.log}(L_1, L_2) &\triangleq \max(L_1, L_2) + \log\left(1 + e^{(-|L_1 - L_2|)}\right) \\ &= \max(L_1, L_2) + f_c(|L_2 - L_1|). \end{aligned} \tag{3}$$

$|\cdot|$ is the absolute value operator. In practice, the correction function $f_c(\cdot)$ in (3) is implemented using a precomputed look-up table [15]. This implementation of the Log-MAP algorithm is sometimes called Max*-Log-MAP algorithm [10].

In this paper, we consider the Log-MAP algorithm, i.e., using the exact calculation of the correction function in (3). So we write (1) in the form

$$\begin{aligned} L^{\text{eq}}(u_k) = &\underset{\substack{(m',m)\\ d_k=+1}}{\text{Jac.log}} \left[\log\ p\Big(S_{k-1} = m', S_k = m, \boldsymbol{X}_1^N\Big)\right] \\ &- \underset{\substack{(m',m)\\ d_k=-1}}{\text{Jac.log}} \left[\log\ p\Big(S_{k-1} = m', S_k = m, \boldsymbol{X}_1^N\Big)\right]. \end{aligned} \tag{4}$$

Each term in (4) is calculated using the forward–backward algorithm as detailed in the following:

$$\begin{aligned} \log p\left(S_{k-1} = m', S_k = m, \boldsymbol{X}_1^N\right) &= \breve{\alpha}_{k-1}(m') \\ &+ \breve{\gamma}_k(m', m) + \breve{\beta}_k(m) \end{aligned} \tag{5}$$

$$\breve{\alpha}_k(m) = \underset{m'}{\text{Jac.log}} \left[\breve{\alpha}_{k-1}(m'), \breve{\gamma}_k(m', m)\right] \tag{6}$$

$$\breve{\beta}_{k-1}(m') = \underset{m}{\text{Jac.log}} \left[\breve{\beta}_k(m), \breve{\gamma}_k(m', m)\right] \tag{7}$$

with the initialization

$$\breve{\alpha}_0(0) = 0, \quad \text{and} \quad \breve{\alpha}_0(m) = -\infty \quad \text{for } m \neq 0 \tag{8}$$

$$\breve{\beta}_N(0) = 0, \quad \text{and} \quad \breve{\beta}_N(m) = -\infty \quad \text{for } m \neq 0. \tag{9}$$

The term $\breve{\gamma}_k(m', m)$ corresponding to the transition probability between two states is computed as

$$\breve{\gamma}_k(m, m') = \breve{\gamma}_k^{\text{ch}}(m, m') + d_k L_{\text{ext}}^{\text{dec}}(u_k) \tag{10}$$

$L_{\text{ext}}^{\text{dec}}(u_k)$ is the *a priori* LLR on a_n, which is in fact the extrinsic information calculated at the decoder output at the previous iteration. Notice that we assume that the information source “generates” u_k with equal probability and that there is no *intrinsic* information on the transmitted bits. $\breve{\gamma}_k^{\text{ch}}(m, m')$ is given by[5] [16]

$$\breve{\gamma}_k^{\text{ch}}(m, m') = \frac{1}{\sigma_n^2} \Re e \left[d_k^* \left(2y_k - g[0]d_k - 2\sum_{i=1}^{L} g[i] d_{k-i} \right) \right] \tag{11}$$

σ_n^2 is the complex noise variance and $(\cdot)^*$ and $\Re e[\cdot]$ denote the complex conjugate and the real-part operators, respectively. Also, $g[n] = h[n] * h^*[-n]$ with $h[n]$ the discrete channel impulse response of length $(L+1)$ and $*$ the convolution operator. y_k is the output of the matched filter.

Remember that we assumed that the channel is perfectly known at receiver. Therefore, $g[n]$ is known exactly. So, in (11), we need σ_n^2, the

[5]The exact expression includes some additive terms, but they can be neglected since regarding (3), which is used in (6) and (7), these additive terms will finally been cancelled out in (4).

variance of noise. In fact, when we speak of SNR knowledge, we mean the knowledge of the noise variance σ_n^2 with the assumption of perfect channel knowledge at receiver.

From (10) and (11), SNR is required in the calculation of the branch metrics and, more precisely, in $\breve{\gamma}_k^{\mathrm{ch}}(m, m')$. If SNR is overtaken, the weight that will be given to the branch metrics is incorrectly greater than the value that deserves. On the other hand, undertaking the SNR has the effect that the *a priori* information on symbols are considered in too strong a manner. However, an SNR mismatch affects also the *a priori* information input to a SISO block, which is in fact the extrinsic information at the output of the other SISO block in the previous iteration. For instance, the LLRs at the output of the Log-MAP equalizer are used by the decoder and, hence, any error in these LLRs affects the decoding procedure too. This, in turn, affects the *a priori* information that the equalizer will use in the next iteration. Discussions on the effect of the mismatched SNR on the overall performance will be later made in Section V.

B. Max-Log-MAP Based Turbo Detector

If the correction term (the right-hand-side term) in (3) is neglected for simplicity, the resulting algorithm is called the Max-Log-MAP algorithm. In other words, we use the following approximation:

$$\log\left(e^{L_1} + e^{L_2}\right) \approx \max(L_1, L_2). \tag{12}$$

Using (12), (4), (6), and (7) are simplified to (13), (14), and (15), respectively

$$\begin{aligned} L^{\mathrm{eq}}(u_k) = & \max_{\substack{(m',m) \\ d_k=+1}} \left[\log p\left(S_{k-1}=m', S_k=m, \boldsymbol{X}_1^N\right)\right] \\ & - \max_{\substack{(m',m) \\ d_k=-1}} \left[\log p\left(S_{k-1}=m', S_k=m, \boldsymbol{X}_1^N\right)\right] \end{aligned} \tag{13}$$

$$\breve{\alpha}_k(m) = \max_{m'} \left[\breve{\alpha}_{k-1}(m'), \breve{\gamma}_k(m', m)\right] \tag{14}$$

$$\breve{\beta}_{k-1}(m') = \max_{m} \left[\breve{\beta}_k(m), \breve{\gamma}_k(m', m)\right]. \tag{15}$$

Other formulations in Section III-A remain unchanged.

1) Need-to-SNR Knowledge: As explained in the following, the Max-Log-MAP turbo detector does not need the SNR information.

Let us consider an arbitrary (inexact) SNR at receiver. If, at the first iteration, there is no intrinsic information on u_k, which is usually the case, from (10) and (11), $\breve{\gamma}_k(m, m')$ is proportional to the assumed SNR. It is also the case for $\breve{\alpha}_k(m)$ and $\breve{\beta}_k(m)$ from (14) and (15) for every k and m. Hence, in (5), $\log p(S_{k-1} = m', S_k = m, \boldsymbol{X}_1^N)$ is also proportional to SNR. As a result, the SNR can be factored out in (13).

In this way, the calculated LLRs at the output of the SISO equalizer are proportional to SNR. If the SISO decoding algorithm is also Max-Log-MAP, the LLRs at its output and, hence, the extrinsic LLRs $L_{\mathrm{ext}}^{\mathrm{dec}}$ will be proportional to SNR too. So the factor SNR can be factored out in all these operations and in all iterations and, since the final hard decisions on the information bits depend only on the sign of the LLRs at the decoder output, the whole turbo-detector algorithm performance does not depend on the assumed SNR.

C. SOVA-Based Turbo Detector

We briefly describe the implementation of SOVA that we consider in this paper. As will be seen, we consider some approximations that permit the algorithm to be insensitive to the SNR knowledge, while its performance remains almost unchanged.

First consider the implementation of the SOVA *decoder* in the trace-back mode, as described in detail in [13] and [17]. We use the same notations as in [13] and avoid explaining unnecessary details.

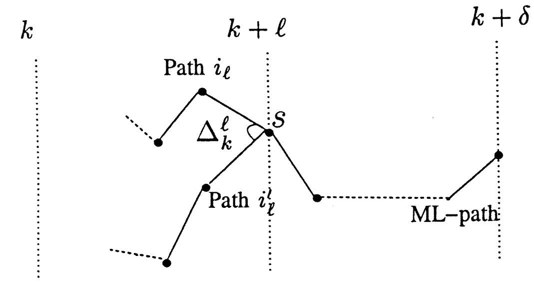

Fig. 2. SOVA decoding: selection of path i_ℓ associated with the state s, which is along the ML path.

The decision $\hat{u}_k$ on an information bit u_k, as well as its LLR $L(\hat{u}_k)$, are obtained after a delay of δ.[6] For each state s of the code trellis, two paths of the trellis merging to this state are considered. The cumulated metric of each path is calculated by adding the metric of the corresponding branch with the (previous) cumulated metric of the path. At time j, we calculate the cumulated metrics $M_j(s^{(i)})$ of each path i in the trellis of the code, ending at each state s.

Assume that at time $k+\ell$, with $0 \le \ell \le \delta$, the path i_ℓ with the larger cumulated metric $M_{k+\ell}(s^{(i_\ell)})$ is selected as the path corresponding to the state s (see Fig. 2). Let Δ_k^ℓ be the difference between the metrics of two paths, i_ℓ and i'_ℓ, ending to the state s

$$\Delta_k^\ell = M_{k+\ell}\left(s^{(i_\ell)}\right) - M_{k+\ell}\left(s^{(i'_\ell)}\right) \ge 0. \tag{16}$$

The likelihood ratio (LR) of this path (i_ℓ) selection is equal to Δ_k^ℓ [13].

Given the maximum likelihood (ML) path (between time k and $k+\delta$), we decide on the bit $\hat{u}_k$ (at the beginning of the window). The corresponding soft value $L(\hat{u}_k)$ is obtained as explained in the following. Considering each path selection along the ML path (with the relative time reference ℓ), we trace the discarded path back to the beginning of the window. If the corresponding decision on u_k, $\breve{u}_k^\ell$ equals $\hat{u}_k$, the reliability of this path selection R_ℓ is infinity. Otherwise, $R_\ell = \Delta_k^\ell$. In this way, we calculate $L(\hat{u}_k)$ as below [13], [17]

$$L(\hat{u}_k) = \hat{u}_k \cdot 2 \operatorname{arctanh}\left(\prod_{\ell=0}^{\delta} \tanh\left(\frac{R_\ell}{2}\right)\right). \tag{17}$$

Rather than (17), we use the following approximation:

$$L(\hat{u}_k) \approx \hat{u}_k \cdot \min_{\ell=0,\ldots,\delta} R_\ell. \tag{18}$$

The implementation of the SOVA equalizer is similar to that of the SOVA decoder explained above, especially for the case of BPSK modulation that we consider in this paper. The branch metric calculation is performed according to (10). The implementation is a little different from the decoder in the case of an RSC code, as detailed in [13] and [17], since the FIR channel is nonrecursive and the incident branches to each state of the channel trellis correspond to the same input.

For modulations with the number of constellation points greater than two, there are more than two incident branches to each state of the trellis. It can be shown that for the number 2^b of constellation points, at time $k+\ell$, the LR of path selection i_ℓ ending to the state s is given by [18]

$$\text{LR of selection of } i_\ell \approx -\underset{\substack{q=1,\ldots,2^b \\ q\neq p}}{\text{Jac.log}} \left(-\Delta_{k,q}^\ell\right) \tag{19}$$

[6] In other words, a window of length δ between time references k and $k+\delta$ is considered. This window is next shifted by one time sample for the decision on the next bit u_{k+1} and so on.

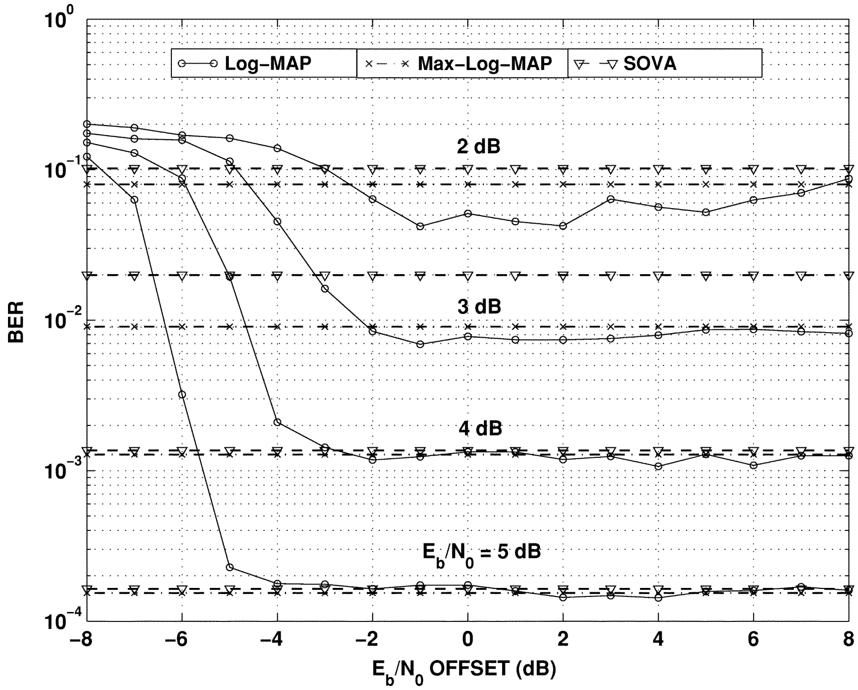

Fig. 3. BER sensitivity of Log-MAP turbo detector to SNR offset; B channel, $M = 2$, fifth iteration, $N = 4096$.

with

$$\Delta_{k,q}^{\ell} = M_{k+\ell}\left(s^{(i_\ell)}\right) - M_{k+\ell}\left(s^{(i'_{\ell,q})}\right); \quad q = 1, \ldots, 2^b, \quad q \neq p. \tag{20}$$

p is the index to the branch corresponding to the path i_ℓ and q is the index to the other branches. Here, instead of (19), we use the same approximation as we used for Max-Log-MAP algorithm, as follows:

$$\text{LR of selection of } i_\ell \approx \min_{\substack{q=1,\ldots,2^b \\ q \neq p}} \left(\Delta_{k,q}^{\ell}\right); \quad q = 1, \ldots, 2^b, \quad q \neq p. \tag{21}$$

It has been verified that the approximations considered here have a very negligible effect on the resulting performance of the SOVA turbo detector. In the following, we explain the interest of these approximations.

1) Need-to-SNR Knowledge: Similar to the case of the Max-Log-MAP turbo detector, the SOVA turbo detector that we present above does not require SNR knowledge.

Consider the SOVA equalization algorithm. At any instant and for each channel state s, the cumulated metric of each path merging to s is calculated by adding the metric of the corresponding branch with the (previous) cumulated metric of the path. Since the branch metric is calculated according to (11), the cumulated metrics of all paths at any instant are proportional to SNR, provided that there is no intrinsic information on transmitted bits. From (16) or (20), Δ_k^{ℓ} or $\Delta_{k,q}^{\ell}$ (for $b > 1$) are also proportional to SNR. By using the approximation of (18) in the calculation of the L-values of each decision $\hat{u}_k$, as well as the approximation of (21) in the calculation of LR of the path selections (for $b > 1$), the obtained LLRs $L(\hat{u}_k)$ will be proportional to SNR. If the decoder is also SOVA with the approximation of (18), the final hard decisions on information bits will not depend on the assumed SNR.

IV. Sensitivity of Turbo Detector to SNR

The important question is how much the Log-MAP turbo detector is sensitive to an SNR mismatch.The sensitivity of Log-MAP turbo *decoder* to SNR mismatch is studied in [9]–[12]. It is shown that the turbo decoder based on Log-MAP is more tolerant of an overestimation of SNR that an underestimation. We expect almost the same behavior in the case of turbo detection, since the SISO equalization and decoding based on Log-MAP are very similar.

We use the notation of SNR for E_b/N_0, with E_b the average received energy per informative bit and N_0 the unilateral noise power spectral density. So, considering $d_k = \pm 1$ and RSC codes with rate $R = (1/2)$, we have $E_b = (T_s/R)$ and $\sigma_n^2 = (N_0/T_s)$, with T_s the symbol duration.

We will consider two channels, B and C, with discrete-time impulse responses of $\{0.407, 0.815, 0.407\}$ and $\{0.227, 0.460, 0.688, 0.460, 0.227\}$, respectively [16]. To quantify the difficulty of channel equalization, we will use the criterion of *channel loss*. Channel loss is defined as the ratio of the square of the minimum distance of an additive-white Gaussian-noise (AWGN) channel to the square of the minimum distance of the channel under consideration [16]. For B and C channels, the loss is about 1.757 dB and 5.007 dB, respectively [19].[7]

Simulation results show that if no channel coding is performed, a Log-MAP equalizer has a low sensitivity to an SNR mismatch and that the sensitivity is almost the same for B and C channels [7]. This is also the case for Log-MAP decoding of convolutional codes when a flat channel is considered [7]. In fact, in pure equalization or pure decoding cases, only the sign of LLRs at the output of Log-MAP equalizer/decoder is used, so low sensitivity to SNR mismatch is observed. However, as we will see in the following, this is not the case when the L-values are used in a turbo-detection scheme. In this case, different

[7]It is useful to remember that the interest of the definition of the channel loss (in decibels) is that, in high enough SNR, the error probability after a Viterbi equalization is equal to the error probability of an AWGN channel (ISI free), with SNR′ (in decibels) equal to [SNR − channel loss (in decibels)] [16].

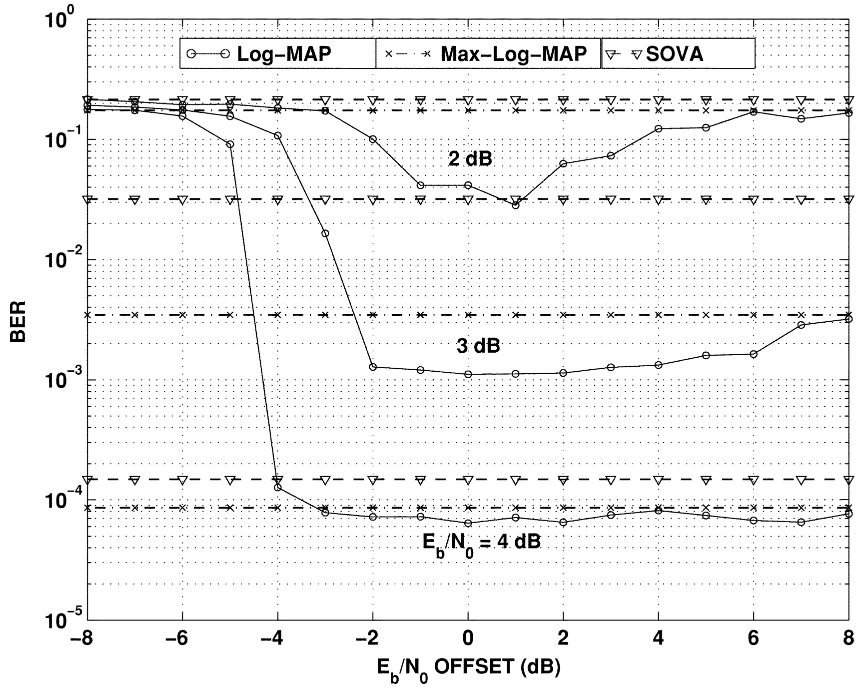

Fig. 4. BER sensitivity of Log-MAP turbo detector to SNR offset; B channel, $M = 5$, fifth iteration, $N = 4096$.

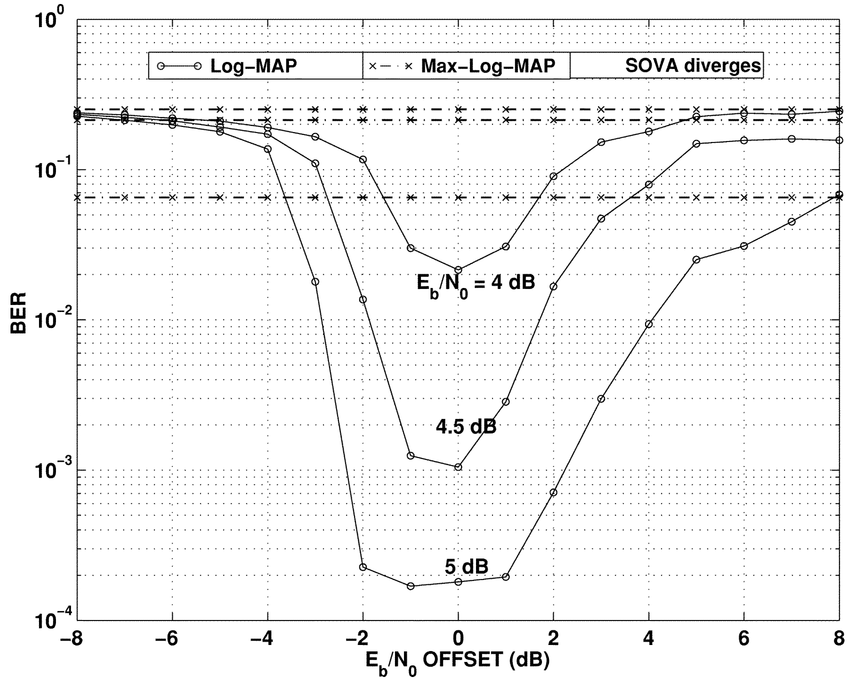

Fig. 5. BER sensitivity of Log-MAP turbo detector to SNR offset; C channel, $M = 2$, eighth iteration, $N = 4096$. SOVA diverges for SNR ≤ 5 dB.

behaviors are observed for different channels and different codes, regarding the sensitivity to SNR mismatch.

Let M be the channel-encoder memory length. Curves of BER (after five iterations) versus SNR offset are given in Figs. 3 and 4, respectively, for the case of B channel with $M = 2$ and B channel with $M = 5$. Also, curves of BER (after eight iterations) versus SNR offset are given in Fig. 5 for C channel with $M = 2$. $M = 2$ and $M = 5$ correspond to the encoder generator matrices of $[1, 7/5]$ and $[1, 75/53]$, respectively. Random interleaving with size $N = 64 \times 64$ is used. In fact, for this interleaver size, the required number of iterations resulting in the complete convergence of the turbo detector (i.e., to approach the BER of the AWGN channel) is about five for the case of B channel with $M = 2, 5$ and eight for the case of C channel with $M = 2$.

Notice that the point offset $= 0$ corresponds to the case of exact SNR knowledge at receiver. In order to compare the BER performance to that of Max-Log-MAP and SOVA turbo detectors, the corresponding curves are also provided for these cases. Remember that Max-Log-MAP and SOVA turbo detectors do not need SNR information; so the corresponding BERs do not change with SNR offset.

From these simulation results, it can be concluded that the sensitivity to SNR is more important for a channel with a greater channel loss. On the other hand, this sensitivity is more considerable for a greater encoder memory length M. Similar to the case of turbo decoding, a positive offset in SNR is more tolerable than a negative offset. Discussion on these observations is left for the next section.

V. Sensitivity Analysis Using EXIT Charts

Understanding the presented results by an analytic approach is difficult because of the nonlinear iterative process of turbo detection. In fact, an SNR mismatch, as will be shown in this section, affects the convergence behavior of the turbo detector. A simple and efficient way to analyze convergence is to make use of EXIT charts, which is, in fact, an asymptotic analysis tool. Use of EXIT charts in convergence analysis was first proposed by ten Brink [20] for the case of turbo decoding is and applied to turbo detection and turbo equalization in [5].

Before studying the effect of mismatched SNR on the convergence of the Log-MAP turbo detector, we first present a brief recall on the principles of this method. The interested reader can refer to the cited references for detailed explanations.

A. EXIT Charts

Use of EXIT charts makes it possible to visualize the exchange of the extrinsic information between the two constituent SISO blocks (equalizer and decoder for the case of turbo detection). The obtained detection trajectory gives insight to the convergence of the iterative process.

Let ξ_A be the *a priori* information at the input of the SISO equalizer. For large interleaver sizes, this *a priori* information can be considered to be uncorrelated from the information obtained from channel. On the other hand, the distribution of the extrinsic information at the output of a SISO block approaches the Gaussian distribution with increasing the number of iterations [20].

These observations suggest that the *a priori* input ξ_A to the SISO block be modeled by

$$\xi_A = \mu_{\xi_A} \cdot x + n_{\xi_A} \tag{22}$$

where $\mu_{\xi_A} = |\mathrm{E}\{\xi_A\}|$ with $|\cdot|$ the absolute value and $\mathrm{E}\{\cdot\}$ the expected value. For simplicity, we have used the notation x (and the corresponding random variable X) for symbols rather than d_k. Also, n_{ξ_A} is a zero-mean Gaussian-random variable with variance $\sigma^2_{\xi_A}$. Moreover, we consider the following relationship between μ_{ξ_A} and σ_{ξ_A}:

$$\mu_{\xi_A} = \frac{\sigma^2_{\xi_A}}{2}. \tag{23}$$

These assumptions are motivated by the fact that, for the case of an AWGN channel with the noise variance σ_n^2, the LLR on the received noisy symbol $x+n$ has a Gaussian distribution with the mean and variance equal to $2x/\sigma_n^2$ and $4/\sigma_n^2$, respectively. The expression of (23) can also be derived by the asymptotic consistency condition for L-value distribution f [20], [21] as

$$f(\xi_A \mid x) = f(-\xi_A \mid x)\exp(\xi_A). \tag{24}$$

To qualify the extrinsic information at the input or output of a SISO block, we use the mutual information $I(X;\xi)$ between the transmitted symbols X and the L-values ξ

$$I(X;\xi) = \frac{1}{2}\cdot \sum_{x=-1,+1}\int_{-\infty}^{\infty} f(\xi \mid X=x) \times \log_2 \frac{2f(\xi \mid X=x)}{f(\xi \mid X=-1)+f(\xi \mid X=+1)}\, d\xi. \tag{25}$$

We have $0 \le I(X;\xi) \le 1$. Considering the assumption of Gaussian distribution for L-values, (25) simplifies to the following expression:

$$I(X;\xi) = \int_{-\infty}^{\infty} f(\xi \mid X=+1)\log_2 \frac{2}{1+\exp(-\xi)}\, d\xi. \tag{26}$$

Let us use the subscripts $\cdot_A$ and $\cdot_E$ to denote the variables at the input and output of a SISO block, respectively. Also, for simplicity, we will denote $I(X;\xi_A)$ and $I(X;\xi_E)$ briefly by I_A and I_E, respectively. We note that I_A is only a (strictly increasing) function of σ_{ξ_A}. It equals 0 and 1, respectively, for no $(\sigma_{\xi_A}=0)$ and perfect $(\sigma_{\xi_A}\to\infty)$ *a priori* information. The EXIT chart of the equalizer or the decoder is considered as the transfer function mapping the input (*a priori*) information $I_A \in [0,1]$ to the output (extrinsic) information $I_E \in [0,1]$. This characteristic curve is obtained separately for the SISO equalizer and decoder.

To obtain the EXIT chart, for each given I_A, corresponding to a special σ_{ξ_A} from (26), we generate at the input of the SISO module, Gaussian distributed *a priori* L-values ξ_A and calculate the mutual information I_E at the output of the module from (25) by a histogram estimation (no Gaussian assumption is made on the ξ_E distribution).

The behavior of the iterative detector is determined by associating $I_E^{\mathrm{EQ}} \Rightarrow I_A^{\mathrm{DEC}}$ and the inverse: $I_E^{\mathrm{DEC}} \Rightarrow I_A^{\mathrm{EQ}}$, where the superscripts $\cdot^{\mathrm{EQ}}$ and $\cdot^{\mathrm{DEC}}$ refer to the equalizer and the decoder, respectively.

The iterative process is characterized by a trace between the EXIT charts of the equalizer and the decoder. It is important to note that the obtained trajectory provides an asymptotic description to the iterative detection. In other words, it corresponds to the case of perfect interleaving with infinite interleaver size. Hence, the results to be presented do not correspond exactly to those of Section IV, presented for an interleaver size of 4096, especially regarding the required number of iterations for the convergence of the turbo detector for a given SNR. Nevertheless, the EXIT chart approach helps us understand the effect of mismatched SNR on the convergence of the turbo detector.

B. Impact of SNR Mismatch

As in Section V-A, to model the *a priori* L-values ξ'_A feeding a SISO module for the case of mismatched SNR, we consider the case of an AWGN channel. Let the noisy received signal be $z = x + n$, where n is the zero-mean additive noise of variance σ_n^2. If the noise variance is considered to be ${\sigma'}_n^2$ (with a mismatch with respect to σ_n^2), the L-value ξ'_A on z is calculated as

$$\xi'_A = \frac{2}{{\sigma'}_n^2}(x+n) = \mu_{\xi'_A}\cdot x + n_{\xi'_A} \tag{27}$$

where $\mu_{\xi'_A} = 2/{\sigma'}_n^2$ and the conditional variance of ξ'_A equals $\sigma^2_{\xi'_A} = 4\sigma_n^2/{\sigma'}_n^4$. Let us define the factor $\theta = \sigma_n^2/{\sigma'}_n^2 = \mathrm{SNR}'/\mathrm{SNR}$, where SNR' is the mismatched SNR. We have $\xi'_A = \theta\,\xi_A$ and consequently

$$\mu_{\xi'_A} = \theta\mu_{\xi_A}, \quad \sigma^2_{\xi'_A} = \theta^2\sigma^2_{\xi_A} \tag{28}$$

where ξ_A is the L-value and $\mu_{\xi_A}\cdot x$ and $\sigma^2_{\xi_A}$ are its mean and variance for perfect knowledge of σ_n^2 (i.e., for $\theta = 1$). As a result, for the case of mismatched SNR, we generate the *a priori* L-values ξ'_A according to the distribution[8] $\mathcal{N}\left(\mu_{\xi'_A}\cdot x = \theta\,\mu_{\xi_A}\cdot x, \sigma_{\xi'_A} = \theta\sigma_{\xi_A}\right)$.[9]

It can be verified from (25) or (26) that the multiplication of the L-values by a factor θ does not change the corresponding mutual information I. As a result, for the case of Max-Log-MAP or SOVA equalizer/decoder, an SNR offset does not change the EXIT characteristic

[8] $\mathcal{N}(\mu,\sigma)$ denotes the Gaussian distribution with mean μ and standard deviation σ.

[9] Notice that, in contrast to (23), here we have $\mu_{\xi'_A} = \sigma^2_{\xi'_A}/2\theta$.

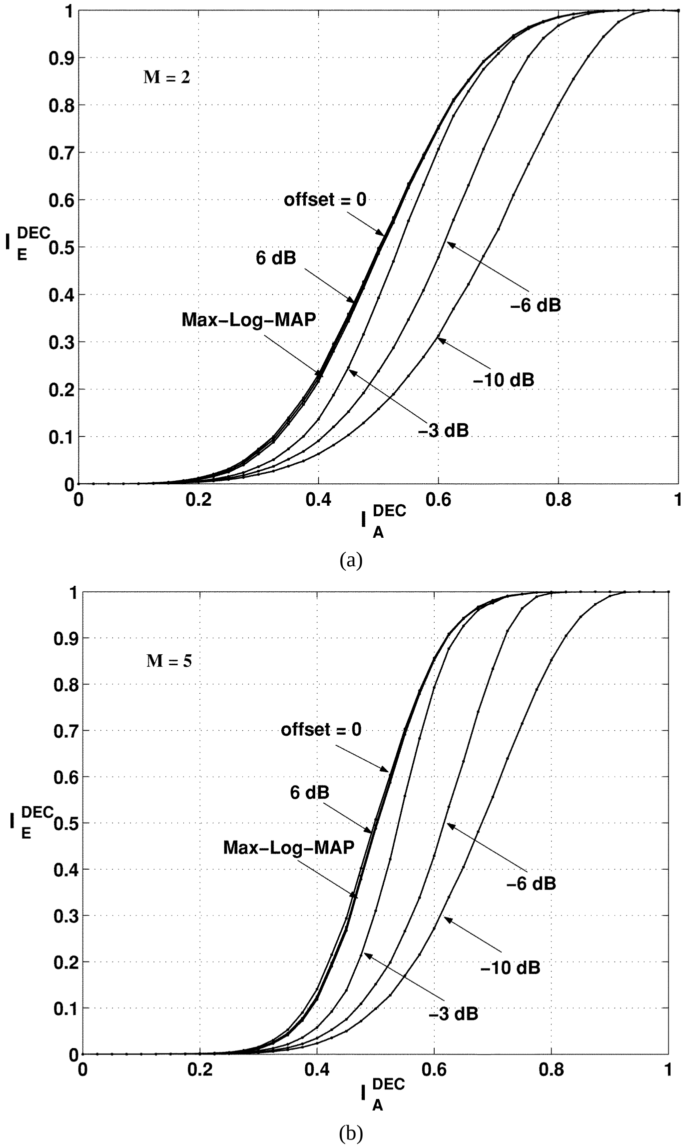

Fig. 6. Log-MAP decoder EXIT chart.

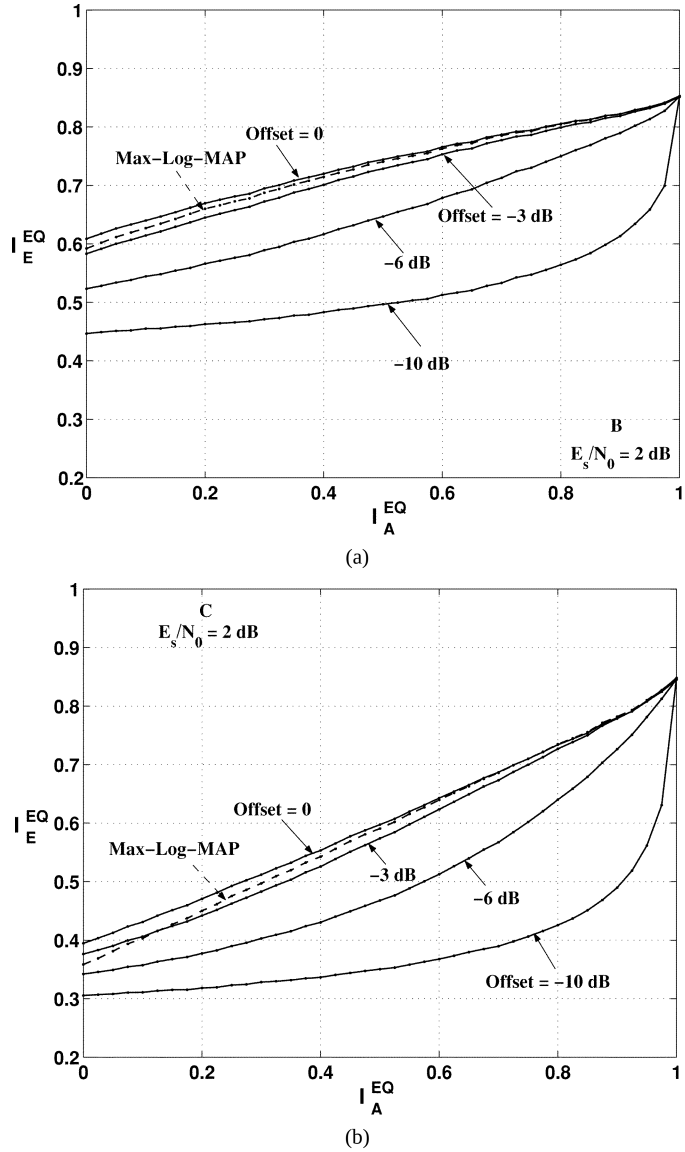

Fig. 7. Log-MAP equalizer EXIT chart, $E_s/N_0 = 2$ dB. (a) B channel and (b) C channel.

curve of the SISO block (see the explanations of Sections III-B and III-C).

For the case of forward–backward Log-MAP algorithm, the variables $\breve{\gamma}_k$ are multiplied by the factor θ. For the case of Log-MAP equalizer, this can be seen from (11), where $\breve{\gamma}_k^{\mathrm{ch}}$ is proportional to θ [see (11)] and also $L_{\mathrm{ext}}^{\mathrm{dec}}$, since it is generated according to $\mathcal{N}(\theta\mu_{\xi_A} \cdot x, \theta\sigma_{\xi_A})$ distribution. For the case of Log-MAP decoder we have $\breve{\gamma}_k(m, m') = d_k L_{\mathrm{ext}}^{\mathrm{eq}}(u_k)$, which is again proportional to θ due to the corresponding distribution.

If we consider the calculation of $\breve{\alpha}_k(m)$ from (6), for example, by a positive offset $(\theta > 1)$, the correction term of Jac.log in (3) becomes less effective with respect to the "max" term, as compared to the case of no SNR offset. At the limit of $\theta \rightarrow \infty$ (large positive SNR offset), this correction term becomes equal to zero and the performance approaches that of max-Log-MAP algorithm. For a negative offset $(\theta < 1)$, however, the correction term in (3) becomes (incorrectly) more important with respect to the "max" term.[10] This falsifies the calculation of $\breve{\alpha}_k(m)$ and $\breve{\beta}_k(m)$ from (6) and (7).

[10]At the limit of $\theta \rightarrow 0$ (large negative SNR offset), the correction term becomes equal to $\log 2$.

C. EXIT Chart Simulation Results

Before studying the effect of mismatched SNR on the EXIT characteristic curves of the turbo detector, let us consider this effect separately on the EXIT charts of the decoder and the equalizer. For the results presented in this paper, for each I_A (corresponding to a given σ_{ξ_A} or $\sigma_{\xi'_A}$), we have randomly generated at least 10^6 *a priori* ξ_A (or ξ'_A) according to the presented distribution and also 10^6 corresponding coded symbols (for the case of the decoder) or noisy received symbols (for the case of the equalizer).

Fig. 6 shows the effect of an SNR offset on the EXIT chart of the Log-MAP decoder for the cases of $M = 2$ and $M = 5$, considered previously in Section IV. With an SNR offset, for a given I_A^{DEC} the L-values at the decoder output are of "worse quality" (since they are not calculated correctly) and, hence, I_E^{DEC} is smaller, as compared to the case of no offset. This difference is more important for a negative offset than for a positive offset, as explained in the previous subsection. For a positive offset, the transfer curve lays between those of Log-MAP with zero-offset and max-Log-MAP decoder.

We have shown in Fig. 7 the effect of mismatched SNR on the EXIT chart of the Log-MAP equalizer, where two channels, B and C, are considered with $E_s/N_0 = 2$ dB. E_s is the average received energy per symbol. For the case of encoded data bits with the code rate of R, we have $E_s = RE_b$. For $R = (1/2)$ considered in Section IV, $E_s = 2$ dB corresponds to $E_b = 5$ dB.

Notice that, for $I_A^{\mathrm{EQ}} = 1$, where we have perfect *a priori* information on the transmitted symbols, the performance becomes equal to that of a Gaussian (ISI-free) channel. It is normal that all curves for different offset values converge to the same point for $I_A^{\mathrm{EQ}} = 1$.

Although the effect of an SNR mismatch on the equalizer EXIT chart is almost the same for B and C channels, as it will be seen, the effect incorporated on the turbo-detector convergence is different. In fact, we will see that since the slope of the $I_A^{\mathrm{EQ}}/I_E^{\mathrm{EQ}}$ curve is greater for C channel (see Fig. 7 for offset $= 0$), the convergence of the turbo detector is more sensitive to an SNR offset in this case.

By superimposing the EXIT charts of the decoder and the equalizer, we have traced the trajectory characterizing the iterative detection process in Fig. 8 for $M = 2$ and two cases of B and C channels with $E_b/N_0 = 5$ dB (corresponding to $E_s/N_0 = 2$ dB in Fig. 7). As expected, the convergence for the case of B channel takes place faster, i.e., with a smaller number of iterations [22]. About three iterations are required for the case of B channel, as compared to six for the case of C channel to attain the intersection point of the EXIT charts, close to $I_E^{\mathrm{DEC}} = 1$.[11]

In general, the number of iterations required for the complete convergence (attaining the intersection point of two EXIT charts) is greater when we have an offset on SNR. Also, the (final) I_A^{DEC} corresponding to this intersection point is different for different offset values. In other words, even after the complete convergence, the final BER differs (is higher) for a mismatched SNR.

Let us consider the case of C channel with $E_b/N_0 = 5$ dB. To compare the case of Log-MAP turbo detector with max-Log-MAP, we have shown the corresponding trajectory in Fig. 9.

For the Log-MAP turbo detector, by a positive SNR offset, the convergence behavior approaches that of the max-Log-MAP, shown in Fig. 9. In fact, regarding the explanations provided in Section V.B, these EXIT charts and the related trajectory can be considered as to show the performance limit for the Log-MAP turbo detector with a large positive offset.

The effect of a negative offset of -3 and -6 dB on the convergence of the Log-MAP turbo detector is shown in Fig. 10. It is seen that a relatively small offset of -3 dB makes the convergence of the turbo detector considerably slower. In other words, it requires more iterations (approximately 16) to arrive to the intersection point of two curves. A relatively large negative offset of -6 dB even makes the turbo detector to diverge. In fact, as it is seen in Fig. 10(b), the system fails to find a trajectory leading to a large I_E^{DEC} and stops improving the performance after about two iterations and reaches the point of $I_E^{\mathrm{DEC}} \approx 0.052$.

To compare the results of Fig. 10 to the case of B channel, we have shown the corresponding results for a negative offset of -3 and -6 dB in Fig. 11. It is seen that the sensitivity to the mismatched SNR is lower, as compared to the case of C channel. In fact, since I_E^{EQ} for $I_A^{\mathrm{EQ}} = 0$ is larger at offset $= 0$ for B channel with respect to C channel (see Fig. 7), the EXIT charts of the equalizer and the decoder stand sufficiently apart to insure the turbo-detector convergence, even for a relatively large SNR offset of -6 dB.[12] So, the convergence of the turbo detector is less constrained by a negative SNR offset for a channel with a lower loss.

For a given channel, when there is no SNR offset, the limit of SNR for the convergence of the turbo detector (based on the same SISO al-

[11]Note that, for a given I_E^{EQ} $(=I_A^{\mathrm{DEC}})$, we can obtain an estimation of the BER at the decoder output [5], [20].

[12]For this offset value, considering Fig. 3, we need more than five iterations to attain the complete convergence.

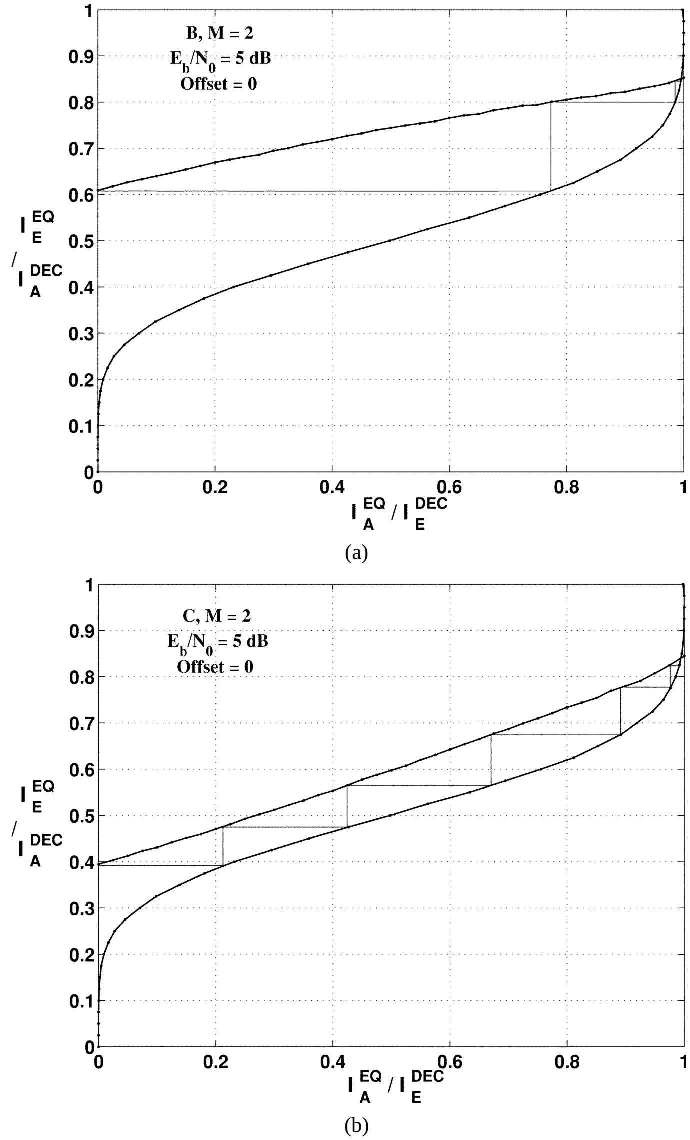

Fig. 8. Log-MAP turbo-detector EXIT charts, $M = 2, E_b/N_0 = 5$ dB. (a) B channel and (b) C channel.

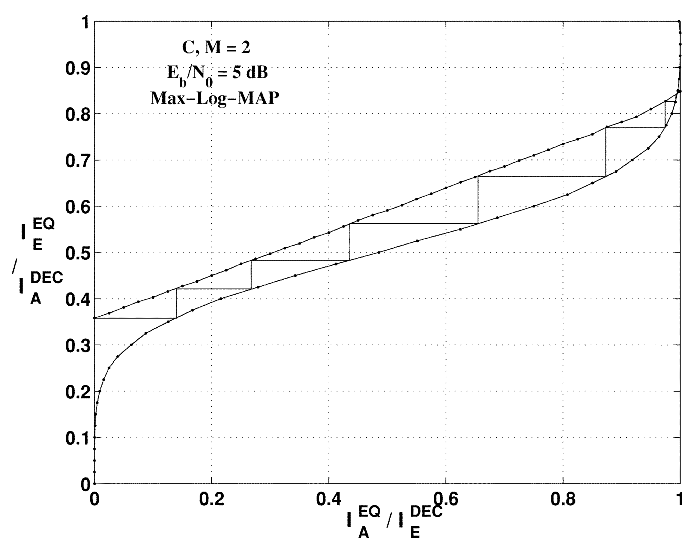

Fig. 9. Max-Log-MAP turbo-detector EXIT charts, C channel, $M = 2, E_b/N_0 = 5$ dB.

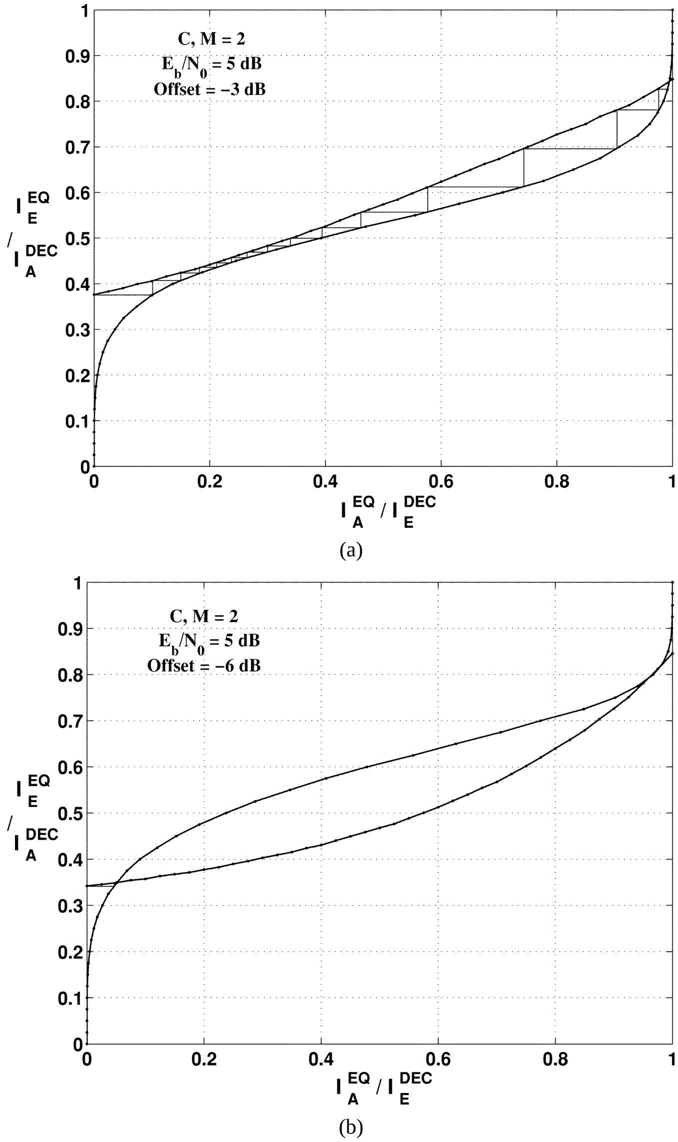

Fig. 10. Log-MAP turbo-detector EXIT charts, C channel, $M = 2, E_b/N_0 = 5$ dB, SNR offset $= -3, -6$ dB.

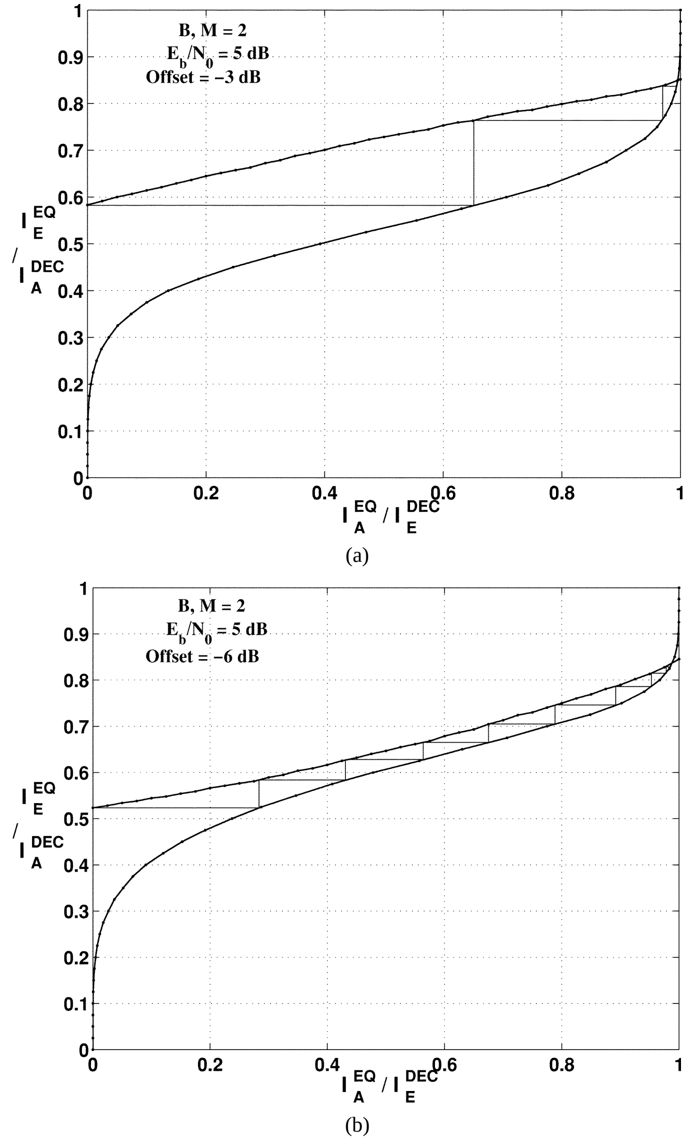

Fig. 11. Log-MAP turbo-detector EXIT charts, B channel, $M = 2, E_b/N_0 = 5$ dB, SNR offset $= -3, -6$ dB.

gorithm) is higher for greater M. However, in the case of convergence, the code with greater M yields lower error floor by approaching the point of $I_E^{\mathrm{DEC}} = 1$ faster [5], [22]. To see the effect of a mismatched SNR, Fig. 12 shows the effect of a negative offset of -3 dB on the convergence of the turbo detector for C channel and $M = 5$. It is seen that an offset of -3 dB makes the turbo detector to diverge for $M = 5$. For $M = 2$, however, the iterative detection can converge after several (although many) iterations [see Fig. 10(a)]. Notice the difference between the slopes of the EXIT charts of the Log-MAP decoder for $M = 2$ and $M = 5$ from Fig. 6. This is why the sensitivity of the Log-MAP turbo detector is more important for greater M, as seen in Section IV.

VI. Online Estimation of SNR at Receiver

In two previous sections, it was seen that the Log-MAP turbo detector can lose its advantage over max-Log-MAP and even SOVA for an important SNR mismatch, especially for a channel with a relatively high loss. This shows the necessity of an online SNR estimation while employing the Log-MAP turbo detection.

For the case of the Log-MAP turbo *decoder*, several methods are proposed for the estimation of SNR from the received signal [11], [12].

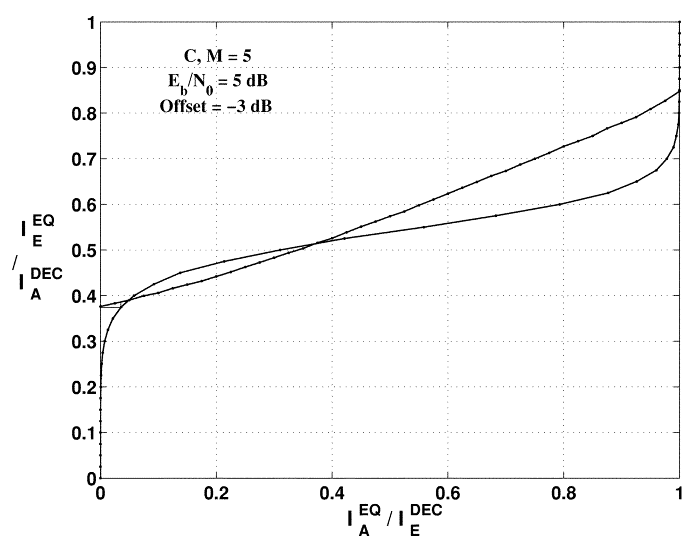

Fig. 12. Log-MAP turbo detector EXIT charts, C channel, $M = 5, E_b/N_0 = 5$ dB, SNR offset $= -3$ dB.

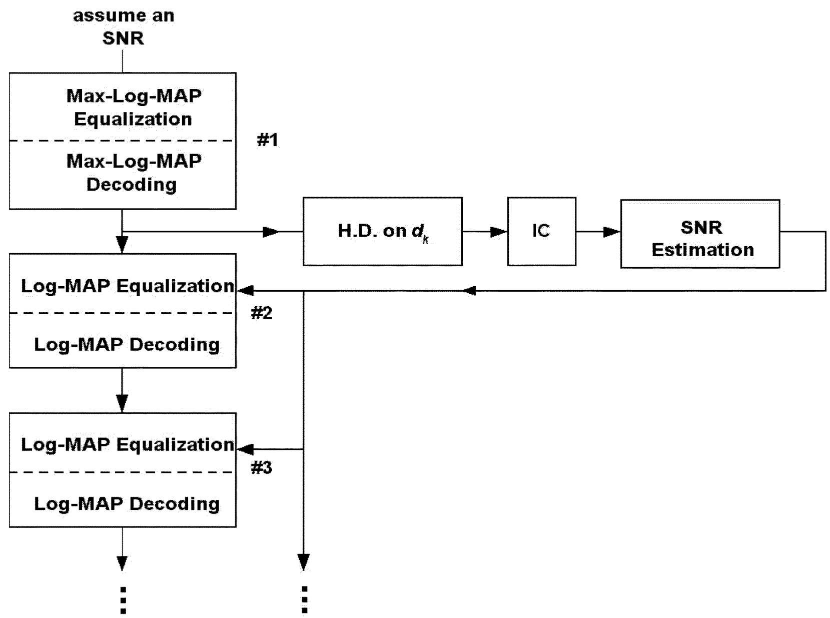

Fig. 13. Log-MAP turbo detection with online SNR estimation.

The estimation of SNR is relatively straightforward in this case, since we have only the effect of the additive noise on the transmitted symbols. However, this is not the case in the context of turbo *detection*. Here, the samples of the received signals are corrupted by ISI and we cannot estimate the SNR directly from these samples. An ISI cancellation should be performed before SNR estimation. We propose a relatively simple method for which we have inspired from the idea of a turbo equalizer, proposed in [4], [23].

A. Estimation of SNR

The block diagram of the proposed turbo detector with online SNR estimation is shown in Fig. 13. We assume an (approximate value for) SNR and use max-Log-MAP algorithm for the first iteration of the turbo detector (for both equalizer and decoder). Using the LLRs at the decoder output at the end of the first iteration, we estimate the SNR, which will be used in the next iterations.

Remember that the turbo detector based on Max-Log-MAP algorithms does not require SNR knowledge. In fact, whatever the assumed SNR value, the LLRs at the output of the decoder will be proportional to this SNR. Therefore, the hard decisions on the transmitted symbols at this moment (the sign of the LLRs) will be independent of the assumed SNR value. We use these hard decisions to estimate the SNR, as we explain in the following.

The hard decisions supply an interference canceller (IC), which tries to remove the ISI from the received symbols. The structure of the IC is shown in Fig. 14. In this figure, $\hat{d}_k$ are the hard decisions on transmitted symbols, obtained at the end of the first iteration of Max-Log-MAP turbo detector, and y_k are the samples at the matched filter output, which are in fact used by the turbo detector (see Fig. 1). Notice that IC should remove the ISI caused by both the channel and the channel matched filter. We have

$$q[n] = \sum_{i=0}^{L} h[i]h^*[i-n] - \delta[n] \tag{29}$$

where $\delta[n]$ is the Kronecker function. In this way, t_k will be the ISI-removed symbols, which are now used to estimate the SNR. Notice that

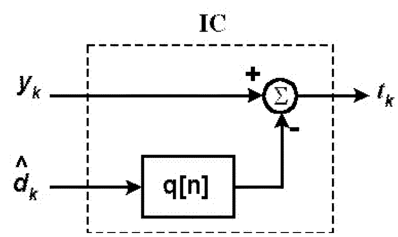

Fig. 14. Interference canceller used to remove the ISI from y_k.t_k are used for SNR estimation.

ISI may not be completely removed, because the data at the end of the first iteration may not be error free. We have

$$t_k = g(0)d_k + w_k \tag{30}$$

where w_k incorporates the filtered noise (by the matched filter) and the residual ISI at the IC output. We can write

$$\mathrm{E}\{(\Re e[t_k])^2\} = g[0]^2 + \sigma_w^2/2 \tag{31}$$

where σ_w^2 is the variance of w_k. Assuming that w_k contains no residual ISI, we have $\sigma_w^2 = g[0]\sigma_n^2$. The expected value in (31) is calculated by averaging over the samples in the block

$$\mathrm{E}\{(\Re e[t_k])^2\} \approx \frac{\sum_{k=1}^{N}(\Re e[t_k])^2}{N}. \tag{32}$$

N is the block size, which is in fact the interleaver size. Therefore, the variance of the noise σ_n^2 can be calculated as in

$$\sigma_n^2 = \frac{\sigma_w^2}{g[0]} \approx 2\left[\frac{\sum_{k=1}^{N}(\Re e[t_k])^2}{N\, g[0]} - g[0]\right]. \tag{33}$$

Notice that we can perform an exponential averaging on the estimated SNR on successive frames, to obtain better results. Notice also that the

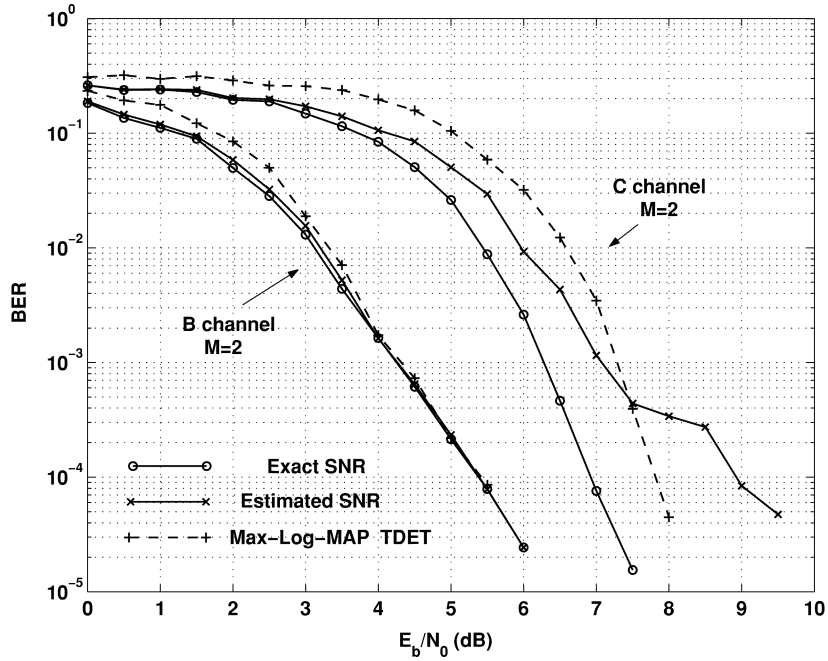

Fig. 15. Online SNR estimation for Log-MAP turbo detector, BER curves after five iterations; $M = 2$, interleaver size 500.

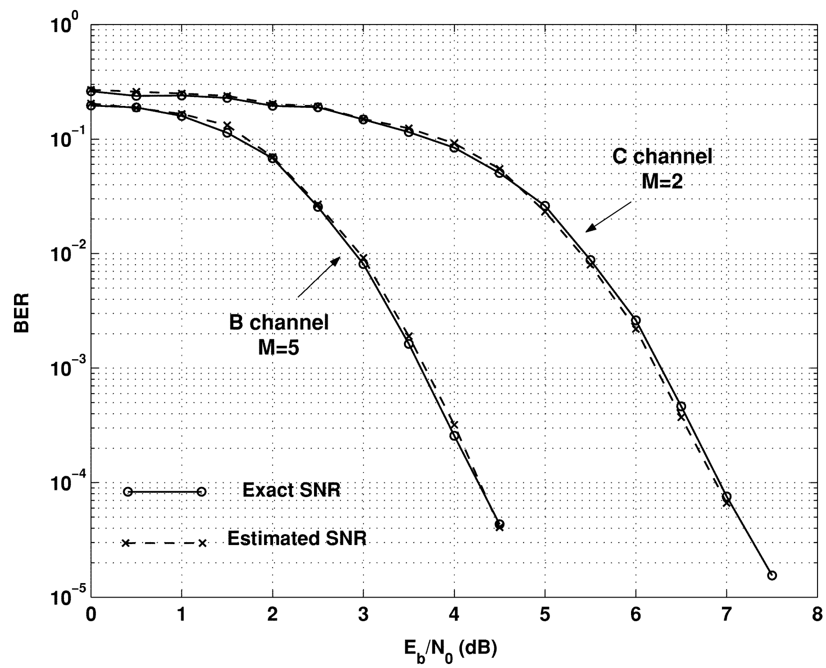

Fig. 16. Online SNR estimation for Log-MAP turbo detector; performing Max-Log-MAP turbo detection for updating estimated SNR; BER curves after five iterations; interleaver size 500.

proposed SNR estimation method can easily been generalized for the case of 4-QAM modulation.

Using the proposed method, simulation results are shown in Fig. 15, where BER curves are provided for two cases of estimated SNR and exactly known SNR for two channels B and C. The estimation is performed on frames of size $N = 500$ without any averaging on successive frames. For comparison, we have also shown the BER curves for Max-Log-MAP turbo detector.

It is seen that the proposed method gives satisfying results for a channel with a relatively moderate loss, as in B channel, but not for C channel, especially at low BER. The reason is, of course, the bias in the estimated SNR at the end of the first iteration.[13] At this moment, the LLRs at the decoder output are not reliable enough, so the IC outputs t_n contain a residual ISI, more or less important, which acts as a negative offset on SNR. For higher SNR values, this bias is more important since σ_n^2 is smaller. Remember from Sections IV and V that this bias is less critical for B channel.

[13]These results do not improve with a greater interleaver size. No improvement is obtained by repeating the proposed SNR estimation in the next iterations either, because of the propagation of the error on the estimated SNR.

If the noise variance can be assumed almost constant over several $(N_f \gg 1)$ frames (which is quite logical in practice), it is not necessary to estimate it for each block of symbols. So, we can perform Max-Log-MAP turbo detection on one frame, estimate σ_n^2 using the proposed method at the end of the last iteration, and to use this σ_n^2 for the $N_f - 1$ next frames, on which Log-MAP turbo detection is performed. Notice that by performing Max-Log-MAP turbo detection only once for estimating (the new) SNR, we do not move away from the performance of the Log-MAP turbo detector (with exact SNR knowledge). The SNR estimation should be repeated frequently enough.

Using this approach, simulation results are provided in Fig. 16 for both B and C channels. It is seen that this approach gives satisfying results.[14]

VII. Conclusion

In this work, the effect of an SNR mismatch at receiver on the performance of the Log-MAP turbo detector was studied. It was shown that the sensitivity to SNR mismatch is more important for a channel with a higher loss. Also, the sensitivity is more considerable for a greater encoder memory length M. While the Log-MAP detector takes its advantage over other suboptimal algorithms for greater M and channel loss, an SNR mismatch can make it lose its preference over them.

For a channel with a relatively small loss, by overtaking (the approximately known) SNR the problem may be resolved, since the Log-MAP turbo detector is specially sensitive to a negative SNR offset in this case. However, the online estimation of SNR seems to be unavoidable for a relatively high loss channel.

A simple method for online SNR estimation was proposed. For the case of slow variations in SNR, it was suggested to use the Max-Log-MAP turbo detector on a frame and to estimate the SNR form the ISI-removed signals at the end of the last iteration. The estimated SNR is then used for the detection of next blocks of symbols, based on Log-MAP turbo detection. Simulation results show satisfying results using this approach.

Acknowledgment

The author thanks C. Laot and R. Le Bidan, ENST-Bretagne, Brest, France, for their fruitful discussions and comments. He also thanks the anonymous reviewers for their useful suggestions for improving the paper.

References

[1] C. Berrou, A. Glavieux, and P. Thitimajshima, “Near Shannon limit error correcting coding and decoding: Turbo codes,” *Proc. IEEE Int. Conf. Communication (ICC)*, pp. 1740–1745, May 1993.

[2] C. Douillard, M. Jézéquel, C. Berrou, A. Picart, P. Didier, and A. Glavieux, “Iterative correction of intersymbol interference: Turbo-equalization,” *Eur. Trans. Telecommun.*, vol. 6, no. 5, pp. 507–511, 1995.

[3] G. Bauch and V. Franz, “A comparison of Soft-In/Soft-Out algorithms for turbo-detection,” in *Proc. Int. Conf. Telecommun. (ICT)*, vol. 2, Thessaloniki, Greece, June 1998, pp. 259–263.

[4] C. Laot, A. Glavieux, and J. Labat, “Turbo equalization: Adaptive equalization and channel decoding jointly optimized,” *IEEE J. Select. Areas Commun.*, vol. 19, pp. 1744–1752, Sept. 2001.

[5] M. Tüchler, R. Kötter, and A. C. Singer, “Turbo equalization: Principles and new results,” *IEEE Trans. Commun.*, vol. 50, pp. 754–767, May 2002.

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

[7] M. A. Khalighi, “Etude des systèmes de communication multi-antennes; capacité de canal et détection itérative,” Ph.D. dissertation, Nat. Polytechnique Institute of Grenoble (INPG), Grenoble, France, Oct. 2002.

[8] ——, “Sensitivity to SNR mismatch in interference cancelling based turbo equalizers,” *Proc. 13th IEEE Int. Symp. Personal, Indoor, and Mobile Radio Communication (PIMRC)*, pp. 379–383, Sept. 2002.

[9] M. Jordan and R. Nichols, “The effects of channel characteristics on turbo code performance,” in *Proc. MILCOM Conf.*, McLean, VA, Oct. 1996, pp. 17–21.

[10] A. Worm, P. Hoeher, and N. When, “Turbo-decoding without SNR estimation,” *IEEE Commun. Lett.*, vol. 4, pp. 193–195, June 2000.

[11] T. A. Summers and S. G. Wilson, “SNR mismatch and online estimation in turbo decoding,” *IEEE Trans. Commun.*, vol. 46, pp. 421–423, Apr. 1998.

[12] M. C. Reed and J. Asenstorfer, “A novel variance estimator for turbo-code decoding,” in *Proc. ICT Conf.*, Melbourne, Australia, 1997, pp. 173–178.

[13] J. Hagenauer, E. Offer, and L. Papke, “Iterative decoding of binary block and convolutional codes,” *IEEE Trans. Inform. Theory*, vol. 42, pp. 429–445, Mar. 1996.

[14] B. Vucetic and J. Yuan, *Turbo Codes: Principles and Applications*. Norwell, MA: Kluwer, 2000.

[15] P. Robertson, P. Hoeher, and E. Villebrun, “Optimal and sub-optimal maximum a posteriori algorithms suitable for turbo decoding,” *Eur. Trans. Telecommun.*, vol. 8, no. 2, pp. 119–125, 1997.

[16] J. G. Proakis, *Digital Communications*, 3rd ed. New York: McGraw-Hill, 1995.

[17] J. Hagenauer, “Source-controlled channel decoding,” *IEEE Trans. Commun.*, vol. 43, pp. 2449–2457, Sept. 1995.

[18] M. A. Khalighi, “Iterative Decoding and Detection (Turbo Methods), Principles and Related Algorithms,” Laboratoire des Images et des Singaux (LIS) , Grenoble, France, Internal Rep. 01/03, Feb. 2003.

[19] I. Fijalkow, A. Roumy, S. Ronger, D. Pirez, and P. Vila, “Improved interference cancellation for turbo-equalization,” *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 1, pp. 416–419, June 2000.

[20] S. ten Brink, “Convergence behavior of iteratively decoded parallel concatenated codes,” *IEEE Trans. Commun.*, vol. 49, pp. 1727–1737, Oct. 2001.

[21] T. J. Richardson, A. Shokrollahi, and R. Urbanke, “Design of capacity-appraoching irregular low-density parity-check codes,” *IEEE Trans. Inform. Theory*, vol. 47, pp. 671–686, Feb. 2001.

[22] R. Le Bidan, C. Laot, and A. Glavieux, “Analyze de la convergence en turbo-détection,” presented at the Proc. 18th GRETSI Conf. Signal Processing, Toulouse, France, Sept. 2001.

[23] A. Glavieux, C. Laot, and J. Labat, “Turbo-equalization over a frequency selective channel,” in *Proc. Int. Symp. Turbo-Codes*, Brest, France, Sept. 1997, pp. 96–102.

[24] M. A. Khalighi, “Iterative equalization and decoding with imperfect channel estimates,” in *Proc. Int. Conf. Telecommun. (ICT)* , vol. 1, Beijing, China, June 2002, pp. 88–92.

[14]Notice that here we assumed ideal channel estimation at the receiver. The imperfect channel estimation causes a degradation on the performance of the turbo detector, even when SNR is exactly known at receiver (see [24]). We expect that the proposed SNR estimation still gives an acceptable performance when applied on estimated channel coefficients. A detailed analysis should be performed to confirm this, however.