

Noncoherent Iterative (Turbo) Decoding

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

Abstract—Recently, noncoherent sequence detection schemes for coded linear and continuous phase modulations have been proposed, which deliver hard decisions by means of a Viterbi algorithm. The current trend in digital transmission systems toward iterative decoding algorithms motivates an extension of these schemes. In this paper, we propose two noncoherent soft-output decoding algorithms. The first solution has a structure similar to that of the well-known algorithm by Bahl *et al.* (BCJR), whereas the second is based on noncoherent sequence detection and a reduced-state soft-output Viterbi algorithm.

Applications to the combined detection and decoding of differential or convolutional codes are considered. Further applications to noncoherent iterative decoding of turbo codes and serially concatenated interleaved codes are also considered. The proposed noncoherent detection schemes exhibit moderate performance loss with respect to corresponding coherent schemes and are very robust to phase and frequency instabilities.

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

I. INTRODUCTION

NONCOHERENT detection of digital signals is an attractive strategy in situations where carrier phase recovery is difficult. Most of the drawbacks of phase-locked loop (PLL) circuits, used to approximately implement coherent detection, may be avoided by means of noncoherent detectors. Specifically, typical problems of PLLs such as false-locks, phase slips, or losses of lock caused by severe fading, Doppler shifts, phase noise, or oscillator frequency instabilities are simply by-passed.

In the technical literature, a growing effort has been recently devoted toward the derivation of improved noncoherent detection or decoding schemes. Two main classes of algorithms have been proposed. *Multiple-symbol differential detection* (MSDD) [1]–[8] is based on maximum-likelihood detection of a block of information symbols based on a corresponding block of signal observations. *Noncoherent sequence detection* (NSD) [9]–[14] approximates the optimal noncoherent maximum-likelihood sequence detection strategy in order to realize simple suboptimal schemes based on the Viterbi algorithm (VA). In both cases, the performance approaches that of ideal coherent detection. Although the receivers presented in [1]–[14] may be used when

channel codes are employed, they are not applicable when iterative processing is required, as for turbo codes [15] or serially concatenated interleaved codes [16].

The need for noncoherent decoding algorithms suited for iterative processing is mainly due to the fact that, for the peculiar decoding process and typical operative signal-to-noise ratio of turbo codes, phase-tracking schemes may deliver a highly unreliable phase estimate or require a systematic use of pilot symbols to avoid tracking losses [17]–[19]. The first application of noncoherent decoding to turbo codes was proposed in [20], where the output of a conventional differential detector is used to feed a standard turbo decoder. In [21], a new receiver for convolutionally encoded, interleaved and differentially encoded M -ary phase-shift keying (M -PSK) was proposed, based on an improved version of MSDD in order to allow iterative decoding. An extension to turbo codes was proposed in [22]. A theoretical analysis of this concatenated code based on a cut-off rate bound was proposed in [23] for noniterative decoding. The iterative noncoherent decoding of convolutionally encoded M -PSK signals was also considered in [24] and [25]. An algorithm similar to that proposed in [24] was presented in [26]. Finally, an iterative algorithm specifically tailored for noisy-phase channels was proposed in [27]. The study of noncoherent iterative detection techniques is also motivated by a general information theoretic result which proves that the capacity of noncoherent channels is very similar to that of corresponding coherent channels [28], [29].

In this paper, we propose two noncoherent soft-output decoding algorithms for coded linear modulations. The first algorithm is derived from an approximation of a minimum symbol-error probability decoder, with a structure similar to that of the well-known algorithm by Bahl *et al.* (BCJR) [30]. While the soft-output decoding strategy proposed in [21] and [22] may be viewed as the equivalent of MSDD based on a minimum symbol-error probability criterion, this algorithm is the minimum symbol-error probability counterpart of NSD [13], [14]. The second algorithm is based on NSD [13], [14], in which reliability information about each symbol is obtained by means of a soft-output VA (SOVA) [31]–[33]. Contrary to the first scheme which operates under a minimum *symbol* error probability criterion, this second proposed algorithm follows a minimum *sequence* error probability criterion. In order to perform iterative processing, reliability information is heuristically obtained using the SOVA principle. We remark that NSD, as proposed in [13] and [14], incorporates reduced-state sequence detection (RSSD) [34], [35] techniques; hence, the second proposed algorithm combines the benefits of trellis-state reduction, use of soft-outputs and enhanced noncoherent detection.

Applications to combined detection and decoding of differential or convolutional codes are considered. The algorithms are

Paper approved by S. S. Pietrobon, the Editor for Coding Theory and Techniques of the IEEE Communications Society. Manuscript received May 31, 1999; revised November 9, 1999 and February 7, 2000. This work was supported by Ministero dell'Università e della Ricerca Scientifica e Tecnologica (MURST), Italy. This paper was presented in part at the International Conference on Communications (ICC'99), Vancouver, BC, Canada, June 1999, and the Global Telecommunications Conference (GLOBECOM'99), Rio de Janeiro, Brazil, December 1999.

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

Publisher Item Identifier S 0090-6778(00)07524-3.

well suited for iterative processing applications and are explicitly employed for noncoherent decoding of turbo codes and serially concatenated interleaved codes. Extensions to the case of linear modulation with intersymbol interference (ISI) or continuous phase modulation may be dealt with using the methods described in [13] and [14].

The paper is organized as follows. In the next section, we describe the assumed system model. A noncoherent BCJR-type algorithm is proposed in Section III. A soft-output NSD algorithm is introduced in Section IV. The use of these algorithms in iterative decoding is considered in Section V. Numerical results are presented in Section VI, and conclusions are drawn in Section VII.

II. SYSTEM MODEL

An information sequence $\{a_k\}$, composed of independent and identically distributed (i.i.d.) symbols belonging to an M -ary alphabet, is mapped into a code sequence $\{c_k\}$ by means of some coding rule. The code sequence is further mapped by a modulator into a time-continuous signal which undergoes a phase rotation θ and is transmitted over an additive white Gaussian noise (AWGN) channel with two-sided power spectral density $N_0/2$. The phase rotation θ is initially assumed to be constant during the entire transmission and modeled as an independent random variable with uniform distribution in the interval $[0, 2\pi)$. However, an approximation of memory truncation is used, as in [13] and [14], which has the convenient feature of allowing us to remove the constant phase assumption and encompass time-varying phase models.

It may be easily shown that the sampled output of a filter matched to the shaping pulse is a sufficient statistic for optimal detection of the information sequence [13]. In the absence of ISI, this sampled output may be expressed as

$$x_k = c_k e^{j\theta} + n_k \quad (1)$$

where $\{n_k\}$ are i.i.d. complex noise samples with independent components, each with variance σ^2 .

III. NONCOHERENT BCJR-TYPE ALGORITHM

In the derivation of the algorithm, we assume that each information symbol generates a single complex code symbol. However, a more general case may be easily dealt with by a vector notation. This assumption is relative to the important cases of trellis-coded modulation and differential encoding, whereas an extended vector notation would be required for a convolutional code. This assumption is adopted in order to simplify the following description of the algorithm.

We denote the *observation* sequence $\{x_k\}$ corresponding to K transmitted code symbols by the vector \mathbf{x}_1^K . The minimum symbol-error probability detection algorithm is

$$\hat{a}_k = \arg \max_{\tilde{a}_k} \{p(\mathbf{x}_1^K | \tilde{a}_k) P\{\tilde{a}_k\}\} \quad (2)$$

in which \tilde{a}_k denotes a hypothetical information symbol,¹ $p(\mathbf{x}_1^K | \tilde{a}_k)$ is the conditional probability density function of

¹The superscript $\tilde{}$ is used throughout to denote a hypothetical value.

the observation and $P\{\tilde{a}_k\}$ is the *a priori* probability of the information symbol \tilde{a}_k . The quantities $p(\mathbf{x}_1^K | \tilde{a}_k) P\{\tilde{a}_k\}$ are proportional to the *a posteriori* probabilities $P\{\tilde{a}_k | \mathbf{x}_1^K\}$ of the information symbols and may be regarded as soft-decisions [15].

Let us denote by $\tilde{\mu}_k$ an encoder state and $\tilde{\delta}_k \triangleq (\tilde{a}_{k-1}, \tilde{a}_{k-2}, \dots, \tilde{a}_{k-N+1}, \tilde{\mu}_{k-N+1})$, where N is an integer, a state which also partially takes into account the channel phase memory. As in [13], [14], we refer to N as the *phase memory* parameter. Since symbols $(\tilde{c}_k, \tilde{c}_{k-1}, \dots, \tilde{c}_{k-N+1})$ are uniquely determined by the transition $(\tilde{a}_k, \tilde{\delta}_k)$, we may express the conditional probability density function of \mathbf{x}_{k-N+1}^k , assuming the transition $(\tilde{a}_k, \tilde{\delta}_k)$ and a given value θ of the channel phase, as

$$\begin{aligned} p(\mathbf{x}_{k-N+1}^k | \tilde{a}_k, \tilde{\delta}_k, \tilde{\theta}) \\ = \frac{1}{(2\pi\sigma^2)^N} \exp\left\{-\frac{1}{2\sigma^2} \sum_{i=0}^{N-1} |x_{k-i} - \tilde{c}_{k-i} e^{j\tilde{\theta}}|^2\right\} \end{aligned} \quad (3)$$

and therefore (see also [2, eq. (9)])

$$\begin{aligned} \gamma_k(\tilde{a}_k, \tilde{\delta}_k) \\ \triangleq p(\mathbf{x}_{k-N+1}^k | \tilde{a}_k, \tilde{\delta}_k) \\ = \int_0^{2\pi} p(\mathbf{x}_{k-N+1}^k | \tilde{a}_k, \tilde{\delta}_k, \tilde{\theta}) p(\tilde{\theta}) d\tilde{\theta} \\ = \frac{1}{(2\pi\sigma^2)^N} \exp\left\{-\frac{1}{2\sigma^2} \sum_{i=0}^{N-1} [|x_{k-i}|^2 + |\tilde{c}_{k-i}|^2]\right\} \\ \cdot \mathbf{I}_0\left(\frac{1}{\sigma^2} \left| \sum_{i=0}^{N-1} x_{k-i} \tilde{c}_{k-i}^* \right|\right) \end{aligned} \quad (4)$$

where $\mathbf{I}_0(x)$ is the zeroth-order modified Bessel function of the first kind.

The probability density function $p(\mathbf{x}_1^K | \tilde{a}_k)$ which appears in (2) may be evaluated as

$$\begin{aligned} p(\mathbf{x}_1^K | \tilde{a}_k) \\ = p(\mathbf{x}_1^{k-N}, \mathbf{x}_{k-N+1}^k, \mathbf{x}_{k+1}^K | \tilde{a}_k) \\ = \sum_{\tilde{\delta}_k} p(\mathbf{x}_1^{k-N}, \mathbf{x}_{k-N+1}^k, \mathbf{x}_{k+1}^K | \tilde{a}_k, \tilde{\delta}_k) P\{\tilde{\delta}_k\} \\ = \sum_{\tilde{\delta}_k} p(\mathbf{x}_{k+1}^K | \mathbf{x}_{k-N+1}^k, \mathbf{x}_1^{k-N}, \tilde{a}_k, \tilde{\delta}_k) \\ \cdot p(\mathbf{x}_1^{k-N} | \mathbf{x}_{k-N+1}^k, \tilde{a}_k, \tilde{\delta}_k) \gamma_k(\tilde{a}_k, \tilde{\delta}_k) P\{\tilde{\delta}_k\} \end{aligned} \quad (5)$$

having exploited the independence of $\tilde{\delta}_k$ and \tilde{a}_k . Neglecting the information on the channel phase in the discrete-time interval $[(k+1), K]$ obtainable from the knowledge of \mathbf{x}_1^{k-N} , we have

$$\begin{aligned} p(\mathbf{x}_1^K | \tilde{a}_k) \simeq \sum_{\tilde{\delta}_k} p(\mathbf{x}_{k+1}^K | \mathbf{x}_{k-N+1}^k, \tilde{a}_k, \tilde{\delta}_k) \\ \cdot p(\mathbf{x}_1^{k-N} | \mathbf{x}_{k-N+1}^k, \tilde{a}_k, \tilde{\delta}_k) \gamma_k(\tilde{a}_k, \tilde{\delta}_k) P\{\tilde{\delta}_k\}. \end{aligned} \quad (6)$$

This approximation is useful in the presence of a time-varying channel phase.

Given state $\tilde{\delta}_{k+1}$, there exist M possible predecessor states $\tilde{\delta}_k$, each in a one-to-one correspondence with an information symbol. We denote this information symbol by \tilde{a}_{k-j} , where integer j depends on the code properties.² Therefore, the transition $(\tilde{a}_k, \tilde{\delta}_k)$ may also be determined by the couple $(\tilde{a}_{k-j}, \tilde{\delta}_{k+1})$, which we denote as $(\tilde{a}_{k-j}^-, \tilde{\delta}_{k+1}^+)$. Defining

$$\begin{aligned}\alpha_k(\tilde{a}_k, \tilde{\delta}_k) &\triangleq p\left(\mathbf{x}_1^{k-N} \middle| \mathbf{x}_{k-N+1}^k, \tilde{a}_k, \tilde{\delta}_k\right) \\ \beta_k(\tilde{a}_{k-j}^-, \tilde{\delta}_{k+1}^+) &\triangleq p\left(\mathbf{x}_{k+1}^k \middle| \mathbf{x}_{k-N+1}^k, \tilde{a}_{k-j}^-, \tilde{\delta}_{k+1}^+\right) \\ &= p\left(\mathbf{x}_{k+1}^k \middle| \mathbf{x}_{k-N+1}^k, \tilde{a}_{k-j}^-, \tilde{\delta}_{k+1}^+\right)\end{aligned}\quad (7)$$

the soft-decisions may be expressed as

$$\begin{aligned}p(\mathbf{x}_1^k | \tilde{a}_k) P\{\tilde{a}_k\} \\ \simeq \sum_{\tilde{\delta}_k} \alpha_k(\tilde{a}_k, \tilde{\delta}_k) \gamma_k(\tilde{a}_k, \tilde{\delta}_k) \beta_k(\tilde{a}_{k-j}^-, \tilde{\delta}_{k+1}^+) P\{\tilde{\delta}_k\} P\{\tilde{a}_k\},\end{aligned}\quad (8)$$

The probability density functions $\alpha_k(\tilde{a}_k, \tilde{\delta}_k)$ and $\beta_k(\tilde{a}_{k-j}^-, \tilde{\delta}_{k+1}^+)$ may be recursively computed by a *forward* and a *backward* recursion, respectively [30]. For $\alpha_k(\tilde{a}_k, \tilde{\delta}_k)$, denoting by $(\tilde{a}_{k-1}^+, \tilde{\delta}_{k-1}^-)$ the couple equivalent to $(\tilde{a}_{k-j-1}, \tilde{\delta}_k)$, we have

$$\begin{aligned}\alpha_k(\tilde{a}_k, \tilde{\delta}_k) &= p\left(\mathbf{x}_1^{k-N} \middle| \mathbf{x}_{k-N+1}^k, \tilde{a}_k, \tilde{\delta}_k\right) \\ &= \sum_{\tilde{a}_{k-j-1}} p\left(\mathbf{x}_1^{k-N} \middle| \mathbf{x}_{k-N+1}^k, \tilde{a}_{k-j-1}, \tilde{a}_k, \tilde{\delta}_k\right) \\ &\quad \cdot P\{\tilde{a}_{k-j-1} \mid \mathbf{x}_{k-N+1}^k, \tilde{a}_k, \tilde{\delta}_k\} \\ &= \sum_{\tilde{a}_{k-j-1}} p\left(\mathbf{x}_1^{k-N-1}, x_{k-N} \middle| \mathbf{x}_{k-N+1}^k, \tilde{a}_{k-j-1}, \tilde{a}_k, \tilde{\delta}_k\right) \\ &\quad \cdot P\{\tilde{a}_{k-j-1} \mid \mathbf{x}_{k-N+1}^k, \tilde{a}_k, \tilde{\delta}_k\} \\ &= \sum_{\tilde{a}_{k-j-1}} p\left(\mathbf{x}_1^{k-N-1} \middle| \mathbf{x}_{k-N}^k, \tilde{a}_{k-j-1}, \tilde{a}_k, \tilde{\delta}_k\right) \\ &\quad \cdot p(x_{k-N} \mid \mathbf{x}_{k-N+1}^k, \tilde{a}_{k-j-1}, \tilde{a}_k, \tilde{\delta}_k) \\ &\quad \cdot P\{\tilde{a}_{k-j-1} \mid \mathbf{x}_{k-N+1}^k, \tilde{a}_k, \tilde{\delta}_k\} \\ &\simeq \sum_{\tilde{a}_{k-j-1}} p\left(\mathbf{x}_1^{k-N-1} \middle| \mathbf{x}_{k-N}^{k-1}, \tilde{a}_{k-j-1}, \tilde{\delta}_k\right) \\ &\quad \cdot p(x_{k-N} \mid \mathbf{x}_{k-N+1}^k, \tilde{a}_{k-j-1}, \tilde{a}_k, \tilde{\delta}_k) P\{\tilde{a}_{k-j-1}\}\end{aligned}\quad (9)$$

where the dependence of $p(\mathbf{x}_1^{k-N-1} | \mathbf{x}_{k-N}^{k-1}, \tilde{a}_{k-j-1}, \tilde{a}_k, \tilde{\delta}_k)$ on x_k has been neglected and the fact that

²For a recursive code, $a_{k-j} = a_{k-N+1}$. For a non recursive code of constraint length ν and state defined as $\mu_k = (a_{k-1}, \dots, a_{k-\nu})$, we have $a_{k-j} = a_{k-N-\nu+1}$.

$P\{\tilde{a}_{k-j-1} | \mathbf{x}_{k-N+1}^k, \tilde{a}_k, \tilde{\delta}_k\} = P\{\tilde{a}_{k-j-1}\}$ has been used. Therefore, the forward recursion is

$$\begin{aligned}\alpha_k(\tilde{a}_k, \tilde{\delta}_k) &\simeq \sum_{\tilde{a}_{k-j-1}} \alpha_{k-1}(\tilde{a}_{k-1}^+, \tilde{\delta}_{k-1}^-) \\ &\quad \cdot p(x_{k-N} \mid \mathbf{x}_{k-N+1}^k, \tilde{a}_{k-j-1}, \tilde{a}_k, \tilde{\delta}_k) \\ &\quad \cdot P\{\tilde{a}_{k-j-1}\}\end{aligned}\quad (10)$$

where

$$\begin{aligned}p(x_{k-N} \mid \mathbf{x}_{k-N+1}^k, \tilde{a}_{k-j-1}, \tilde{a}_k, \tilde{\delta}_k) \\ = \frac{p(\mathbf{x}_{k-N}^k \mid \tilde{a}_{k-j-1}, \tilde{a}_k, \tilde{\delta}_k)}{p(\mathbf{x}_{k-N+1}^k \mid \tilde{a}_{k-j-1}, \tilde{a}_k, \tilde{\delta}_k)} \\ = \frac{\eta_k(\tilde{a}_k, \tilde{a}_{k-j-1}, \tilde{\delta}_k)}{\gamma_k(\tilde{a}_k, \tilde{\delta}_k)}\end{aligned}\quad (11)$$

and

$$\begin{aligned}\eta_k(\tilde{a}_k, \tilde{a}_{k-j-1}, \tilde{\delta}_k) \\ \triangleq p\left(\mathbf{x}_{k-N}^k \middle| \tilde{a}_{k-j-1}, \tilde{a}_k, \tilde{\delta}_k\right) \\ = \frac{1}{(2\pi\sigma^2)^{N+1}} \exp\left\{-\frac{1}{2\sigma^2} \sum_{i=0}^N [|x_{k-i}|^2 + |\tilde{c}_{k-i}|^2]\right\} \\ \cdot \mathbf{I}_0\left(\frac{1}{\sigma^2} \left| \sum_{i=0}^N x_{k-i} \tilde{c}_{k-i}^* \right| \right).\end{aligned}\quad (12)$$

The probability density function $\beta_k(\tilde{a}_{k-j}, \tilde{\delta}_{k+1})$ may be recursively computed in a similar fashion, according to

$$\begin{aligned}\beta_k(\tilde{a}_{k-j}, \tilde{\delta}_{k+1}) &\simeq \sum_{\tilde{a}_{k+1}} \beta_{k+1}(\tilde{a}_{k-j+1}^-, \tilde{\delta}_{k+2}^+) \\ &\quad \cdot \frac{\eta_{k+1}(\tilde{a}_{k+1}, \tilde{a}_{k-j}, \tilde{\delta}_{k+1})}{\gamma_k(\tilde{a}_k^+, \tilde{\delta}_k^-)} P\{\tilde{a}_{k+1}\}\end{aligned}\quad (13)$$

where, using the notation already introduced, $(\tilde{a}_{k-j+1}^-, \tilde{\delta}_{k+2}^+)$ is the couple equivalent to $(\tilde{a}_{k+1}, \tilde{\delta}_{k+1})$. As in the original BCJR algorithm, where $\alpha_k(\tilde{a}_k, \tilde{\delta}_k)$ and $\beta_k(\tilde{a}_{k-j}, \tilde{\delta}_{k+1})$ may be arbitrarily normalized.

Proper boundary conditions for the forward and the backward recursions may be defined in terms of initial and final states of the encoder [30]. In fact, if the receiver knows the initial encoder state and the first $N-1$ transmitted symbols, the initial state δ_N for the forward recursion is uniquely determined. Similarly, the final state of the encoder and the last $N-1$ transmitted symbols uniquely determine the final state δ_K of the receiver for the backward recursion.

By careful and tedious algebraic manipulations, it can be shown that the proposed algorithm reduces to the noncoherent algorithm in [24] and [26], in the special case of differentially encoded M -PSK. However, the algorithm given here is more general, being applicable to any coded linear modulation format. Furthermore, as shown in Section V-A, this algorithm is directly applicable to iterative decoding without an explicit use of differential encoding, whereas the algorithm in [24] and [26] is not.

It is worth noting that reduced-search algorithms may be used to decrease the overall complexity [36]. The so-called M-BCJR and T-BCJR algorithms may reduce the computational effort in this case, in which the increased number of states of the receiver with respect to that of the encoder is due to the phase memory. For the M-BCJR, the forward recursion on α_{k-1} , which produces α_k , is performed using only a fixed number of the largest components of α_{k-1} , whereas the others are set to zero. The same method may be adopted in the backward recursion. In the T-BCJR algorithm, all the components which fall below a given threshold are set to zero. A further complexity reduction may be obtained by operating in the logarithmic domain, in order to avoid common numerical problems and reduce the number of additions and multiplications [37].

Even using these methods of complexity reduction, the computational effort required for the implementation of the noncoherent BCJR-type algorithm is still heavy and exponentially grows with N . As a consequence, only small values of N may be adopted. For this reason, in the next section we propose an alternative algorithm based on NSD and SOVA. Although SOVA produces a simplified estimate of the *a posteriori* probabilities of the data symbols [31]–[33], the reduced complexity of this decoding scheme allows the use of larger values of N . As shown in the numerical results, in some applications the benefit of increasing the phase memory greatly compensates for the imperfect estimation of the *a posteriori* probabilities.

IV. SOFT-OUTPUT NSD

In [13] and [14], noncoherent algorithms for combined demodulation and decoding of coded modulations have been proposed, based on the VA with appropriate branch metrics, in which the tradeoff between performance and complexity may be controlled by means of the assumed phase memory parameter N and the level of state-complexity reduction. For increasing values of N , the performance of the optimal coherent detector may be approached. On the other hand, the state-complexity of these detection schemes may be limited by RSSD [34], [35]. This technique allows one to independently choose two parameters: phase memory N and number of trellis states S . Hence, the number of states may be limited without excessively reducing the value of N .

NSD schemes may be enhanced in order to deliver soft-outputs, by means of a SOVA [31]–[33]. The expression of the branch metrics may be found in [13] and will be rederived in the next section for the cases of interest [see (18)–(23)]. We use the soft-output Viterbi decoder architecture proposed in [33] in order to obtain a real-time scheme, whose complexity is roughly doubled with respect to that of a classical Viterbi decoder.

In the next sections, we present examples of the application of the proposed noncoherent BCJR-type algorithm and soft-output NSD (SO-NSD) in iterative decoding schemes.

V. NONCOHERENT ITERATIVE DECODING

A. Parallel Concatenated Codes

A turbo encoder is the parallel concatenation of two simple constituent encoders separated by a *nonuniform* interleaver [15].

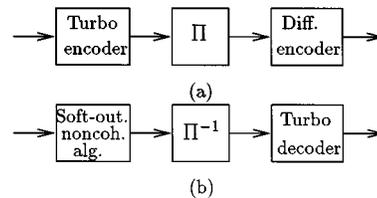


Fig. 1. Schemes with noncoherent predetection using the proposed soft-output noncoherent algorithms. (a) Transmitter. (b) Receiver.

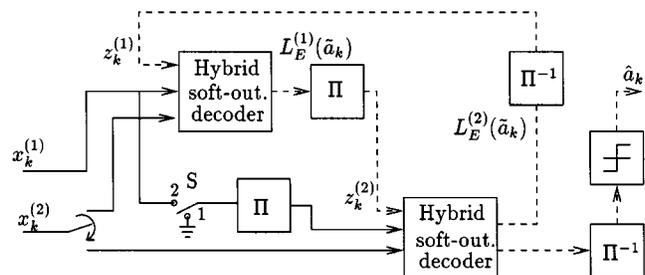


Fig. 2. Receiver with combined detection and decoding for a turbo code of rate 1/2.

The constituent codes are recursive systematic convolutional (RSC) codes and puncturing is used to achieve the desired code rate. The decoding process is based on a sub-optimal iterative algorithm in which each component decoder takes advantage of the *extrinsic* information produced by the other decoder at the previous step. This iterative decoding process is made possible by employing soft-output component decoders. Using the soft-output noncoherent algorithms introduced in the previous sections, two schemes for noncoherent decoding of turbo codes are considered.

In a first scheme, shown in Fig. 1, noncoherent detection is separate from the decoding of the turbo code. For this purpose, the code sequence is interleaved, by means of the block denoted by Π in Fig. 1(a), and differentially encoded. At the receiver, shown in Fig. 1(b), the detection is performed by means of one of the described soft-output noncoherent algorithms, designed to take into account differential encoding. The soft-output of this noncoherent block is then deinterleaved, by the block denoted by Π^{-1} , and sent to a standard turbo decoder which provides symbol decisions after the iterations necessary for the decoding process. These iterations do not involve the soft-output noncoherent predetection block. The interleaver is placed before the differential encoder in order to break up the dependence introduced by the turbo code. Differential encoding is necessary for noncoherent detection.

In a second scheme, combined noncoherent detection and turbo decoding is performed. In this case, it is not necessary to use differential encoding if the considered constituent RSC codes are noncoherently noncatastrophic [11], [13]. As an example, for a turbo code of rate 1/2 for the transmission of binary symbols $a_k \in \{\pm 1\}$ ($M = 2$) described in [15], the receiver is shown in Fig. 2, where $x_k^{(1)}$ and $x_k^{(2)}$ denote the systematic and redundant information, respectively. The role of switch S in the figure will be explained below. The overall structure is equal to that of the turbo decoder in [15], the only difference

being in the constituent decoders, which may employ the noncoherent BCJR-type algorithm or SO-NSD. These algorithms operate in a slightly different fashion with respect to those described in Sections III and IV. In fact, the described noncoherent BCJR-type algorithm and SO-NSD assume that the input signals are all noncoherent. In turbo decoding, however, each component decoder uses, in addition to the noncoherent channel output, a soft-output generated by the other decoder. Hence, the algorithms have to be modified in order to take into account this additional input. Unlike the channel output, the signal generated by each decoder is not affected by an unknown phase. For this reason, this signal may be viewed as “coherent.” In Fig. 2, these coherent signals are represented by dashed lines, whereas solid lines indicate channel (noncoherent) outputs.

We denote by z_k the input of a component decoder which has been generated by the other decoder. As in [15], each component decoder should calculate the logarithmic likelihood ratio of the information symbol \tilde{a}_k , i.e.,

$$\begin{aligned} L(\tilde{a}_k) &\triangleq \ln \frac{P\{\tilde{a}_k = 1 | \text{inputs}\}}{P\{\tilde{a}_k = -1 | \text{inputs}\}} \\ &= \ln \frac{p\{\text{inputs} | \tilde{a}_k = 1\} P\{\tilde{a}_k = 1\}}{p\{\text{inputs} | \tilde{a}_k = -1\} P\{\tilde{a}_k = -1\}} \end{aligned} \quad (14)$$

where the word “inputs” refers to both the noncoherent channel outputs and the soft-output produced by the other decoder. More precisely, the noncoherent BCJR-type algorithm computes the logarithmic likelihood ratios (14) on the basis of approximations of the *a posteriori* probabilities, as shown in Section III, whereas SO-NSD outputs a sequence whose elements are approximations of the logarithmic likelihood ratios (14), according to the SOVA principle.

The output of each component decoder is the extrinsic information extracted from $L(\tilde{a}_k)$. In fact, a fundamental principle of iterative decoding is not to feed a decoder with information that originates from itself. Moreover, this choice leads to an increased speed of convergence of the iterative process [15]. Since the extrinsic information at the output of each decoder may be considered as a “coherent signal,” each component decoder is *hybrid*, in the sense that it has both coherent (denoted by z_k) and noncoherent (denoted by x_k) inputs.

In the technical literature, there exist essentially two methods for the use of the extrinsic information z_k received by each decoder. In the original paper on turbo codes and iterative decoding [15], the input sequence $\{z_k\}$ extracted from the reliability values of the information sequence $\{a_k\}$, is modeled as Gaussian distributed. Specifically, it is assumed that $z_k = \eta_z a_k + n'_k$, where $\{n'_k\}$ are independent, zero-mean, real Gaussian random variables, with variance σ_z^2

and $\eta_z \triangleq |E\{z_k | a_k\}|$. In [15], it is noted that this Gaussian assumption, although not satisfied for the first few 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 and each iteration. An alternative method, which does not require an estimation of η_z and σ_z^2 , is proposed in [38] and [39]. In this *direct* method, 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 decoding step. A unified interpretation of these two methods is presented in [40], where it is shown that the second (direct) method is strictly equivalent to the first one based on the Gaussian model if $\eta_z/\sigma_z^2 = 1/2$. In the following, this Gaussian model is adopted for illustration purposes. However, the direct method is also considered in the numerical results.

For the proposed noncoherent BCJR-type algorithm, the generalization to the hybrid case is straightforward—the only modification consists of the term $\gamma_k(\tilde{a}_k, \tilde{\delta}_k)$ which has to take into account the extrinsic information as a further coherent input. As an example, let us consider the first decoder in Fig. 2, which has two noncoherent inputs $x_k^{(1)}$ and $x_k^{(2)}$ and one coherent input $z_k^{(1)}$. The logarithmic likelihood ratio of this decoder $L^{(1)}(\tilde{a}_k)$ may be easily expressed as follows [15]:

$$L^{(1)}(\tilde{a}_k) = \frac{2\eta_z z_k^{(1)}}{\sigma_z^2} + L_E^{(1)}(\tilde{a}_k) \quad (15)$$

where $L_E^{(1)}(\tilde{a}_k)$ is the extrinsic information of symbol \tilde{a}_k defined as shown in (16), at the bottom of the page, and

$$\begin{aligned} \gamma'_k(\tilde{a}_k, \tilde{\delta}_k) &\triangleq p\left(\mathbf{x}_{k-N+1}^{(1)k}, \mathbf{x}_{k-N+1}^{(2)k}, \mathbf{z}_{k-N+1}^{(1)k-1} \mid \tilde{a}_k, \tilde{\delta}_k\right) \\ &= \frac{1}{\sqrt{(2\pi\sigma_z^2)^{N-1}}} \\ &\quad \cdot \exp\left\{-\frac{1}{2\sigma_z^2} \sum_{i=1}^{N-1} \left(z_{k-i}^{(1)} - \tilde{a}_{k-i}\eta_z\right)^2\right\} \\ &\quad \cdot p\left(\mathbf{x}_{k-N+1}^{(1)k}, \mathbf{x}_{k-N+1}^{(2)k} \mid \tilde{a}_k, \tilde{\delta}_k\right) \end{aligned} \quad (17)$$

in which $\mathbf{x}_i^{(1)m} \triangleq \{x_k^{(1)}\}_{k=i}^m$ and $\mathbf{x}_i^{(2)m} \triangleq \{x_k^{(2)}\}_{k=i}^m$ are sequences of channel outputs and $\mathbf{z}_i^{(1)m} \triangleq \{z_k^{(1)}\}_{k=i}^m$ is the extrinsic information generated by the second decoder and fed back to the first one. In (16), the couple $(\tilde{a}_{k-j}^-, \tilde{\delta}_{k+1}^+)$ corresponds to the couple $(\tilde{a}_k = 1, \tilde{\delta}_k)$ in the numerator and the couple $(\tilde{a}_k = -1, \tilde{\delta}_k)$ in the denominator.

For SO-NSD, the decoders are still hybrid. We assume that the coherent input has a Gaussian probability density function

$$L_E^{(1)}(\tilde{a}_k) \triangleq \ln \frac{\sum_{\tilde{\delta}_k} \alpha_k(\tilde{a}_k = 1, \tilde{\delta}_k) \gamma'_k(\tilde{a}_k = 1, \tilde{\delta}_k) \beta_k(\tilde{a}_{k-j}^-, \tilde{\delta}_{k+1}^+) P\{\tilde{\delta}_k\} P\{\tilde{a}_k = 1\}}{\sum_{\tilde{\delta}_k} \alpha_k(\tilde{a}_k = -1, \tilde{\delta}_k) \gamma'_k(\tilde{a}_k = -1, \tilde{\delta}_k) \beta_k(\tilde{a}_{k-j}^-, \tilde{\delta}_{k+1}^+) P\{\tilde{\delta}_k\} P\{\tilde{a}_k = -1\}} \quad (16)$$

with mean η_z and variance σ_z^2 . Considering the first component decoder, the maximum-likelihood sequence detection strategy corresponds to the maximization of the following metric:³

$$\begin{aligned} \Lambda'_K(\tilde{\mathbf{a}}_1^K) &\triangleq \ln p(\mathbf{x}_1^{(1)K}, \mathbf{x}_1^{(2)K}, \mathbf{z}_1^{(1)K} | \tilde{\mathbf{a}}_1^K) \\ &\sim -\frac{1}{2\sigma^2} \sum_{k=1}^K [|x_k^{(1)}|^2 + |\tilde{c}_k^{(1)}|^2 + |x_k^{(2)}|^2 + |\tilde{c}_k^{(2)}|^2] \\ &\quad + \ln I_0 \left(\frac{1}{\sigma^2} \left| \sum_{k=1}^K [x_k^{(1)} \tilde{c}_k^{(1)*} + x_k^{(2)} \tilde{c}_k^{(2)*}] \right| \right) \\ &\quad - \frac{1}{2\sigma_z^2} \sum_{k=1}^K (z_k^{(1)} - \tilde{a}_k \eta_z)^2 \end{aligned} \quad (18)$$

where K is the number of transmitted information symbols, $\tilde{\mathbf{a}}_1^K \triangleq \{\tilde{a}_k\}_{k=1}^K$ is the sequence of information symbols, $\mathbf{x}_1^{(1)K} \triangleq \{x_k^{(1)}\}_{k=1}^K$ and $\mathbf{x}_1^{(2)K} \triangleq \{x_k^{(2)}\}_{k=1}^K$ are the sequences of channel outputs, corresponding to the encoded symbol sequences $\mathbf{c}_1^{(1)K} \triangleq \{c_k^{(1)}\}_{k=1}^K$ and $\mathbf{c}_1^{(2)K} \triangleq \{c_k^{(2)}\}_{k=1}^K$, received by the first decoder (according to the puncturing pattern), and $\mathbf{z}_1^{(1)K} \triangleq \{z_k^{(1)}\}_{k=1}^K$ is the extrinsic information generated by the second decoder and fed back to the first one.

Noting that in a turbo code the transmitted symbols belong to the alphabet $\{\pm 1\}$ and $\tilde{c}_k^{(1)} = \tilde{a}_k$, the metric to be maximized becomes

$$\Lambda_K(\tilde{\mathbf{a}}_1^K) \triangleq \frac{1}{\sigma^2} \left| \sum_{k=1}^K [x_k^{(1)} \tilde{a}_k + x_k^{(2)} \tilde{c}_k^{(2)}] \right| + \frac{\eta_z}{\sigma_z^2} \sum_{k=1}^K z_k^{(1)} \tilde{a}_k \quad (19)$$

where the approximation $\ln I_0(x) \simeq x$, valid for large x , has been used. Proceeding as in [13] and [14], we may define a partial sequence metric at the n th signaling interval

$$\Lambda_n(\tilde{\mathbf{a}}_1^K) \triangleq \frac{1}{\sigma^2} \left| \sum_{k=1}^n [x_k^{(1)} \tilde{a}_k + x_k^{(2)} \tilde{c}_k^{(2)}] \right| + \frac{\eta_z}{\sigma_z^2} \sum_{k=1}^n z_k^{(1)} \tilde{a}_k \quad (20)$$

and an incremental metric

$$\begin{aligned} \Delta_n(\tilde{\mathbf{a}}_1^K) &\triangleq \Lambda_n(\tilde{\mathbf{a}}_1^K) - \Lambda_{n-1}(\tilde{\mathbf{a}}_1^K) \\ &= \frac{1}{\sigma^2} \left| \sum_{k=1}^n [x_k^{(1)} \tilde{a}_k + x_k^{(2)} \tilde{c}_k^{(2)}] \right| \\ &\quad - \frac{1}{\sigma^2} \left| \sum_{k=1}^{n-1} [x_k^{(1)} \tilde{a}_k + x_k^{(2)} \tilde{c}_k^{(2)}] \right| + \frac{\eta_z}{\sigma_z^2} z_n^{(1)} \tilde{a}_n. \end{aligned} \quad (21)$$

By memory truncation [13], we obtain the following branch metric:

$$\begin{aligned} \lambda_n(\tilde{\mathbf{a}}_1^K) &= \frac{1}{\sigma^2} \left| \sum_{i=0}^{N-1} [x_{n-i}^{(1)} \tilde{a}_{n-i} + x_{n-i}^{(2)} \tilde{c}_{n-i}^{(2)}] \right| \\ &\quad - \frac{1}{\sigma^2} \left| \sum_{i=1}^{N-1} [x_{n-i}^{(1)} \tilde{a}_{n-i} + x_{n-i}^{(2)} \tilde{c}_{n-i}^{(2)}] \right| + \frac{\eta_z}{\sigma_z^2} z_n^{(1)} \tilde{a}_n. \end{aligned} \quad (22)$$

Employing SO-NSD for both component decoders, it is difficult to identify the extrinsic information embedded in the heuristic reliability value generated by each decoder. Based on the remarks in [41], it is logical to define the extrinsic information $L_E(\tilde{a}_k)$ as the difference between the generated soft-output value $L(\tilde{a}_k)$ and the coherent term present in $L(\tilde{a}_k)$, related to the same information symbol, according to

$$L_E(\tilde{a}_k) \triangleq L(\tilde{a}_k) - 2 \frac{\eta_z}{\sigma_z^2} z_k. \quad (23)$$

We now comment on the role of switch S in Fig. 2. Depending on its position, two schemes of noncoherent turbo decoding may be obtained. In the first one, referred to as *asymmetric*, the second soft-output decoder does not directly use the channel outputs $\{x_k^{(1)}\}$, suitably interleaved, related to the systematic bits $\{c_k^{(1)}\} = \{a_k\}$. This configuration may be obtained when the switch S is in position “1.” A second scheme, referred to as *symmetric*, may be obtained when the switch S is in position “2.” The symmetric and asymmetric schemes are not equivalent for both conceptual and practical reasons.

Except for channels with constant or slowly varying phase over each data block, where both symmetric or asymmetric schemes may be adopted, for a time-varying phase, only the asymmetric scheme exhibits good performance. In fact, interleaving the channel outputs $\{x_k^{(1)}\}$ in order to pass them to the second component decoder may increase the phase difference between successive symbols. Therefore, the performance of the symmetric scheme necessarily degrades. On the contrary, the asymmetric configuration is very robust with respect to a time-varying phase, as will be shown in the numerical results. For a channel phase constant over each data block, the symmetric scheme allows us to obtain a better performance because the second noncoherent component decoder operates with a double number of noncoherent channel outputs and hence, with a more refined implicit phase estimation⁴ [13], [14]. This second component decoder is the bottleneck in the asymmetric scheme and, for this reason, may require a larger value of parameter N and thus have larger complexity.

B. Serially Concatenated Codes

We now use the two proposed algorithms to realize an iterative noncoherent decoding of serially concatenated interleaved codes. More precisely, we consider two types of serial concatenation. In the first scheme, we consider a concatenated code

³The expression $x \sim y$ denotes that x and y are monotonically related quantities.

⁴If puncturing is used, the number of channel outputs used by the second decoder may be more than doubled.

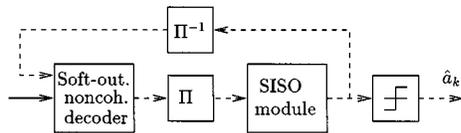


Fig. 3. Iterative decoding of serially concatenated interleaved codes.

given by a convolutional code followed by a nonuniform interleaver and a differential encoder [21]–[25]. The idea is that the interleaved convolutionally encoded symbols feed, as a new information sequence, the differential encoder. This allows the separation of the noncoherent detector, which takes into account the channel memory only, from the decoder of the convolutional code. As a second scheme, we consider the concatenation of two interleaved convolutional codes [16]. In fact, serially concatenated convolutional codes are emerging as an appealing alternative to the more classical turbo codes based on parallel concatenation.

The receiver is composed of two decoders concatenated by a deinterleaver as shown in Fig. 3. The inner decoder is noncoherent and utilizes the proposed algorithms. The output of this decoder represents the reliability value of each symbol. The outer coherent decoder uses a standard BCJR algorithm in order to obtain the best reliability information. Since the inner decoder operates on code symbols, the outer decoder has to provide the a posteriori probabilities of these code symbols in order to realize iterative decoding. For this reason, the outer decoder uses the soft-input soft-output (SISO) module described in [42].

The structure of the resulting iterative scheme is composed of an inner decoder which uses the noncoherent channel outputs and, from the second iteration ahead, the reliability values generated by the outer decoder. Based on the noncoherent BCJR-type algorithm or SO-NSD, it computes reliability values which approximate logarithmic likelihood ratios. The inner decoder passes the extrinsic information, which is extracted from these ratios, to the outer decoder. On the other hand, the outer decoder feeds back the extrinsic information on code symbols, which may be obtained as described in [42].

VI. NUMERICAL RESULTS

The performance of the proposed decoding algorithms is assessed by means of computer simulations in terms of bit-error rate (BER) versus E_b/N_0 , E_b being the received signal energy per information bit. Unless otherwise stated, the channel phase is assumed constant. In order to assess the performance of the proposed noncoherent BCJR-type algorithm, we initially consider combined detection and decoding of a single code (hence, no iterations are performed).

For differentially encoded binary PSK (BPSK), we consider a receiver using the noncoherent BCJR-type algorithm with $N = 3$. Fig. 4 shows the relevant performance and compares it with that of optimal coherent and differential detection. We may note that the performance of the proposed receiver is almost identical to that of the optimal coherent detector. As an example of coded BPSK, we consider the RSC code with 16 states, feedback and feedforward generators⁵ $G_1 = (37)_8$ and $G_2 = (21)_8$, respec-

⁵Octal representation.

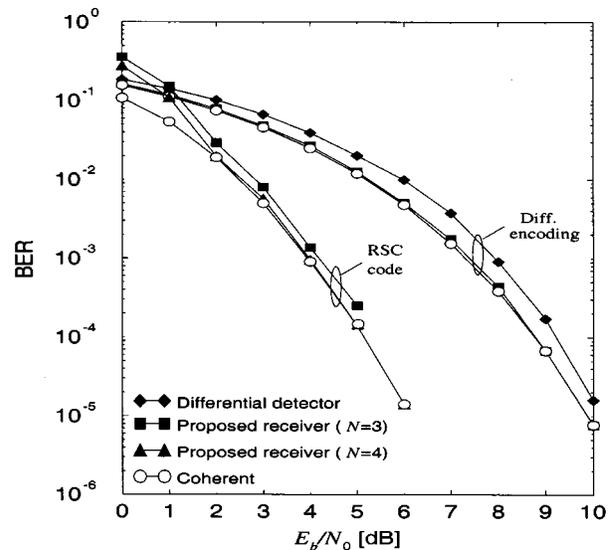


Fig. 4. BER of the proposed noncoherent BCJR-type algorithm for BPSK with differential encoding or a 16-state RSC code and comparison with a coherent receiver.

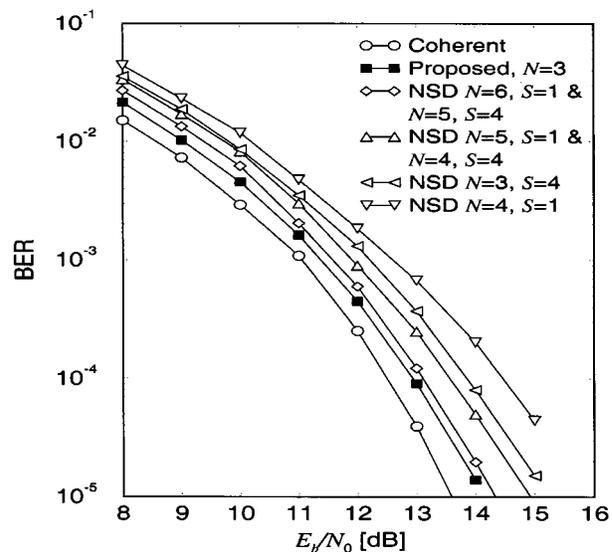


Fig. 5. BER of the proposed noncoherent BCJR-type algorithm (black marks) with $N = 3$ for differentially encoded 16-QAM and comparison with a coherent receiver and NSD schemes (white marks) of various complexity.

tively, and rate 2/3 obtained by means of puncturing, as used in the turbo code presented in [15]. In this case, the use of differential encoding is not necessary, since this code is noncoherently noncatastrophic. From Fig. 4, we observe that the proposed receiver based on the noncoherent BCJR-type algorithm with $N = 4$ exhibits a negligible performance loss with respect to coherent detection for values of E_b/N_0 greater than 2 dB.

The performance of the proposed receiver using the noncoherent BCJR-type algorithm with $N = 3$ is also assessed for 16-point quadrature amplitude modulation (16-QAM) employing quadrant differential encoding as described in [13]. In Fig. 5, it may be noted that the BER approaches that of coherent detection with a loss of only 0.5 dB. A comparison with NSD schemes of various complexity and a coherent receiver is also performed. The complexity of NSD is related

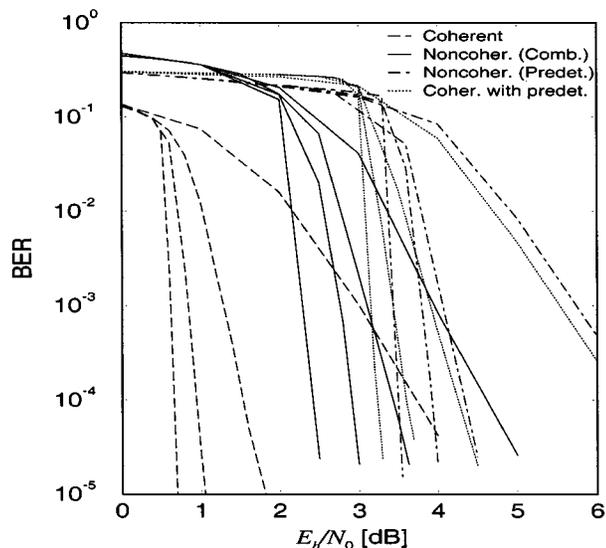


Fig. 6. BER of the proposed detection schemes using the noncoherent BCJR-type algorithm with predetection (dotted-dashed curves), combined detection and decoding (solid curves), coherent decoding (dashed curves) and coherent predetection (dotted curves). The numbers of iterations are 1, 3, 6 and 18, in all cases.

to the phase memory N and the number of trellis states S [13]. In the receivers with $S = 1$ the Viterbi processor degenerates into a symbol-by-symbol detector with decision-feedback. In the figure, curves labeled with two couples (N, S) show the similar performance of different receivers. We observe that the proposed noncoherent BCJR-type algorithm has very good performance.

In the above cases, the proposed noncoherent BCJR-type algorithm would likely not be used due to the higher complexity with respect to NSD schemes. This complexity may be justified when a soft-output is necessary, i.e., when the algorithm is employed in iterative decoding. Consider the turbo code of rate $1/2$, with 16-state RSC constituent codes with generators $G_1 = (37)_8$, $G_2 = (21)_8$, the 256×256 interleaver described in [15], and BPSK modulation. The RSC code is noncoherently noncatastrophic. Therefore, differential encoding is not used when combined noncoherent detection and turbo decoding is performed. The performance of the considered schemes using the noncoherent BCJR-type algorithm and SO-NSD is now addressed. The extrinsic information is modeled as a Gaussian random variable, as in Section V-A, or used to update the *a priori* probabilities (direct method), as described in [38]–[40]. In each case, the method giving best performance is presented. Unless otherwise stated, the Gaussian model is used.

Numerical results for the noncoherent BCJR-type algorithm are shown in Fig. 6 for various numbers of iterations (1, 3, 6, and 18, in all cases). A value of $N = 3$ is used for the noncoherent predetection block of Fig. 1 and each hybrid soft-output decoder of Fig. 2. The direct method is used in the combined scheme only. A performance loss of about 3 dB at a BER of 10^{-4} is exhibited by the predetection scheme in Fig. 1 with respect to coherent combined decoding. This predetection scheme exhibits a loss of only 0.25 dB with respect to a reference scheme which performs soft-output coherent predetection followed by turbo decoding. This reference

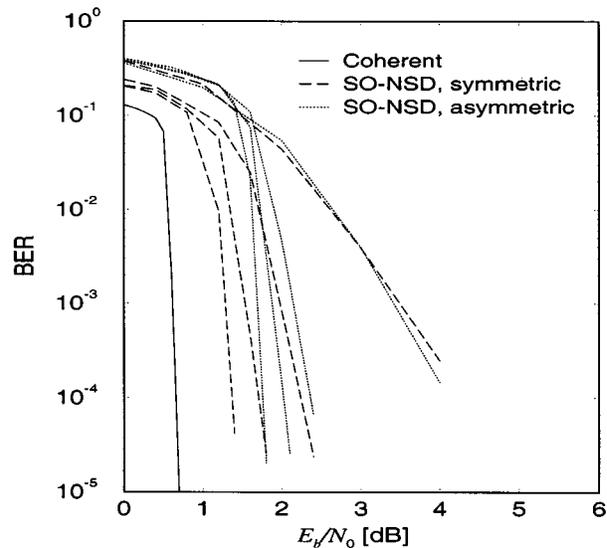


Fig. 7. BER of the proposed receivers using SO-NSD with combined detection and decoding for asymmetric (dotted curves) and symmetric (dashed curves) schemes. The numbers of iterations are 1, 3, 6, and 18 in both cases. The performance for coherent decoding (solid curve) and 18 iterations is also shown.

scheme is obtained by substituting the predetection block based on the noncoherent BCJR-type algorithm, with a coherent BCJR algorithm operating on the differential encoder trellis (see also [24], [43], and [44] for a similar decoding scheme). An improvement of about 1 dB is achieved using noncoherent combined detection and decoding according to Fig. 2 and the asymmetric scheme (switch S in position “1”). Considering the performance of the single component decoder in Fig. 4, the remaining performance loss of this combined scheme appears to be related to the E_b/N_0 threshold above which the single component decoders perform well.

The performance of SO-NSD with combined detection and decoding is shown in Fig. 7 for the same turbo code, using the direct method. Both asymmetric (switch in position “1”) and symmetric (switch in position “2”) schemes are considered. For the asymmetric scheme, the constituent decoders have $N = 9$ and $S = 128$, whereas in the symmetric scheme these values are $N = 9$ and $S = 64$. For comparison, the performance for coherent decoding and 18 iterations is also shown. At a BER of 10^{-4} , the performance loss with respect to coherent decoding is about 0.7 dB for the symmetric scheme and 1.1 dB for the asymmetric scheme. The specific choices of parameters N and S correspond to levels of complexity such that the computer simulation time is of the order of that necessary for the noncoherent BCJR-type algorithm with $N = 3$. A comparison of the proposed noncoherent BCJR-type and SO-NSD algorithms for equal values of N shows a superiority of the first scheme. Using SO-NSD, however, the performance may be greatly improved with respect to that of the noncoherent BCJR-type algorithm, due to the allowed larger values of N .

As mentioned in Section II, the assumption of a channel with a constant phase over each data block, used in the derivation of the algorithms, may be removed for the asymmetric scheme. Considering the presence of phase noise, modeled by a time-continuous Wiener phase process with

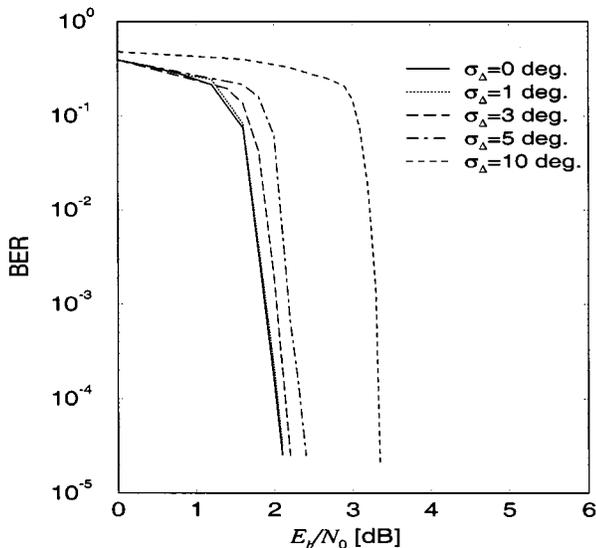


Fig. 8. BER of the proposed detection scheme using SO-NSD with combined detection and decoding for various levels of phase noise. The number of iterations is 6.

incremental variance over a signaling interval equal to σ_{Δ}^2 , we investigated the robustness of the proposed algorithms. In Fig. 8, the performance of an asymmetric scheme employing SO-NSD with constituent decoders characterized by $N = 9$ and $S = 128$ is shown. Various levels of phase noise with six decoder iterations are considered. As shown in the figure, for a phase noise standard deviation of 5° the performance loss is only 0.3 dB at a BER of about 10^{-4} . It is interesting to note that, for the considered range of BER values, the phase noise only affects the E_b/N_0 threshold above which convergence of the iterative process takes place—a larger phase noise has the effect of increasing this threshold. Incidentally, we note that the robustness of the proposed noncoherent algorithms to phase noise is higher for lower values of phase memory N . If a specific application requires a high robustness, the proposed BCJR-type algorithm may be a proper option.

Finally, we address the performance of the two proposed algorithms when used to iteratively decode serially concatenated codes. The extrinsic information at the input of the outer decoder is considered as an update of the a priori probability (direct method); hence, this outer decoder acts as an SISO module [42]. On the other hand, the extrinsic information at the input of the inner decoder is modeled as Gaussian. In the following, we present numerical results for the noncoherent BCJR-type algorithm. A similar performance may be obtained by means of SO-NSD.

In the first case, the overall code is composed of a concatenation of the 16-state RSC code with generators $G_1 = (37)_8$, $G_2 = (21)_8$ and rate 1/2, a 256×256 nonuniform interleaver [15], and a differential encoder. The performance of the noncoherent BCJR-type algorithm is shown in Fig. 9 for various numbers of iterations (1, 3, and 10). The component noncoherent decoder assumes $N = 4$. For comparison, the performance of optimal coherent decoding of the RSC code and iterative coherent decoding of the overall code is also shown. As we may observe, the performance of the noncoherent receiver is obviously

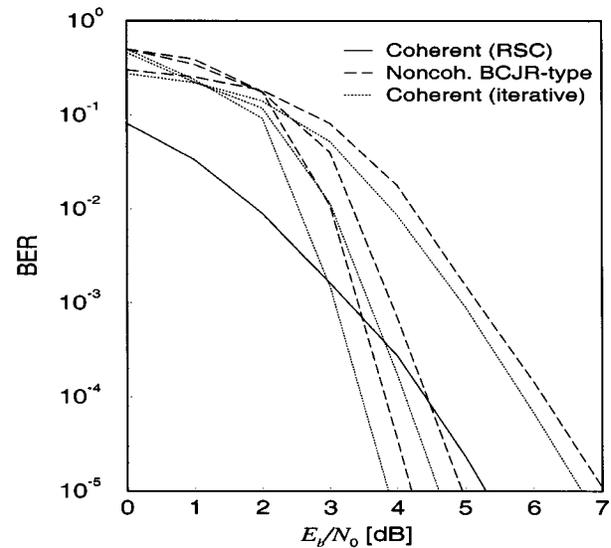


Fig. 9. BER of the proposed detection scheme using the noncoherent BCJR-type algorithm with $N = 4$ (dashed curves) for the serial concatenation of a convolutional code, an interleaver and a differential encoder. For comparison, the performance of iterative coherent decoding (dotted curves) and optimal coherent decoding of the single convolutional code (solid curve) is also shown. The numbers of iterations are 1, 3, and 10.

lower-bounded by that of the corresponding iterative coherent system, for an equal number of iterations. For ten iterations, the performance loss of the noncoherent receiver with respect to that of the coherent one is about 0.3 dB at a BER of 10^{-5} . At low BER, the iterative noncoherent system outperforms the optimal coherent detector for the single RSC code. This behavior, already observed in [22], [24], [25], [43], and [44], is related to the fact that the concatenation of a convolutional and a differential code, through interleaving, may generate a new code with better performance, even if the differential encoder has a unitary coding rate. This is due to the recursive nature of differential encoding. In fact, in a serial concatenation, an interleaver gain is possible only for a recursive inner code [16], [43], [44].

In the second case, we consider the concatenation of a 4-state nonrecursive nonsystematic convolutional code with generators $G_1 = (7)_8$, $G_2 = (5)_8$ and rate 1/2, and a four-state RSC code with generators $G_1 = (5)_8$, $G_2 = (7)_8$ and rate 1/2. The 256×256 nonuniform interleaver is used [15]. The performance for various numbers of iterations, assuming $N = 3$ and $N = 5$, is shown in Fig. 10 and compared with that of the corresponding coherent system. We again note that the performance of the noncoherent receiver at each iteration is lower-bounded by that of the corresponding iterative coherent system for the same number of iterations. For 18 iterations, the performance loss ($N = 5$) is about 0.7 dB at a BER of 10^{-5} .

The performance of the considered predetection scheme in Fig. 6 is now further investigated in view of the results in Fig. 9. As shown in Fig. 6, this receiver exhibits a loss of only 0.25 dB with respect to a reference scheme which performs soft-output coherent predetection followed by turbo decoding. This result, theoretically analyzed in [23], shows the intrinsic limit of schemes based on predetection and was not recognized in [20], where the performance loss was erroneously interpreted as due to the noncoherent approach. An improved reference receiver

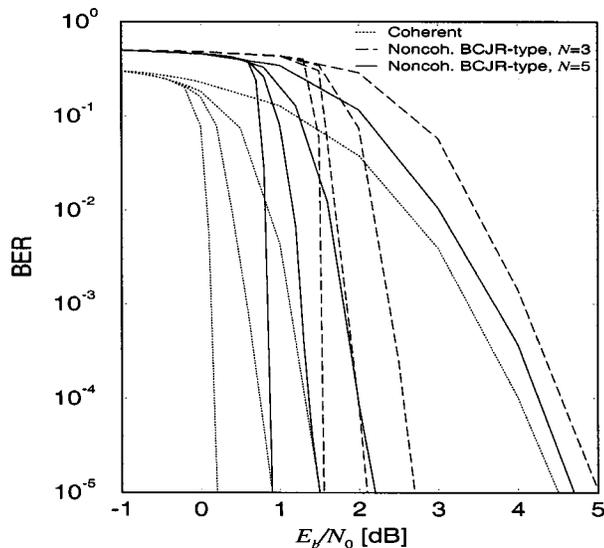


Fig. 10. BER of the proposed detection scheme using the noncoherent BCJR-type algorithm with $N = 3$ (dashed curves) and $N = 5$ (solid curves) for the serial concatenation of two interleaved convolutional codes. For comparison, the performance of iterative coherent decoding (dotted curves) is also shown. The numbers of iterations are 1, 3, 6, and 18.

was also considered in which the iteration process incorporates a coherent predetection block. This scheme, where extrinsic information is exchanged among the predetection block and the two component decoders, exhibits a further performance improvement of only 0.25 dB compared to the previously described reference coherent scheme and confirms the intrinsic limit of predetection schemes.

There are two possible alternative interpretations for this result. According to a first interpretation, differential encoding may render the overall concatenated code significantly worse than the original turbo code. In fact, by adding an interleaver and a differential encoder a new code is obtained, whose performance may be intuitively worse than that of the “practically optimal” turbo code described in [15]. This interpretation does not contradict the results obtained in Fig. 9 and [24], [43], and [44] for the concatenation of a differential encoder and a convolutional encoder, where the resulting code is better than the simple convolutional code. According to a second interpretation, the overall concatenated code may indeed be no worse⁶ than the original turbo code, but for this specific concatenation the iterative decoding process may fail. This second interpretation was suggested in [45].

VII. CONCLUSIONS

In this paper, two noncoherent soft-output detection algorithms have been proposed. The first one, the so-called noncoherent BCJR-type algorithm, is an approximation of a minimum symbol-error probability decoder under the assumption of a constant and uniformly distributed channel phase. The second proposed algorithm is a soft-output version of NSD, and generates a heuristic reliability information, based

⁶The overall code cannot be appreciably better than the original turbo code, whose performance is close to the theoretical limit.

on the principle of SOVA. The approximations involved in the derivation of the proposed detection algorithms make them effective in coping with time-varying phase models. These algorithms are well suited for iterative processing applications, in which the refined input symbol probability estimates are fed back to the demodulator.

Applications for combined detection and decoding of differential or convolutional codes have been considered and a comparison between the two algorithms has been performed. A performance very close to that of corresponding coherent schemes has been achieved in all the considered cases. Further applications to the noncoherent detection of turbo codes and serially concatenated interleaved codes have also been considered. Numerical results show that the proposed noncoherent turbo decoding schemes exhibit a moderate performance loss with respect to the corresponding coherent schemes. A specific investigation has shown a significant robustness of these noncoherent schemes to phase noise.

REFERENCES

- [1] S. G. Wilson, J. Freebersyser, and C. Marshall, “Multi-symbol detection of M -DPSK,” *Proc. Global Commun. Conf. (GLOBECOM’89)*, pp. 1692–1697, Nov. 1989.
- [2] D. Divsalar and M. K. Simon, “Multiple-symbol differential detection of MPSK,” *IEEE Trans. Commun.*, vol. 38, pp. 300–308, Mar. 1990.
- [3] H. Leib and S. Pasupathy, “Noncoherent block demodulation of PSK,” in *Proc. Veh. Technol. Conf. (VTC’90)*, May 1990, pp. 407–411.
- [4] D. Divsalar, M. K. Simon, and M. Shahshahani, “The performance of trellis-coded MDPSK with multiple symbol detection,” *IEEE Trans. Commun.*, vol. 38, pp. 1391–1403, Sept. 1990.
- [5] D. Divsalar and M. K. Simon, “Maximum-likelihood differential detection of uncoded and trellis coded amplitude phase modulation over AWGN and fading channels metrics and performance,” *IEEE Trans. Commun.*, vol. 42, pp. 76–89, Jan. 1994.
- [6] —, “Multiple symbol partially coherent detection of MPSK,” *IEEE Trans. Commun.*, vol. 42, pp. 430–439, Feb./Mar./Apr. 1994.
- [7] Y. Kofman, E. Zehavi, and S. Shamai, “ n d -convolutional codes—Part I: Performance analysis,” *IEEE Trans. Inform. Theory*, vol. 43, pp. 558–575, Mar. 1997.
- [8] Y. Kofman, E. Zehavi, and S. Shamai, “ n d -convolutional codes—Part II: Structural analysis,” *IEEE Trans. Inform. Theory*, vol. 43, pp. 576–589, Mar. 1997.
- [9] D. Makrakis and P. T. Mathiopoulos, “Trellis coded noncoherent QAM: A new bandwidth and power efficient scheme,” in *Proc. Veh. Technol. Conf. (VTC’89)*, May 1989, pp. 95–100.
- [10] D. Makrakis and K. Feher, “Optimal noncoherent detection of PSK signals,” *Electron. Lett.*, vol. 26, pp. 398–400, Mar. 15, 1990.
- [11] D. Raphaeli, “Noncoherent coded modulation,” *IEEE Trans. Commun.*, vol. 44, pp. 172–183, Feb. 1996.
- [12] F. Adachi, “Reduced state transition Viterbi differential detection of M -ary DPSK signals,” *Electron. Lett.*, vol. 32, pp. 1064–1066, June 6, 1996.
- [13] G. Colavolpe and R. Raheli, “Noncoherent sequence detection,” *IEEE Trans. Commun.*, vol. 47, pp. 1376–1385, Sept. 1999.
- [14] —, “Noncoherent sequence detection of continuous phase modulations,” *IEEE Trans. Commun.*, vol. 47, pp. 1303–1307, Sept. 1999.
- [15] C. Berrou and A. Glavieux, “Near optimum error correcting coding and decoding: Turbo-codes,” *IEEE Trans. Commun.*, vol. 44, pp. 1261–1271, Oct. 1996.
- [16] 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.
- [17] L. Lu and S. G. Wilson, “Synchronization of turbo coded modulation systems at low SNR,” in *Proc. IEEE Int. Conf. Commun. (ICC’98)*, Atlanta, GA, June 1998, pp. 428–432.
- [18] I. Bar-David and A. Elia, “Augmented APP (A^2P^2) Module for a posteriori probability calculation and channel parameter tracking,” *IEEE Commun. Lett.*, vol. 3, pp. 18–20, Jan. 1999.

- [19] A. Anastopoulos and K. M. Chugg, "Adaptive iterative detection for turbo codes with carrier phase uncertainty," in *Proc. IEEE Global Commun. Conf. (GLOBECOM'99)*, Rio de Janeiro, Brazil, Dec. 1999, pp. 2369–2374.
- [20] E. K. Hall and S. G. Wilson, "Turbo codes for noncoherent channels," in *Proc. 6th IEEE Commun. Theory Mini-Conf. (CTMC'97)*, Phoenix, AZ, Nov. 1997, pp. 66–70.
- [21] M. Peleg and S. Shamai (Shitz), "Iterative decoding of coded and interleaved noncoherent multiple symbol detected DPSK," *Electron. Lett.*, vol. 33, no. 12, pp. 1018–1020, June 5, 1997.
- [22] M. Peleg, S. Shamai (Shitz), and S. Galán, "On iterative decoding for coded noncoherent MPSK communications over block-noncoherent AWGN channel," in *Proc. IEEE Int. Conf. Telecommun. (ICT'98)*, Chalkidiki, Greece, June 1998, pp. 109–114.
- [23] M. Peleg and S. Shamai (Shitz), "On coded and interleaved noncoherent multiple symbol detected MPSK," *Eur. Trans. Telecommun. (ETT)*, vol. 10, no. 1, pp. 65–73, Jan./Feb. 1999.
- [24] I. D. Marsland and P. T. Mathiopoulos, "On the performance of iterative noncoherent detection of coded M -PSK signals," *IEEE Trans. Commun.*, vol. 48, pp. 588–596, Apr. 2000.
- [25] P. Hoher and J. Lodge, "'Turbo-DPSK': Iterative differential PSK demodulation and channel decoding," *IEEE Trans. Commun.*, vol. 47, pp. 837–843, June 1999.
- [26] X. Wu, C. Ling, and S. Sun, "Iterative joint demodulation and decoding of turbo codes over an AWGN channel with unknown phase," *IEEE Trans. Commun.*, submitted for publication.
- [27] S. Galán, M. Peleg, and S. Shamai (Shitz), "On iterative phase trellis based noncoherent detection of coded MPSK in a noisy phase regime," in *Proc. MELECON*, Tel-Aviv, Israel, May 1998, pp. 834–838.
- [28] M. Peleg and S. Shamai (Shitz), "On the capacity of the block-wise incoherent MPSK channel," *IEEE Trans. Commun.*, vol. 46, pp. 603–609, May 1998.
- [29] G. Colavolpe and R. Raheli, "The capacity of noncoherent channels," *Eur. Trans. Telecommun. (ETT)*, to be published.
- [30] 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, p. 284, March 1974.
- [31] G. Battail, "Pondération des symboles décodé par l'algorithme de Viterbi," *Ann. Télécommun.*, vol. 42, no. 1–2, pp. 31–38, January 1987.
- [32] J. Hagenauer and P. Hoher, "A Viterbi algorithm with soft-decision outputs and its applications," *Proc. IEEE Global Commun. Conf. (GLOBECOM'89)*, pp. 1680–1686, Nov. 1989.
- [33] C. Berrou, P. Adde, E. Angui, and S. Faudeil, "A low complexity soft-output Viterbi decoder architecture," *Proc. IEEE Int. Conf. on Commun. (ICC'93)*, pp. 737–740, June 1993.
- [34] M. V. Eyuboğlu and S. U. H. Qureshi, "Reduced-state sequence estimation with set partitioning and decision feedback," *IEEE Trans. Commun.*, vol. 36, pp. 13–20, Jan. 1988.
- [35] P. R. Chevillat and E. Eleftheriou, "Decoding of trellis-encoded signals in the presence of intersymbol interference and noise," *IEEE Trans. Commun.*, vol. 37, pp. 669–676, July 1989.
- [36] V. Franz and J. B. Anderson, "Concatenated decoding with a reduced-search BCJR algorithm," *IEEE J. Select. Areas Commun.*, vol. 16, pp. 186–195, Feb. 1998.
- [37] P. Robertson, E. Villerbrun, and P. Hoher, "Optimal and sub-optimal maximum a posteriori algorithms suitable for turbo decoding," *Euro. Trans. Telecommun. (ETT)*, vol. 8, no. 2, pp. 119–125, Mar.–Apr. 1997.
- [38] 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.
- [39] C. Douillard, M. Jezequel, C. Berrou, A. Picart, P. Didier, and A. Glavieux, "Iterative correction of intersymbol interference: Turbo-equalization," *Euro. Trans. Telecommun. (ETT)*, vol. 6, pp. 507–511, Sept./Oct. 1995.
- [40] G. Colavolpe, G. Ferrari, and R. Raheli, "Extrinsic information in turbo decoding: A unified view," in *Proc. IEEE Global Commun. Conf. (GLOBECOM'99)*, Rio de Janeiro, Brazil, Dec. 1999, pp. 505–509.

- [41] L. Papke and P. Robertson, "Improved decoding with the SOVA in a parallel concatenated (turbo-code) scheme," in *Proc. Int. Conf. Comm. (ICC'96)*, June 1996, pp. 102–106.
- [42] 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.
- [43] K. R. Narayanan and G. L. Stüber, "A serial concatenation approach in iterative demodulation and decoding," *IEEE Trans. Commun.*, vol. 47, pp. 956–961, July 1999.
- [44] M. Peleg, I. Sason, S. Shamai (Shitz), and A. Elia, "On interleaved, differentially encoded convolutional codes," *IEEE Trans. Inform. Theory*, vol. 45, pp. 2572–2582, Nov. 1999.
- [45] T. M. Duman and M. Salehi, "Performance bounds for turbo-coded modulation systems," *IEEE Trans. Commun.*, vol. 47, pp. 511–521, Apr. 1999.



and signal processing.

Giulio Colavolpe (S'97) was born in Cosenza, Italy, in 1969. He received the Dr. Ing. degree (*cum laude*) in telecommunications engineering 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 at the University of Parma. Since November 1999, he has been a Research Professor at the University of Parma. His main research interests include digital transmission theory, channel coding,



Gianluigi Ferrari (S'99) was born in Parma, Italy, in 1974. He received the Dr. Ing. degree (*Laurea*) (*cum laude*) in electrical engineering from the University of Parma, Parma, Italy, in 1998. Since November 1998, he has been working toward the Ph.D. degree at the University of Parma.

His main research interests include digital transmission and detection theory, channel coding and iterative decoding techniques.

Mr. Ferrari is a CNIT member and an AEI young member.



Riccardo Raheli (M'87) was born in Alezio (Lecce), Italy, in 1957. He received the Dr. Ing. degree (*Laurea*) from the University of Pisa, Pisa, Italy, in 1983, the M.S. degree from the University of Massachusetts, Amherst, in 1986, and the Doctoral degree (*Perfezionamento*) from the Scuola Superiore di Studi Universitari e di Perfezionamento, Pisa, Italy, in 1987, all in electrical engineering.

From 1986 to 1988, he was a Project Engineer with Siemens Telecomunicazioni, Milan, Italy. From 1988 to 1991, he was a Research Professor at the Scuola Superiore di Studi Universitari e di Perfezionamento S. Anna, Pisa, Italy. In 1990, he was a Visiting Assistant Professor at the University of Southern California, Los Angeles. Since 1991, he has been with the University of Parma, Parma, Italy, where he is currently an Associate Professor of Telecommunications. His research activity has led to about 60 scientific publications in international journals and conference proceedings and a few industrial patents. His main research interests include digital transmission theory, sequence detection techniques, signal processing, and adaptive algorithms for communications.

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