Transactions Papers

Extrinsic Information in Iterative Decoding: A Unified View

Giulio Colavolpe, *Associate Member, IEEE*, Gianluigi Ferrari, *Student Member, IEEE*, and Riccardo Raheli, *Member, IEEE*

***Abstract*—In this paper, we address the use of the extrinsic information generated by each component decoder in an iterative decoding process. The algorithm proposed by Bahl *et al.* (BCJR) and the soft-output Viterbi algorithm (SOVA) are considered as component decoders. In both cases, we consider, in a unified view, various feedback schemes which use the extrinsic information in different fashions. Numerical results for a classical rate-$1/2$ turbo code and a serially concatenated code transmitted over a memoryless additive white Gaussian noise (AWGN) channel are provided. The performance of the considered schemes leads to interesting remarks about the nature of the extrinsic information.**

***Index Terms*—Iterative decoding, soft-input/soft-output algorithms, turbo (de)coding.**

I. Introduction

In CONJUNCTION with the proposal of "turbo codes," based on a parallel concatenation of two *recursive systematic convolutional* (RSC) codes linked together by a *nonuniform* interleaver, a suboptimal decoding scheme based on iterative decoding has been introduced [1]. Although not widely known, the concept of iterative decoding was also independently introduced in [2] in the context of concatenated convolutional codes and simple block codes. The iterative decoding technique was extended to serially concatenated codes, based on a serial concatenation, through an interleaver, of an outer nonrecursive convolutional code and an inner recursive code [3]. The heart of the iterative decoding procedure is the use, in each component decoder, of an algorithm that computes the *a posteriori probability* (APP) of the information symbols or, more generally, a reliability value for each information symbol (and/or code symbol). The sequence of reliability values generated by a decoder is passed to the other one. In this way, each decoder takes advantage of the "suggestions" of the other one. To improve the correctness of its decisions, each decoder has to be fed with information which does not originate from itself [1], [4]. In [1], [2], the original concept of *extrinsic information* was introduced to identify the component of the generated reliability value which depends on redundant information introduced by the considered constituent code.[1]

A natural reliability value, in the binary case, is the *logarithmic likelihood ratio* (LLR), defined as

$$L(a_k) \triangleq \ln \frac{P\{a_k = +1 \,|\, \text{inputs}\}}{P\{a_k = -1 \,|\, \text{inputs}\}} \tag{1}$$

where the word "inputs" refers to all the decoder inputs. The LLR may be exactly computed employing the BCJR algorithm, which allows one to calculate the APPs $P\{a_k = i \,|\, \text{inputs}\}, i \in \{\pm 1\}$ [5]. The BCJR algorithm is the optimum algorithm to generate the sequence of APPs, but its computational complexity is large with respect to that of the Viterbi algorithm (VA). Besides "hard" symbol decisions, the soft-output Viterbi algorithm (SOVA) provides reliability information, which can be interpreted as an approximation of the LLRs [6]–[8].

For both the BCJR algorithm and SOVA, in the literature there exist essentially two methods to process the extrinsic information received by each decoder (and generated by the other one). In a first method, the extrinsic information at the input of a decoder is modeled as the output of an additive white Gaussian noise (AWGN) *meta-channel* [1], [4], [9]. In a second method, the extrinsic information is used to update the "*a priori*" probabilities which are used in the next decoding step, in the sense that the a posteriori probabilities computed by a decoder become *a priori* probabilities for the other one [10]–[12].

In this paper, we present a unified interpretation of these two methods and emphasize their commonalities and differences. More precisely, we show that, either using the BCJR algorithm or SOVA, the two methods only differ for a multiplicative factor used in the metric computation. When the input is modeled as a Gaussian random variable, this multiplicative factor depends on the variance and mean of the received LLRs, whereas, in the case of extraction of the *a priori* probabilities, it is a constant equal to $1/2$. We finally consider the use of a heuristic multiplicative parameter for both algorithms and evaluate the performance of the considered decoding schemes for various values of this parameter.

Paper approved by P. Hoeher, the Editor for Coding and Communication Theory of the IEEE Communications Society. This work was supported by Ministero dell'Università e della Ricerca Scientifica e Tecnologica (MURST), Italy. This paper was presented in part at the Global Telecommunications Conference (GLOBECOM '99), Rio de Janeiro, Brazil, December 1999. Manuscript received January 15, 2000; revised January 30, 2001.

The authors are with the Dipartimento di Ingegneria dell'Informazione, Università di Parma, I-43100 Parma, Italy.

Publisher Item Identifier S 0090-6778(01)10617-3.

[1]In [2], the extrinsic information is referred to as "refinement factor."

II. A Review on the use of the Extrinsic Information

The decoding process of turbo and serially concatenated codes is based on a suboptimal iterative processing in which each component decoder takes advantage of the extrinsic information produced by the other decoder at the previous step [1]. This iterative decoding process is made possible by employing soft-output component decoders. As an example, for the turbo code of rate $1/2$ described in [1], the turbo decoder is shown in Fig. 1. In the figure, blocks Π and Π^{-1} denote interleaver and deinterleaver, respectively, $\{x_k^{(j)}\}, j = 1, 2$, denote the channel output sequences and $\{z_k^{(j)}\}, j = 1, 2$, denote the extrinsic information sequences at the input of the jth soft-output decoder (i.e., produced by the other one). These sequences are derived, by means of an interleaver or a deinterleaver, from the sequences $\{w_k^{(j)}\}, j = 1, 2$, produced by the component decoders. Obviously, a serial concatenated decoder presents a serial concatenation, instead of a parallel concatenation, of two decoders.

In this section, we describe the possible methods to the use of the extrinsic information at the input of each decoder. To this purpose, we consider, without loss of generality, a soft-output decoder which receives a sequence $\{x_k\}$ of channel outputs and a sequence $\{z_k\}$ of extrinsic information values generated by the other decoder and produces a sequence $\{w_k\}$ of soft-output values. This is the case of decoder 2 in Fig. 1, but may be easily generalized to the other decoder with an extended vector notation for x_k. Moreover, we assume that the input sequence $\{z_k\}$ and the generated sequence $\{w_k\}$ are both relative to the sequence $\{a_k\}$ of information symbols. The proposed formulation can be generalized if the received extrinsic information is that of the code symbols $\{c_k\}$ and soft outputs relative to the code symbols $\{c_k\}$ are needed, besides those of the information symbols, as in the case of nonsystematic codes. In [2], [13], this generalization is carried out considering the BCJR algorithm; however, an extension to SOVA is straightforward. Hence, by assuming that the received and generated extrinsic information sequences are related to the information sequence $\{a_k\}$, we are implicitly assuming that the code is systematic. For simplicity, in Sections III and IV we refer to an RSC code (the component code of a turbo code and the inner code of a serially concatenated code). In the numerical results, we will consider the performance of both turbo codes and serially concatenated codes.

The channel outputs may be expressed as

$$x_k = c_k + n_k \tag{2}$$

where $\{n_k\}$ is a sequence of independent, zero-mean, real Gaussian random variables, with variance σ^2. In the original paper on turbo codes and turbo decoding [1], the input sequence $\{z_k\}$, i.e., the extrinsic information extracted from the reliability values of the information sequence $\{a_k\}$, is interpreted as the output of a Gaussian meta-channel. Specifically, it is assumed that

$$z_k = \eta_z a_k + n'_k \tag{3}$$

where the information symbols $\{a_k\}$ belong to the binary alphabet $\{\pm 1\}$, $\{n'_k\}$ are independent, zero-mean, real Gaussian

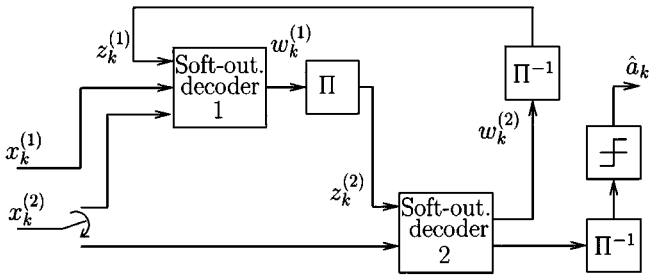

Fig. 1. Decoder for a turbo code of rate $1/2$.

random variables, with variance σ_z^2 and $\eta_z \triangleq |E\{z_k \,|\, a_k\}|$. In [1], it is noted that the Gaussian assumption, even if it is not satisfied for the first iterations, is a good approximation when the number of iterations increases. The values of η_z and σ_z^2 are estimated for each data block.

An alternative method, which does not require an estimation of η_z and σ_z^2, is proposed in [10], [11]. In this case, the extrinsic information z_k at the input of the considered decoder is used to extract a new estimate of the "*a priori*" probabilities to be employed in the new decoding step. In fact, each decoder interprets z_k as an approximation of the LLR of the *a priori* probabilities according to

$$z_k \simeq \ln \frac{P\{a_k = +1\}}{P\{a_k = -1\}} \tag{4}$$

which allows to derive [10]

$$\begin{aligned} P\{a_k = +1\} &\simeq \frac{e^{z_k}}{1 + e^{z_k}} \\ P\{a_k = -1\} &= 1 - P\{a_k = +1\} \simeq \frac{1}{1 + e^{z_k}}. \end{aligned} \tag{5}$$

Therefore, the APPs generated by a decoder are used as *a priori* probabilities by the other one.

III. BCJR Algorithm

We begin by summarizing the formulation of the BCJR algorithm [5], as given in [1], in order to introduce the used notation. Let us denote by ζ the number of states of each constituent encoder ($\zeta = 2^{\nu-1}$, where ν is the code constraint length) and S_k the state of the encoder at time k. The bit a_k is associated with the transition from state S_{k-1} to state S_k. The generated LLR may be expressed as

$$L(a_k) = \ln \frac{\sum_m \sum_{m'} \gamma_k(+1, m', m) \alpha_{k-1}(m') \beta_k(m)}{\sum_m \sum_{m'} \gamma_k(-1, m', m) \alpha_{k-1}(m') \beta_k(m)}. \tag{6}$$

The probability density functions $\gamma_k(i, m', m), i \in \{\pm 1\}, m', m \in 0, 1, \ldots, \zeta - 1$, are defined as

$$\begin{aligned} \gamma_k(i, m', m) \triangleq\ & p(R_k \,|\, a_k = i, S_k = m, S_{k-1} = m') \\ & \cdot P\{a_k = i \,|\, S_k = m, S_{k-1} = m'\} \\ & \cdot P\{S_k = m \,|\, S_{k-1} = m'\} \end{aligned} \tag{7}$$

where $R_k = (x_k, z_k)$ if z_k is interpreted as the output of a Gaussian meta-channel, or $R_k = x_k$ if z_k is used to update the *a priori* probabilities, $P\{a_k = i \,|\, S_k = m, S_{k-1} = m'\}$ is either one or zero depending on whether bit i is or is not associated

with the transition from state m' to state m, respectively, and $P\{S_k = m \,|\, S_{k-1} = m'\}$ is the transition probability.

The probability density functions $\alpha_k(m)$ and $\beta_k(m)$ may be calculated using the forward and backward recursions. As an example, in the case of $\alpha_k(m)$ we have

$$\alpha_k(m) = \frac{\sum_{m'} \sum_{i=0}^{1} \gamma_k(i, m', m) \alpha_{k-1}(m')}{\sum_{m''} \sum_{m'} \sum_{i=0}^{1} \gamma_k(i, m', m'') \alpha_{k-1}(m')}. \quad (8)$$

From (6) and (8), it is obvious that $\gamma_k(i, m', m)$ may be arbitrarily multiplied by any constant independent of m', m and i.

In the following subsections, we present the two mentioned methods for using the extrinsic information within a unified interpretation which, to our knowledge, has not been emphasized in the technical literature.

A. Extrinsic Information as Gaussian-Distributed Input

In this case, the information symbols are assumed independent and identically distributed, i.e., $P\{a_k = +1\} = P\{a_k = -1\} = 1/2$. Hence, $P\{S_k = m \,|\, S_{k-1} = m'\} = 1/2$ for each possible transition. Since $R_k = (x_k, z_k)$ and due to the assumed independence of x_k and z_k, we may write

$$\begin{aligned} &p(R_k \,|\, a_k = i, S_k = m, S_{k-1} = m') \\ &\quad = p(x_k \,|\, a_k = i, S_k = m, S_{k-1} = m') \\ &\qquad \cdot p(z_k \,|\, a_k = i, S_k = m, S_{k-1} = m'). \end{aligned} \quad (9)$$

Recalling (3) and the Gaussian assumption for z_k, we have

$$\begin{aligned} &p(z_k \,|\, a_k = i, S_k = m, S_{k-1} = m') \\ &\quad = p(z_k \,|\, a_k = i) \\ &\quad = \frac{1}{\sqrt{2\pi\sigma_z^2}} \exp\left\{-\frac{(z_k - i\eta_z)^2}{2\sigma_z^2}\right\} \\ &\quad = \frac{1}{\sqrt{2\pi\sigma_z^2}} \exp\left\{-\frac{z_k^2}{2\sigma_z^2} - \frac{\eta_z^2}{2\sigma_z^2}\right\} \exp\left\{\frac{i z_k \eta_z}{\sigma_z^2}\right\}. \end{aligned} \quad (10)$$

Thus, from (7) we may express the probability density function $\gamma_k(i, m', m)$ used in the forward and backward recursions as

$$\begin{aligned} &\gamma_k(i, m', m) \\ &\quad = \delta_k p(x_k \,|\, a_k = i, S_k = m, S_{k-1} = m') \\ &\qquad \cdot P\{a_k = i \,|\, S_k = m, S_{k-1} = m'\} \exp\left\{\frac{i z_k \eta_z}{\sigma_z^2}\right\} \end{aligned} \quad (11)$$

where δ_k is a suitable constant, independent of i, m' and m.

In this case, we may express the LLR (6) as

$$L(a_k) = 2\frac{z_k \eta_z}{\sigma_z^2} + w_k \quad (12)$$

where w_k is the generated extrinsic information defined as

$$w_k \triangleq \ln \frac{\sum_m \sum_{m'} \gamma'_k(+1, m', m) \alpha_{k-1}(m') \beta_k(m)}{\sum_m \sum_{m'} \gamma'_k(-1, m', m) \alpha_{k-1}(m') \beta_k(m)} \quad (13)$$

in which

$$\begin{aligned} \gamma'_k(i, m', m) \triangleq\ & \delta_k p(x_k \,|\, a_k = i, S_k = m, S_{k-1} = m') \\ & \cdot P\{a_k = i \,|\, S_k = m, S_{k-1} = m'\}. \end{aligned} \quad (14)$$

B. Extrinsic Information Used to Update the a Priori Probabilities

In this case, $R_k = x_k$ and [10]

$$\begin{aligned} &P\{S_k = m \,|\, S_{k-1} = m'\} \\ &\quad = \begin{cases} \frac{e^{z_k}}{1+e^{z_k}} & \text{if } P\{a_k = 1 \,|\, S_k = m, S_{k-1} = m'\} = 1 \\ \frac{1}{1+e^{z_k}} & \text{if } P\{a_k = -1 \,|\, S_k = m, S_{k-1} = m'\} = 1. \end{cases} \end{aligned} \quad (15)$$

Defining $\rho_k \triangleq e^{z_k/2}(1+e^{z_k})^{-1}$, we may express (15) as shown in (16) at the bottom of the page. Substituting in (7), we obtain

$$\begin{aligned} &\gamma_k(i, m', m) \\ &\quad = \rho_k p(x_k \,|\, a_k = i, S_k = m, S_{k-1} = m') \\ &\qquad \cdot P\{a_k = i \,|\, S_k = m, S_{k-1} = m'\} \exp\left\{\frac{i z_k}{2}\right\}. \end{aligned} \quad (17)$$

In this case, we may express the LLR (6) as

$$L(a_k) = z_k + w_k \quad (18)$$

where w_k, the generated soft-output, is defined as in (13) with the following definition

$$\begin{aligned} \gamma'_k(i, m', m) \triangleq\ & \rho_k p(x_k \,|\, a_k = i, S_k = m, S_{k-1} = m') \\ & P\{a_k = i \,|\, S_k = m, S_{k-1} = m'\}. \end{aligned} \quad (19)$$

C. Discussion and Heuristic Method

Based on the aforementioned results, we may observe that, with the exception of the irrelevant constants δ_k and ρ_k, the two methods in Sections III-A and III-B differ in the sense that the extrinsic information is weighted by different coefficients. This may be noted by comparing the expressions of the probability density functions $\gamma_k(i, m', m)$ (11) and (17) (the coefficient is η_z/σ_z^2 in the first method and $1/2$ in the second one), and the relations which implicitly define the extrinsic information w_k (12) and (18) (the coefficient is $2\eta_z/\sigma_z^2$ in the first method and 1 in the second one).

Based on this interpretation, a heuristic method may be conceived with the aim of evaluating an optimal weight for the extrinsic information. In this case, the extrinsic information z_k is weighted by a parameter θ to be optimized by trial and error. The performance of the receiver for various values of the parameter θ leads to useful remarks about the way the extrinsic information should be processed when the BCJR algorithm is used in the component decoders.

$$P\{S_k = m \,|\, S_{k-1} = m'\} = \begin{cases} \rho_k \exp\left\{\frac{z_k}{2}\right\} & \text{if } P\{a_k = 1 \,|\, S_k = m, S_{k-1} = m'\} = 1 \\ \rho_k \exp\left\{-\frac{z_k}{2}\right\} & \text{if } P\{a_k = -1 \,|\, S_k = m, S_{k-1} = m'\} = 1 \end{cases} = \rho_k \exp\left\{\frac{a_k z_k}{2}\right\} \quad (16)$$

IV. SOVA

An alternative to the use of the BCJR algorithm is represented by SOVA [6]–[8], whose soft-output is an approximation of the LLR. In the numerical results, we use the soft-output Viterbi decoder architecture proposed in [8] (with the updating rule proposed in [6]) in order to obtain a real-time scheme, whose complexity is roughly doubled with respect to that of a classical Viterbi decoder. The conclusions drawn when using this algorithm also hold for a suboptimal version of the BCJR algorithm, namely the "max-log-MAP" algorithm [14], since these two algorithms have been proven to be equivalent [15].

Denoting by N the number of samples of each data block, we define $\mathbf{x} \triangleq \{x_k\}_{k=1}^N$ and $\mathbf{z} \triangleq \{z_k\}_{k=1}^N$. We also denote by $\mathbf{R} \triangleq \{R_k\}_{k=1}^N$ the sequence of inputs of the considered decoder and $\mathbf{a} \triangleq \{a_k\}_{k=1}^N$. As in the case of the BCJR algorithm, $R_k = (x_k, z_k)$ if z_k is interpreted as the output of a Gaussian meta-channel, or $R_k = x_k$ if z_k is used to update the *a priori* probabilities. The maximum likelihood sequence detection (MLSD) strategy corresponds to the maximization of the following metric

$$\Lambda(\mathbf{a}) = \ln[p(\mathbf{R}\,|\,\mathbf{a})P\{\mathbf{a}\}] = \ln p(\mathbf{R}\,|\,\mathbf{a}) + \ln P\{\mathbf{a}\}. \quad (20)$$

This metric may be arbitrarily multiplied by any constant independent of the information sequence.

A. Extrinsic Information as Gaussian-Distributed Input

Since $R_k = (x_k, z_k)$ and due to the assumed independence of x_k and z_k, we have

$$\Lambda(\mathbf{a}) = \ln p(\mathbf{x}\,|\,\mathbf{a}) + \ln p(\mathbf{z}\,|\,\mathbf{a}) + \ln P\{\mathbf{a}\}. \quad (21)$$

The probability density functions may be expressed as

$$p(\mathbf{x}\,|\,\mathbf{a}) = \frac{1}{(2\pi\sigma^2)^{\frac{N}{2}}} \exp\left\{-\frac{1}{2\sigma^2}\sum_{k=1}^{N}(x_k - c_k)^2\right\} \quad (22)$$

$$p(\mathbf{z}\,|\,\mathbf{a}) = \frac{1}{(2\pi\sigma_z^2)^{\frac{N}{2}}} \exp\left\{-\frac{1}{2\sigma_z^2}\sum_{k=1}^{N}(z_k - a_k\eta_z)^2\right\}. \quad (23)$$

In this case, the information symbols are assumed independent and identically distributed, i.e., $P\{a_k = +1\} = P\{a_k = -1\} = 1/2$. Therefore

$$P\{\mathbf{a}\} = \frac{1}{2^N}. \quad (24)$$

Substituting (22), (23), and (24) in (21) and discarding terms independent of the information sequence, we obtain the equivalent metric

$$\Lambda(\mathbf{a}) = \sum_{k=1}^{N}\left(\frac{x_k c_k}{\sigma^2} + \frac{z_k a_k \eta_z}{\sigma_z^2}\right) \quad (25)$$

which may be recursively computed adopting the following branch metrics

$$\lambda_k(\mathbf{a}) = \frac{x_k c_k}{\sigma^2} + \frac{z_k a_k \eta_z}{\sigma_z^2}. \quad (26)$$

SOVA does not produce soft-outputs by considering all paths in the trellis diagram as in the case of the BCJR algorithm, but only two paths—the maximum likelihood path and its strongest competitor. In the case of a binary RSC code, paths terminating in the same state S_k are relative to different information symbols, i.e., $a_k = 1$ and $a_k = -1$. We denote the corresponding code symbols by c_k^{+1} and c_k^{-1} and the corresponding cumulated metrics by Λ_k^{+1} and Λ_k^{-1}, respectively. Assuming that the winning path includes state S_k at time k, an initial reliability value of symbol a_k is obtained by considering the absolute value of the difference between the cumulated metrics of the two paths terminating in state S_k.

Let us consider the case $\Lambda_k^{+1} > \Lambda_k^{-1}$. An initial reliability value is [6]–[8]

$$\begin{aligned}\Lambda_k^{+1} - \Lambda_k^{-1} &= \lambda_k^{+1} - \lambda_k^{-1} + \Lambda_{k-1}^{+1} - \Lambda_{k-1}^{-1} \\ &= \frac{2\eta_z z_k}{\sigma_z^2} + \frac{x_k}{\sigma^2}\left(c_k^{+1} - c_k^{-1}\right) + \Lambda_{k-1}^{+1} - \Lambda_{k-1}^{-1} \\ &= \frac{2\eta_z z_k}{\sigma_z^2} + \Xi \end{aligned} \quad (27)$$

where $\Xi \triangleq x_k/\sigma^2(c_k^{+1} - c_k^{-1}) + \Lambda_{k-1}^{+1} - \Lambda_{k-1}^{-1}$. Similarly, in the case $\Lambda_k^{-1} > \Lambda_k^{+1}$, we have $\Lambda_k^{-1} - \Lambda_k^{+1} = -2\eta_z z_k/\sigma_z^2 - \Xi$. In general, the initial reliability value may be expressed as

$$\left|\Lambda_k^{+1} - \Lambda_k^{-1}\right| = a_k\frac{2\eta_z z_k}{\sigma_z^2} + a_k\Xi. \quad (28)$$

This value is then updated at successive time instants $(k+1, k+2, \ldots)$, according to a suitable rule [6]–[8]. Denoting by v_k the final reliability value derived from (28), a reasonable definition of the extrinsic information w_k of symbol a_k is

$$w_k \triangleq a_k v_k - \frac{2\eta_z z_k}{\sigma_z^2}. \quad (29)$$

B. Extrinsic Information Used to Update the a Priori Probabilities

In this case, $R_k = x_k$ and the decoder assumes that $P\{a_k = +1\}$ and $P\{a_k = -1\}$ are given by (5). Since

$$P\{\mathbf{a}\} = \prod_{k=1}^{N} P\{a_k\} \quad (30)$$

and $p(\mathbf{x}\,|\,\mathbf{a})$ is given by (22), substituting (22) and (30) in (20) and discarding terms independent of the information sequence, we obtain

$$\Lambda(\mathbf{a}) = \sum_{k=1}^{N}\left[\frac{x_k c_k}{\sigma^2} + \ln P\{a_k\}\right]. \quad (31)$$

Adding the constant $K \triangleq \sum_{k=1}^{N}[z_k/2 + \ln(1 + e^{z_k})]$, independent of $\mathbf{a}$, we have the equivalent metric

$$\Lambda(\mathbf{a}) = \sum_{k=1}^{N}\left[\frac{x_k c_k}{\sigma^2} + \frac{a_k z_k}{2}\right] \quad (32)$$

and the corresponding branch metrics

$$\lambda_k(\mathbf{a}) = \frac{x_k c_k}{\sigma^2} + \frac{a_k z_k}{2}. \quad (33)$$

In this case, the extrinsic information at the decoder output may be obtained as

$$w_k \triangleq a_k v_k - z_k. \quad (34)$$

C. Discussion and Heuristic Method

As in the case of the BCJR algorithm, the two methods differ for the constant which multiplies the received extrinsic infor-

mation z_k, both in the expression of the branch metrics (26) and (33) (η_z/σ_z^2, which appears in the first method, is substituted by $1/2$ in the second one), and in the definition of the soft-output w_k (29) and (34) (in this case, the constant is $2\eta_z/\sigma_z^2$ in the first method and 1 in the second one).

In [9], the reliability value z_k at the input of each component decoder is normalized by multiplying it by the factor $2\eta_z/\sigma_z^2$, and used to update the *a priori* probabilities. Although [9] claims to use the second method, because of this normalization the method actually used is the first one.

Even in this case, a heuristic method may be conceived by introducing a weighting parameter θ to be optimized by trial and error.

V. Numerical Results

The performance of the proposed decoding schemes is assessed for the classical turbo code of rate $1/2$, 16-state RSC constituent codes with generators $G_1 = 37, G_2 = 21$ (octal notation), and 256×256 nonuniform interleaver described in [1], and for a serial concatenated code of rate $1/4$, outer 4-state nonrecursive nonsystematic code with generators $G_1 = 7, G_2 = 5$ and inner 4-state RSC code with generators $G_1 = 5, G_2 = 7$ and 64×64 nonuniform interleaver [3].[2] The considered component decoders are based on the BCJR algorithm or the soft-output Viterbi decoder architecture proposed in [8]. We refer to the method described in Sections III-A and IV-A, in which the extrinsic information is assumed Gaussian, as *first method*; similarly, the method described in Sections III-B and IV-B, in which the extrinsic information is used to update the *a priori* probabilities, and the heuristic method described in Sections III-C and IV-C are referred to as *second method* and *third method*, respectively. We consider first the performance of the turbo code (in Figs. 2–4) and then the performance of the serially concatenated code (in Figs. 5–7). In the following simulation results, the performance is expressed in terms of bit error rate (BER) versus E_b/N_0, E_b being the received signal energy per information bit and N_0 the one-sided noise power spectral density.

In Fig. 2, the performance for the BCJR algorithm is shown for various numbers of iterations. It may be observed that the second method, in which the extrinsic information is used to update the *a priori* probabilities, corresponds to a better use of the extrinsic information with respect to the first method, which models the extrinsic information as a Gaussian-distributed random variable. Specifically, the second method exhibits a BER of 10^{-5} for a value of E_b/N_0 of approximately 0.7 dB. Moreover, the third (heuristic) method does not give any improvement. In fact, for each iteration and each signal-to-noise ratio (SNR), the best value of θ is $1/2$, which corresponds to the second method.

For the first method, Fig. 3 shows the behavior of the average value of the ratio η_z/σ_z^2 for the extrinsic information at the input of the first decoder as a function of the number of iterations and for various values of SNR. It may be observed that, almost independently of the considered iteration number, for values of SNR below a convergence threshold (about 0.7 dB), this ratio takes on values greater than $1/2$. Therefore, in the case of the first method the extrinsic information is overweighted. This has been previously observed in [1], in which a heuristic normalization of the extrinsic information has been proposed with the aim of improving the performance at low SNR. We may conclude that in the case of the BCJR algorithm, the second method corresponds to a better use of the extrinsic information and that the first method is asymptotically optimal for a SNR above 0.7 dB and a sufficiently large number of iterations. In addition, the second method does not require the estimation of the ratio η_z/σ_z^2.

We performed similar simulations considering SOVA as component decoder. The performance for the three methods considered in Section IV is shown in Fig. 4. As expected, for any of

[2]In the case of a recursive code, the generator G_1 refers to the feedback line, whereas in the case of a nonrecursive code the same generator refers to the first generated code symbol.

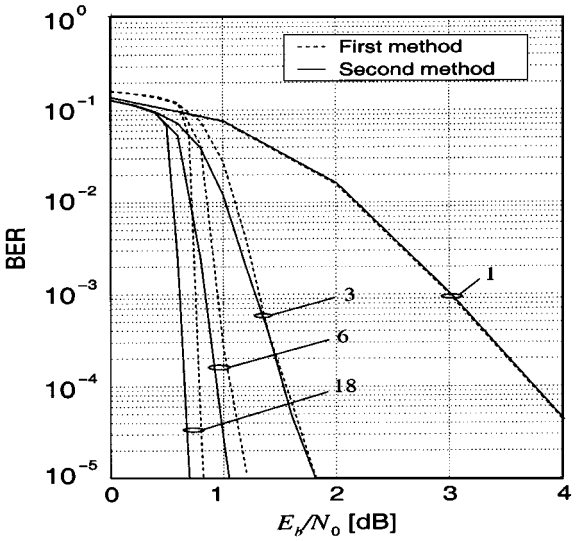

Fig. 2. BER of a turbo code and the BCJR algorithm. The extrinsic information generated by each decoder is either modeled as a Gaussian-distributed random variable (first method) or used to update the *a priori* probabilities (second method). The considered numbers of iterations are 1, 3, 6, and 18.

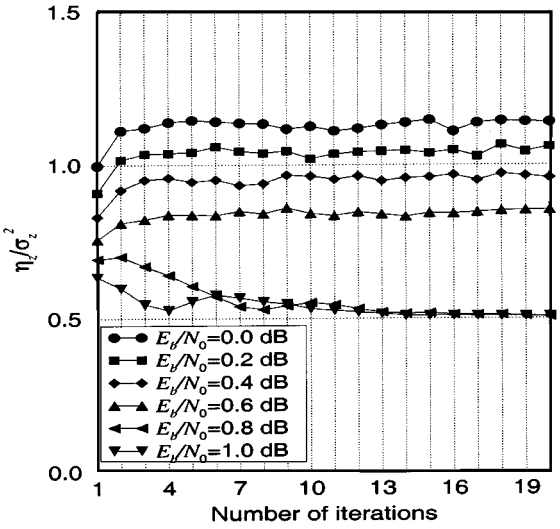

Fig. 3. Average value of ratio η_z/σ_z^2 versus the number of iterations, for various values of SNR and a turbo code. The component decoders use the BCJR algorithm. The extrinsic information generated by each decoder is modeled as a Gaussian-distributed random variable (first method).

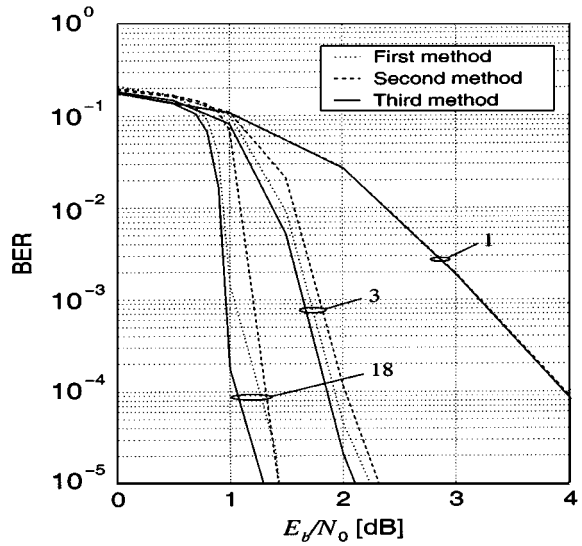

Fig. 4. BER of the considered detection schemes for a turbo code and SOVA. The extrinsic information generated by each decoder is either modeled as a Gaussian-distributed random variable (first method) or used to update the *a priori* probabilities (second method) or heuristically weighted (third method). The considered numbers of iterations are 1, 3, and 18.

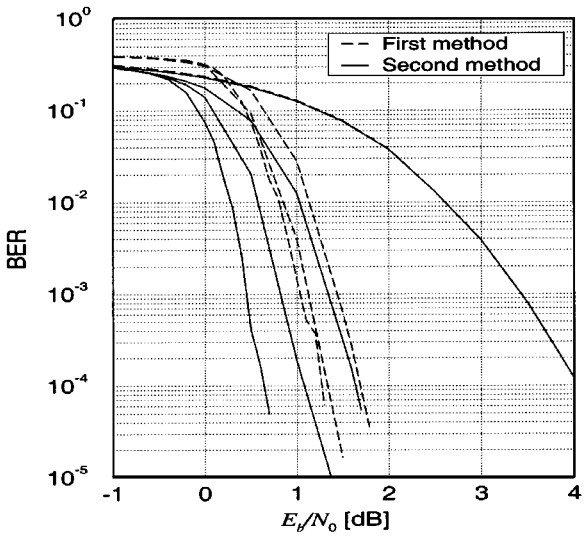

Fig. 5. BER of a serially concatenated code and BCJR algorithm. The considered numbers of iterations are 1, 3, 6, and 18.

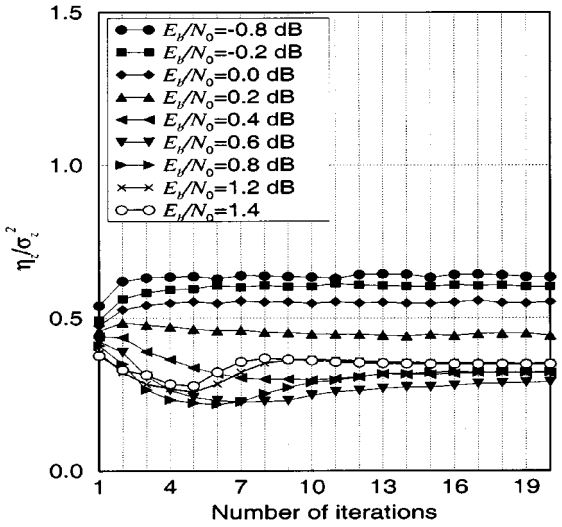

Fig. 6. Average value of ratio η_z/σ_z^2 versus number of iterations, for various values of SNR and a serially concatenated code. The component decoders use the BCJR algorithm. The extrinsic information generated by each decoder is modeled as a Gaussian-distributed random variable (first method).

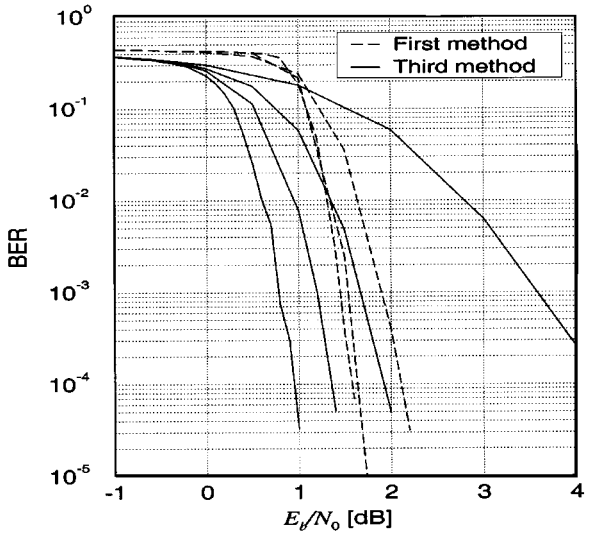

Fig. 7. BER of a serially concatenated code and SOVA. The considered numbers of iterations are 1, 3, 6, and 18.

the three methods, the performance degrades with respect to that of the corresponding scheme which uses the BCJR algorithm, due to the suboptimality of SOVA (compare with Fig. 2). Using SOVA, we may note that, unlike the BCJR algorithm, the best method is the heuristic method, by considering a value $\theta \simeq 0.4$ (optimal for any number of iterations). Moreover, in this case the second method is even worse than the first one. As observed in [9], [12], SOVA overestimates the reliability values—the obtained results are consistent with these references. In fact, the coefficient θ multiplies the extrinsic information z_k generated by the other decoder; hence, a reduced value of θ "compresses" the sequence $\{z_k\}$, correcting the overestimation. An analysis of the behavior of the average value of the ratio η_z/σ_z^2 in this case, shows that it does not tend to the optimal value $\theta = 0.4$ when the number of iterations increases and the SNR is sufficiently high. A simple conclusion is that in this case the first method is not asymptotically optimal.

As for the considered turbo code, considering the serial concatenated code in conjunction with the BCJR algorithm, the optimal method proves to be the second one. In Fig. 5 we consider the performance of the first two methods because for values of θ different from 0.5 the performance degrades. Even in this case, the first method is not asymptotically optimal. In fact, Fig. 6 shows that the ratio η_z/σ_z^2 at the input of the first encoder tends to a value approximately equal to 0.35, whereas the optimal performance was obtained with the second method, i.e., with $\theta = 0.5$.

Similarly to the case of turbo codes, when using SOVA to iteratively decode the considered serially concatenated code, the best method is the third one, and the optimal value of θ is approximately 0.3. Fig. 7 shows the performance of the first and

third method. This is consistent with the analysis of the limit of ratio η_z/σ_z^2. We can conclude that the first method is asymptotically optimal in this case.

VI. Conclusion

In this paper, iterative decoding schemes based on the BCJR algorithm and SOVA for the component decoders have been considered. In both cases, we presented a unified interpretation of different methods for using the extrinsic information: as a Gaussian-distributed random variable, as an *a priori* probability or heuristically by introducing a variable weight θ. In the case of the BCJR algorithm and turbo codes, the best method consists in updating the *a priori* probabilities: a BER of 10^{-5} is obtained with $E_b/N_0 = 0.7$ dB. This performance was also achieved in [1] where the extrinsic information is modeled as a Gaussian-distributed random variable and heuristically normalized. The same conclusions hold for serially concatenated codes with the BCJR algorithm. In the case of SOVA, the optimal value of the parameter θ is less than 0.5. This is consistent with the known overestimation effect of this algorithm [9], [12]: when considering turbo codes the optimal value proves to be 0.4, whereas for serially concatenated codes it is 0.3. By evaluating, with the fist method, the ratio η_z/σ_z^2 of the extrinsic information generated by the second decoder as a function of the number of iterations and for various values of SNR, we assessed the asymptotical optimality of this method.

Acknowledgment

Reference [2] was brought to the authors' attention during the review process. An anonymous reviewer is gratefully acknowledged.

References

[1] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: Turbo-codes," *IEEE Trans. Commun.*, vol. 44, pp. 1261–1271, Oct. 1996.

[2] J. Lodge, R. Young, P. Hoeher, and J. Hegenauer, "Separable MAP 'filters' for the decoding of product and concatenated codes," *Proc. IEEE Int. Conf. Commun. (ICC'93)*, pp. 102–106, June 1993.

[3] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "Serial concatenation of interleaved codes: Performance analysis, design, and iterative decoding," *IEEE Trans. Inform. Theory*, vol. 44, pp. 909–926, May 1998.

[4] B. Sklar, "A primer on turbo code concepts," *IEEE Commun. Mag.*, pp. 94–102, Dec. 1997.

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

[6] G. Battail, "Pondération des symboles décodé par l'algorithme de Viterbi," *Ann. Télecommun. Fr.*, vol. 42, no. 1–2, pp. 31–38, Jan. 1987.

[7] J. Hagenauer and P. Hoeher, "A Viterbi algorithm with soft-decision outputs and its applications," in *Proc. IEEE Global Commun. Conf. (GLOBECOM'89)*, Dallas, TX, Nov. 1989, pp. 1680–1686.

[8] C. Berrou, P. Adde, E. Angui, and S. Faudeil, "A low complexity soft-output Viterbi decoder architecture," *Proc. IEEE Int. Conf. Commun. (ICC'93)*, pp. 737–740, June 1993.

[9] L. Papke, P. Robertson, and E. Villerbrun, "Improved decoding with the SOVA in a parallel concatenated (turbo-code) scheme," in *Proc. Intern. Conf. Comm. (ICC'96)*, June 1996, pp. 102–106.

[10] P. Robertson, "Illuminating the structure of code and decoder of parallel concatenated recursive systematic (turbo) codes," in *Proc. IEEE Global Commun. Conf. (GLOBECOM'94)*, San Francisco, CA, Dec. 1994, pp. 1298–1303.

[11] C. Douillard, M. Jezequel, C. Berrou, A. Picart, P. Didier, and A. Glavieux, "Iterative correction of intersymbol interference: Turbo-equalization," *Europ. Trans. on Telecommun. (ETT)*, vol. 6, pp. 507–511, Sept./Oct. 1995.

[12] L. Lin and R. Cheng, "Improvements in SOVA-based decoding for turbo codes," in *Proc. Int. Conf. Comm. (ICC'97)*, Montréal, QC, Canada, June 1997, pp. 1473–1478.

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

[14] P. Robertson, E. Villerbrun, and P. Hoeher, "A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain," in *Proc. Intern. Conf. Comm. (ICC'95)*, June 1995, pp. 1009–1013.

[15] M. P. C. Fossorier, F. Burkert, S. Lin, and J. Hagenauer, "On the equivalence between SOVA and max-log-MAP decodings," *IEEE Commun. Lett.*, vol. 2, pp. 137–139, May 1998.

Giulio Colavolpe (S'96–A'00) was born in Cosenza, Italy, in 1969. He received the Dr. Ing. degree in telecommunications engineering (*cum laude*) from the University of Pisa, Pisa, Italy, in 1994, and the Doctoral degree in information technology from the University of Parma, Parma, Italy, in 1998.

From December 1997 to October 1999 he was a Research Associate with the University of Parma, and, since November 1999, has been a Research Professor with the University. In 2000, he was a Visiting Scientist with the Insitut Eurécom, Valbonne, France. His main research interests include digital transmission theory, channel coding and signal processing.

Gianluigi Ferrari (S'01) was born in Parma, Italy, in 1974. He received the Dr. Ing. degree (Laurea) in electrical engineering (*cum laude*) from the University of Parma, in 1998. Since November 1998, he has been a Ph.D. student with the University of Parma.

Since July 2000 he has been a Visiting Scholar with the Communication Sciences Institute, University of Southern California, Los Angeles. His main research interests include digital transmission and detection theory, channel coding, and iterative decoding techniques.

Riccardo Raheli (M'87) received the Dr. Ing. degree (Laurea) in electrical engineering (*summa cum laude*) from the University of Pisa, Pisa, Italy, in 1983, the M.S. degree in electrical and computer engineering from the University of Massachusetts, Amherst, in 1986, and the Doctoral degree (Perfezionamento) in electrical engineering (*summa cum laude*) from the Scuola Superiore di Studi Universitari e di Perfezionamento (currently "S. Anna"), Pisa, in 1987.

From 1986 to 1988 he was a Project Engineer with Siemens Telecomunicazioni, Cassina de' Pecchi, Milan, Italy. From 1988 to 1991, he was a Research Professor with the Scuola Superiore di Studi Universitari e di Perfezionamento (S. Anna). In 1990, he was a Visiting Assistant Professor with the University of Southern California, Los Angeles. Since 1991, he has been with the University of Parma, Parma, Italy, where he is currently a Professor of Telecommunications. His scientific interests are in the general area of statistical communication theory, with special attention toward digital transmission systems, data-sequence detection techniques, digital signal processing, and adaptive algorithms for telecommunications. His research activity has lead to numerous scientific publications in leading international journals and conference proceedings and a few industrial patents.

Dr. Raheli has served on the Editorial Board of the IEEE TRANSACTIONS ON COMMUNICATIONS as an Editor for Detection, Equalization, and Coding since 1999.