Noncoherent Iterative Decoding of Spectrally Efficient Coded Modulations

Gianluigi Ferrari
Communication Sciences Institute
Department of EE - Systems
University of Southern California
Los Angeles, CA 90089-2565

Giulio Colavolpe and Riccardo Raheli
Università di Parma
Dipartimento di Ingegneria dell'Informazione
Parco Area delle Scienze 181A
43100 Parma - ITALY

Submitted: January 15, 2001
Revised: April 24, 2001

Abstract
In this paper, we consider possible solutions for noncoherent decoding of concatenated codes with spectrally efficient modulations. Two main classes of schemes are considered. A first class is obtained by concatenating parallel coding schemes with differential encoding. A second class considers serially concatenated coding structures and possible schemes derived from turbo trellis coded modulation (T-TCM), which do not employ differential encoding. In the first case, at the receiver side we consider separate detection and decoding, while in the second case we consider joint detection and decoding. The major problem connected with such an iterative decoding procedure is that taking into account an augmented channel memory leads to an intolerable trellis size, and hence to an impractical decoding complexity. Reduced-complexity techniques suited to iterative decoding become fundamental, and we consider a recently proposed state-reduction technique. This way, the performance of a coherent receiver is approached, by keeping the number of receiver states fixed.

This paper was presented in part at the *2nd International Symposium on Turbo Codes & Related Topics*, Brest, France, September 2000, and the *International Conference on Communications* (ICC'01), Helsinki, Finland, June 2001.

1 Introduction

Since their appearance, concatenated codes with iterative decoding [1, 2] have stimulated a great research interest because of their performance close to the Shannon limit. Due to the growing data flow in future communication systems, where concatenated codes could be used, it will be more and more important to achieve high bit-rate transmissions, i.e., merging large coding gains with spectral efficient modulations. Hence, a very promising research area is related to the combination of concatenated codes and iterative decoding [1, 2] with modulation schemes which allow bandwidth efficiency, such as trellis coded modulation (TCM) [3].

Possible combinations of concatenated codes and spectrally efficient modulations have been considered in the literature. They are usually referred to as turbo trellis coded modulation (T-TCM) schemes. The first scheme in the literature appeared in [4], where the output bits of a turbo code are mapped, after puncturing, to a phase shift keying (PSK) or quadrature amplitude modulation (QAM) constellation. Another example of "pragmatic approach" to spectrally efficient modulations for turbo coded systems has been proposed in [5]. In [6] an "ad-hoc" approach has been considered, by using Ungerboeck codes [3] as component codes and puncturing the modulated symbols. In [7] possible schemes to jointly optimize the parallel concatenated code and the mapping are proposed. In [8] a T-TCM scheme identical to that proposed in [6] is described and a suitable application of soft-output Viterbi algorithm (SOVA) to multilevel modulation is considered. The versatility of T-TCM schemes, besides the performance, is the main concern in [9].

All the proposed schemes [4]-[9] consider transmission over an additive white Gaussian noise (AWGN) channel. It becomes a difficult task to extend the proposed structures to channels having memory. Bandpass transmission channels can be modeled as noncoherent in the sense that the transmitted signal undergoes an unknown phase rotation. This static phase rotation is responsible for an unlimited memory, at least in principle. Recently, noncoherent iterative decoding schemes for concatenated codes have been proposed based on suboptimal soft-output decoding algorithms suited for noncoherent channels [10]. The interest in noncoherent decoding algorithms to be used in iterative processing arises because phase-tracking schemes may deliver an unreliable phase estimate or require use of pilot symbols to avoid tracking losses for very low values of signal-to-noise ratio typical of concatenated coding schemes. Furthermore, noncoherent schemes exhibit inherent robustness to phase and frequency instabilities such as those caused by phase noise and uncompensated time-varying frequency offsets in local oscillators and Doppler shifts in wireless channels. These problems become more critical with an increased constellation size, as in the schemes considered in this paper. Moreover, since there is no inherent performance degradation in noncoherent decoding, provided the channel phase is sufficiently stable [11], an extension of the schemes proposed in [10], where binary phase shift keying (BPSK) was considered, to structures with an increased spectral efficiency, is challenging and of interest.

In this paper, we propose possible solutions for noncoherent decoding of concatenated codes with spectrally efficient modulations. We consider two main classes of schemes. A first class is obtained by concatenating parallel coding schemes (turbo codes) with a differential code. In this case, at the receiver side we consider separate detection and decoding: a noncoherent differential detector is followed by a coherent turbo decoder. A second class is obtained by considering serially concatenated coding structures [2, 12] and parallelly concatenated coding schemes derived from the structures proposed in [7]. At the receiver side we consider joint detection and decoding for the component decoders which directly receive the channel outputs (the inner decoder for serially concatenated codes and both component decoders for parallel schemes). The basic noncoherent decoder uses the noncoherent soft-output algorithm proposed in [10], where a parameter N is related to the assumed phase memory. In order to achieve satisfactory decoding performance, N must be sufficiently large. Nonetheless, since the memory and computational requirements grow exponentially with N, it becomes essential to apply reduced-state techniques, such as those recently proposed in [13].

In Section 2, we extend the considered soft-output noncoherent decoding algorithm to M-ary modulations. In Section 3, we describe a suitable state reduction technique. In Section 4, we consider spectrally efficient schemes which employ separate detection and decoding at the receiver side, whereas in Section 5 we propose schemes which employ joint detection and decoding at the receiver side. Numerical results are presented in Section 6 and conclusions are drawn in Section 7.

2 Noncoherent Soft-Output Decoding

In this section, we extend the noncoherent soft-output algorithm proposed for binary modulations in [10] to M-ary modulations. The algorithm will be described in the special case of trellis coded modulation [3], where each information symbol is related to more than one bit and the output symbol is mapped to a multilevel complex symbol. For the formulation, we consider the case of a recursive trellis code [3, 14]. Generalizations to other codes, in particular differential encoding, are straightforward.

We assume that a sequence of independent $M-$ary information symbols $\{a_k\}$ undergoes trellis encoding. Each information symbol a_k corresponds to a group of $m = \log_2 M$ bits, i.e., $a_k = (a_k^{(1)}, \ldots, a_k^{(m)})$. These information bits are coded into m_o output bits, through a recursive encoding rule. The M_o-ary output symbol $(c_k^{(1)}, \ldots, c_k^{(m_o)})$, where $m_o = \log M_o$, is then mapped to a complex symbol c_k belonging to the considered constellation. For systematic binary Ungerboeck codes of rate $n/(n+1)$ [3], $M = 2^n$ and $M_o = 2^{n+1}$. However, considering puncturing of the systematic output bits [7], M_o may be less than 2^{n+1}. The sampled output $\{x_k\}$ of a filter matched to the transmitted pulse is a sufficient statistic for noncoherent decoding [15]. Each sample may be expressed as $x_k = c_k e^{j\theta} + n_k$, where $\{n_k\}$ are samples of a zero mean complex-valued white Gaussian noise process and θ is a random variable uniformly distributed in $(-\pi, \pi]$. We denote by $\mathbf{x}_1^K = \{x_k\}_{k=1}^K$ the entire sequence of received samples or *observations*, where K is the transmission length. Similarly, we denote by $\mathbf{a}_1^K = \{a_k\}_{k=1}^K$ and $\mathbf{c}_1^K = \{c_k\}_{k=1}^K$ the entire sequence of information and code symbols, respectively. We now extend the algorithm introduced in [10], relative to the case of a recursive systematic code (RSC), to a TCM code. Note that in this case we must substitute a single information bit with an M-ary information symbol a_k carrying $\log_2 M$ bits.

Denoting by μ_k an encoder state, the decoder state, which takes partially into account the channel memory, may be expressed as

$$S_k = (a_{k-1}, a_{k-2}, \ldots, a_{k-N+1}, \mu_{k-N+1}) \tag{1}$$

where N is an integer. To account for the possible presence of parallel transitions, it is convenient to identify a trellis branch e_k by its beginning state S_k and driving information symbol a_k. In fact, two states S_k and S_{k+1} could be connected by parallel transitions driven by different information symbols. We showed in [10] that a good approximation of the a posteriori probability (APP) of symbol a_k can be determined on the basis of the considered observations $\mathbf{x}_1^K$. Denoting this value by $P\{a_k|\mathbf{x}_1^K\}$, it may be written as

$$P\{a_k|\mathbf{x}_1^K\} \simeq P\{a_k\} \sum_{e_k:a(e_k)=a_k} \gamma_k(e_k)\alpha_k(e_k)\beta_k(e_k)P\{S^-(e_k)\} \tag{2}$$

in which $a(e_k)$ denotes the information symbol driving transition e_k and

$$\begin{aligned}
\gamma_k(e_k) &\triangleq p\left(\mathbf{x}_{k-N+1}^k \,|e_k\right) \\
&\propto \exp\left\{-\frac{1}{2\sigma^2}\sum_{i=0}^{N-1}\left[|x_{k-i}|^2 + |c_{k-i}|^2\right]\right\} \mathrm{I}_0\left(\frac{1}{\sigma^2}\left|\sum_{i=0}^{N-1} x_{k-i}c_{k-i}^*\right|\right) && (3)\\
\alpha_k(e_k) &\triangleq p\left(\mathbf{x}_1^{k-N}\,\middle|\mathbf{x}_{k-N+1}^k, e_k\right) && (4)\\
\beta_k(e_k) &\triangleq p\left(\mathbf{x}_{k+1}^K\,\middle|\mathbf{x}_{k-N+1}^k, e_k\right) && (5)
\end{aligned}$$

where $[\cdot]^*$ is the conjugate operator and $\propto$ denotes proportionality. The sum in (2) is extended over all transitions of epoch k driven by information symbol a_k. The probability density function $\gamma_k(e_k)$, relative to a particular trellis transition, depends on the coding structure. $P\{a_k\}$ and $P\{S^-(e_k)\}$ denote the a priori probabilities of information symbol a_k and state $S^-(e_k)$, respectively, where $S^-(e_k)$ denotes the beginning state of transition e_k. In the following we will consider the expression of probabilities and probability density functions in the natural or logarithmic domain depending on the specific case, with the implicit assumption that the two formulations are equivalent.

The performance of iterative decoding at low bit error rate (BER) can be improved considering bit-interleaving [16]. Hence, equation (2) has to be modified in order to provide the a posteriori probabilities of single bits. Assuming that the information bits are independent within each symbol[1], we can consider $P\{a_k\} = P\{a_k^{(1)}\} \cdots P\{a_k^{(m)}\} = \prod_{i=1}^{m} P\{a_k^{(i)}\}$. In the case of an iterative decoding process, where $P\{a_k^{(i)}\}$ are derived from input extrinsic information, this assumption is just an approximation. Equation (2) may be extended as follows

$$P\{a_k^{(i)}|\mathbf{x}_1^K\} \simeq P\{a_k^{(i)}\} \sum_{e_k : a(e_k)^{(i)} = a_k^{(i)}} \gamma_k(e_k)\alpha_k(e_k)\beta_k(e_k)P\{S^-(e_k)\}\prod_{l \neq i} P\{a_k^{(l)}\} \tag{6}$$

where $a(e_k)^{(i)}$ denotes the i-th bit of the information symbol driving transition e_k.

Similarly to the well-known algorithm by Bahl, Cocke, Jelinek, and Raviv (BCJR), the probability density functions $\alpha_k(e_k)$ and $\beta_k(e_k)$ can be approximately computed by means of forward and backward recursions [10]. For this reason, we refer to the considered noncoherent soft-output algorithm as noncoherent BCJR-type algorithm. Denoting by $S^+(e_k)$ the final state of transition e_k we may write

$$\alpha_k(e_k) \simeq \sum_{e_{k-1} : S^+(e_{k-1}) = S^-(e_k)} \psi_k(e_{k-1}, e_k)\alpha_{k-1}(e_{k-1})P\{\mathring{a}(e_{k-1})\} \tag{7}$$

$$\beta_k(e_k) \simeq \sum_{e_{k+1} : S^+(e_k) = S^-(e_{k+1})} \phi_{k+1}(e_k, e_{k+1})\beta_{k+1}(e_{k+1})P\{a(e_{k+1})\} \tag{8}$$

where

$$\psi_k(e_{k-1}, e_k) \;\; = \;\; p(x_{k-N} \left| \mathbf{x}_{k-N+1}^k, e_{k-1}, e_k \right.)$$

[1]This assumption is motivated by the presence of bit interleaving.

$$\propto \exp\left\{-\frac{|x_{k-N}|^2+|c_{k-N}|^2}{2\sigma^2}\right\}\frac{\mathrm{I}_0\left(\frac{1}{\sigma^2}\left|\sum_{i=0}^{N}x_{k-i}c_{k-i}^*\right|\right)}{\mathrm{I}_0\left(\frac{1}{\sigma^2}\left|\sum_{j=0}^{N-1}x_{k-j}c_{k-j}^*\right|\right)} \tag{9}$$

$$\phi_k(e_{k-1},e_k) = p(x_k\left|\mathbf{x}_{k-N}^{k-1},e_{k-1},e_k\right.)$$

$$\propto \exp\left\{-\frac{|x_k|^2+|c_k|^2}{2\sigma^2}\right\}\frac{\mathrm{I}_0\left(\frac{1}{\sigma^2}\left|\sum_{i=0}^{N}x_{k-i}c_{k-i}^*\right|\right)}{\mathrm{I}_0\left(\frac{1}{\sigma^2}\left|\sum_{j=1}^{N}x_{k-j}c_{k-j}^*\right|\right)} \tag{10}$$

and $\mathring{a}(e_{k-1})$ denotes the information symbol "lost" in the transition e_{k-1}, i.e., the oldest information symbol in the initial state $S^-(e_{k-1})$. The couple $(S^+(e_{k-1}),\mathring{a}(e_{k-1}))$ uniquely identifies $S^-(e_{k-1})$. With the present definition of state S_k and for a recursive code, $\mathring{a}(e_{k-1})=a_{k-N}$. In (7), the sum is extended over all the transitions of epoch $k-1$ that end in the initial state of branch e_k. The sum in (8), relative to the trellis section at epoch $k+1$, may be interpreted similarly. Proper boundary conditions have to be considered in order to correctly initialize the forward and backward recursions.

3 Reduced-State Algorithm

Assuming that there are ζ_e possible encoder states, the decoder states are $\zeta_d=\zeta_e M^{N-1}$. For example, if $\zeta_e=16$, $M=4$ and $N=5$, then $\zeta_d=4096$. In order to make noncoherent decoding with spectrally efficient modulations practical, a complexity reduction suitable to the proposed soft-output decoding algorithm is needed. We consider a recently proposed method which is an extension of reduced-state sequence detection (RSSD) [17]-[19] to BCJR-type algorithms [13]. The basic idea is reducing the number of states and building a "survivor map" during the forward recursion (run first) to be used in the backward recursion and in the calculation of a posteriori probabilities. By defining a reduced state as $s_k=(a_{k-1},\ldots\ldots,a_{k-Q+1},\mu_{k-Q+1})$, with $Q<N$, a transition ϵ_k in the reduced-state trellis is associated with the symbols $(c_{k-Q+1},\ldots,c_k)$. We showed in [13] that a survivor may be associated with each transition ϵ_k in the reduced-state trellis. We may define by $E_{k-m}^{(l)}(\epsilon_k)$ the sequence of l transitions reaching epoch $k-m$ along the survivor

of transition ϵ_k, i.e., $(\hat{\epsilon}_{k-m-l+1}, \ldots, \hat{\epsilon}_{k-m}) \equiv (\hat{\mu}_{k-m-l-Q+2}, \hat{a}_{k-m-l-Q+2}, \ldots, \hat{a}_{k-m-Q+1}) \equiv (\hat{c}_{k-m-l-Q+2}, \ldots, \hat{c}_{k-m-Q+1})$. The transitions $\hat{\epsilon}_{k-j}$, encoder state $\hat{\mu}_{k-m-l-Q+2}$, information symbols $\hat{a}_{k-j}$ and code symbols $\hat{c}_{k-j}$ in $E^{(l)}_{k-m}(\epsilon_k)$ are those associated with the path history of transition ϵ_k. Hence, the probability density function γ_k may be correctly computed, making use of the built survivor map, as

$$\gamma_k(E^{(N-Q)}_{k-1}(\epsilon_k), \epsilon_k) = p\left(\mathbf{x}^k_{k-N+1} \left| E^{(N-Q)}_{k-1}(\epsilon_k), \epsilon_k\right.\right) \propto \exp\left\{-\frac{1}{2\sigma^2}\sum_{i=0}^{Q-1}\left[|x_{k-i}|^2 + |c_{k-i}|^2\right]\right\}$$
$$\cdot \exp\left\{-\frac{1}{2\sigma^2}\sum_{j=Q}^{N-1}\left[|x_{k-j}|^2 + |\hat{c}_{k-j}|^2\right]\right\} \mathrm{I}_0\left(\frac{1}{\sigma^2}\left|\sum_{i=0}^{Q-1} x_{k-i}c^*_{k-i} + \sum_{j=Q}^{N-1} x_{k-j}\hat{c}^*_{k-j}\right|\right) . \quad (11)$$

In the reduced state trellis, in analogy with equation (2), we wish to approximate the a posteriori probability as

$$P\{a_k|\mathbf{x}^K_1\} \simeq \sum_{\epsilon_k : a(\epsilon_k)=a_k} \gamma_k((E^{(N-Q)}_{k-1}(\epsilon_k), \epsilon_k))\alpha_k(\epsilon_k)\beta_k(\epsilon_k)P\{s^-(\epsilon_k)\} \quad (12)$$

where the two quantities α_k and β_k, in the reduced-state case, are defined as follows

$$\alpha_k(\epsilon_k) \triangleq p\left(\mathbf{x}^{k-N}_1 \left| \mathbf{x}^k_{k-N+1}, \epsilon_k\right.\right) \quad (13)$$
$$\beta_k(\epsilon_k) \triangleq p\left(\mathbf{x}^K_{k+1} \left| \mathbf{x}^k_{k-N+1}, \epsilon_k\right.\right) . \quad (14)$$

For a recursive code we use the following approximation for the a priori probability of state s_k [10]: $P\{s_k\} \simeq \prod_{i=1}^{Q-1} P\{a_{k-i}\}$.

If $Q < N$, then $\alpha_k(\epsilon_k)$, as defined in (13) for the reduced-state case, is different from $\alpha_k(e_k)$ as defined in (4) for the full-state case. Similarly, $\beta_k(\epsilon_k) \neq \beta_k(e_k)$. However, recursions for the computation of α_k and β_k may be found in the reduced-state case as well. The survivor map is built during the forward recursion and employed in the backward recursion and to evaluate γ_k in (11). Referring to the original formulation proposed in [10], the extension of the previously introduced general recursions (7) and (8) to (13) and (14) is not immediate. We now show the mathematical derivation which leads to the forward recursion in the reduced-state trellis. More precisely, assuming the survivor map is known up to epoch $k-1$, we show how to extend it to epoch k.

The detailed mathematical derivation of the forward recursion in [10] for the full-state case cannot be applied in this case. In fact, considering in the reduced-state case the same approach followed in [10], we should compute α_k as follows

$$\begin{aligned}\alpha_k(\epsilon_k) &= p\left(\mathbf{x}_1^{k-N}\left|\mathbf{x}_{k-N+1}^k, \epsilon_k\right.\right)\\ &= \sum_{a_{k-Q}} p\left(\mathbf{x}_1^{k-N-1}\left|\mathbf{x}_{k-N}^k, a_{k-Q}, \epsilon_k\right.\right) p\left(x_{k-N}\left|\mathbf{x}_{k-N+1}^k, a_{k-Q}, \epsilon_k\right.\right)\\ &\quad \cdot P\left\{a_{k-Q}\left|\epsilon_k, \mathbf{x}_{k-N+1}^k\right.\right\}. \end{aligned} \tag{15}$$

Assuming $Q \leq N-1$ (state reduction), a_{k-Q} depends on $\mathbf{x}_{k-N+1}^k$. Hence $P\left\{a_{k-Q}\left|\epsilon_k, \mathbf{x}_{k-N+1}^k\right.\right\}$ $\neq P\left\{a_{k-Q}\right\}$, making it impossible to evaluate this probability. Another approach has to be considered. More precisely, we may express α_k as follows

$$\begin{aligned}\alpha_k(\epsilon_k) &= p\left(\mathbf{x}_1^{k-N}\left|\mathbf{x}_{k-N+1}^k, \epsilon_k\right.\right) = \frac{p\left(\mathbf{x}_1^k\left|\epsilon_k\right.\right)}{p\left(\mathbf{x}_{k-N+1}^k\left|\epsilon_k\right.\right)}\\ &= \frac{\sum_{a_{k-Q}} p\left(\mathbf{x}_1^k\left|a_{k-Q}, \epsilon_k\right.\right) P\{a_{k-Q}|\epsilon_k\}}{p\left(\mathbf{x}_{k-N+1}^k\left|\epsilon_k\right.\right)}\\ &= \frac{\sum_{a_{k-Q}} p\left(\mathbf{x}_1^{k-N-1}\left|\mathbf{x}_{k-N}^{k-1}, x_k, a_{k-Q}, \epsilon_k\right.\right) p\left(\mathbf{x}_{k-N}^k\left|a_{k-Q}, \epsilon_k\right.\right) P\{a_{k-Q}|\epsilon_k\}}{p\left(\mathbf{x}_{k-N+1}^k\left|\epsilon_k\right.\right)}. \end{aligned} \tag{16}$$

Since $P\{a_{k-Q}|\epsilon_k\} = P\{a_{k-Q}\}$, observing that ϵ_{k-1} is uniquely determined by (a_{k-Q}, ϵ_k) and using as in [10] the approximation

$$p\left(\mathbf{x}_1^{k-N-1}\left|\mathbf{x}_{k-N}^{k-1}, x_k, a_{k-Q}, \epsilon_k\right.\right) \simeq p\left(\mathbf{x}_1^{k-N-1}\left|\mathbf{x}_{k-N}^{k-1}, \epsilon_{k-1}\right.\right) \tag{17}$$

we obtain the following approximate forward recursion in the reduced-state trellis

$$\begin{aligned}\alpha_k(\epsilon_k) &\simeq \frac{\sum_{a_{k-Q}} p\left(\mathbf{x}_1^{k-N-1}\left|\mathbf{x}_{k-N}^{k-1}, \epsilon_{k-1}\right.\right) p\left(\mathbf{x}_{k-N}^k\left|a_{k-Q}, \epsilon_k\right.\right) P\{a_{k-Q}|\epsilon_k\}}{p\left(\mathbf{x}_{k-N+1}^k\left|\epsilon_k\right.\right)}\\ &= \frac{\sum_{a_{k-Q}} \alpha_{k-1}(\epsilon_{k-1}) p\left(\mathbf{x}_{k-N}^k\left|a_{k-Q}, \epsilon_k\right.\right) P\{a_{k-Q}\}}{p\left(\mathbf{x}_{k-N+1}^k\left|\epsilon_k\right.\right)} \end{aligned} \tag{18}$$

where $\alpha_{k-1}(\epsilon_{k-1}) = p\left(\mathbf{x}_1^{k-N-1}\left|\mathbf{x}_{k-N}^{k-1}, \epsilon_{k-1}\right.\right)$ in agreement with (13).

The problem in the computation of (18) is the evaluation of the two probability density functions $p\left(\mathbf{x}_{k-N}^k\left|a_{k-Q}, \epsilon_k\right.\right)$ and $p\left(\mathbf{x}_{k-N+1}^k\left|\epsilon_k\right.\right)$. In fact, since $Q < N$, each of the two probability density functions should be correctly computed by averaging over previous information symbols. Since at epoch k the survivor of each transition ϵ_{k-1} is known and since $(a_{k-Q},\epsilon_k)\equiv(\epsilon_{k-1},\epsilon_k)$, we replace $p(\mathbf{x}_{k-N}^{k}\,|a_{k-Q},\epsilon_k)=p(\mathbf{x}_{k-N}^{k}\,|\epsilon_{k-1},\epsilon_k)$ with the probability density function $p(\mathbf{x}_{k-N}^{k}\,\big|E_{k-2}^{(N-Q)}(\epsilon_{k-1}),\epsilon_{k-1},\epsilon_k)$, obtaining the following modified recursion

$$\alpha_k(\epsilon_k)=\frac{\sum_{a_{k-Q}}\alpha_{k-1}(\epsilon_{k-1})p(\mathbf{x}_{k-N}^{k}\,\big|E_{k-2}^{(N-Q)}(\epsilon_{k-1}),\epsilon_{k-1},\epsilon_k)P\{a_{k-Q}\}}{p\left(\mathbf{x}_{k-N+1}^{k}\,|\epsilon_k\right)}\,. \tag{19}$$

We now express the forward recursion (19) in the logarithmic domain as follows

$$\begin{aligned}\bar{\alpha}_k(\epsilon_k) &\stackrel{\triangle}{=} \ln\alpha_k(\epsilon_k)\\ &= \ln\Big\{\sum_{a_{k-Q}}\exp\Big[\bar{\alpha}_{k-1}(\epsilon_{k-1})+\ln p(\mathbf{x}_{k-N}^{k}\,\big|E_{k-2}^{(N-Q)}(\epsilon_{k-1}),\epsilon_{k-1},\epsilon_k)\\ &\quad+\ln P\{a_{k-Q}\}\Big]\Big\}-\ln p(\mathbf{x}_{k-N+1}^{k}\,|\epsilon_k)\end{aligned} \tag{20}$$

and using the "max-log" approximation [26] we obtain

$$\begin{aligned}\bar{\alpha}_k(\epsilon_k) &\simeq \max_{a_{k-Q}}\Big\{\bar{\alpha}_{k-1}(\epsilon_{k-1})+\ln p(\mathbf{x}_{k-N}^{k}\,\big|E_{k-2}^{(N-Q)}(\epsilon_{k-1}),\epsilon_{k-1},\epsilon_k)+\ln P\{a_{k-Q}\}\Big\}\\ &\quad-\ln p(\mathbf{x}_{k-N+1}^{k}\,|\epsilon_k).\end{aligned} \tag{21}$$

The choice of the survivor associated with ϵ_k may be based on this max operation, which can be correctly carried out since the quantities $\bar{\alpha}_{k-1}(\epsilon_{k-1})$ and $\ln P\{a_{k-Q}\}$ are known and $\ln p(\mathbf{x}_{k-N}^{k}\,\big|E_{k-2}^{(N-Q)}(\epsilon_{k-1}),\epsilon_{k-1},\epsilon_k)$ can be computed. The term $\ln p(\mathbf{x}_{k-N+1}^{k}\,|\epsilon_k)$ does not affect the max operation and, as a consequence, the survivor selection, but it affects the exact value of $\bar{\alpha}_k(\epsilon_k)$. We denote by ϵ_{k-1}^{max} the previous transition of the survivor of transition ϵ_k. Equivalently, the symbol a_{k-Q}^{max} may be considered. Once the transition ϵ_{k-1}^{max} has been associated with ϵ_k, we replace $\ln p(\mathbf{x}_{k-N+1}^{k}\,|\epsilon_k)$ with the following probability density function in the logarithmic domain

$$\begin{aligned}\ln p(\mathbf{x}_{k-N+1}^{k}\,\big|E_{k-2}^{(N-Q-1)}(\epsilon_{k-1}^{max}),\epsilon_{k-1}^{max},\epsilon_k) &\sim -\frac{1}{2\sigma^2}\sum_{i=0}^{Q-2}\Big[|x_{k-i}|^2+|c_{k-i}|^2\Big]\\ +\frac{1}{2\sigma^2}\sum_{j=Q-1}^{N-1}\Big[|x_{k-j}|^2+|\hat{c}_{k-j}|^2\Big]&+\ln \mathrm{I}_0\left(\frac{1}{\sigma^2}\left|\sum_{i=0}^{Q-2}x_{k-i}c_{k-i}^{*}+\sum_{j=Q-1}^{N-1}x_{k-j}\hat{c}_{k-j}^{*}\right|\right)\end{aligned} \tag{22}$$

where the expression $x \sim y$ denotes that x and y are monotonically related quantities. The resulting forward recursion finally assumes the following form

$$\begin{aligned}
\bar{\alpha}_k(\epsilon_k) &= \bar{\alpha}_{k-1}(\epsilon_{k-1}^{max}) + \ln p(\mathbf{x}_{k-N}^k \left| E_{k-2}^{(N-Q)}(\epsilon_{k-1}^{max}), \epsilon_{k-1}^{max}, \epsilon_k \right.) \\
&\quad - \ln p(\mathbf{x}_{k-N+1}^k \left| E_{k-2}^{(N-Q-1)}(\epsilon_{k-1}^{max}), \epsilon_{k-1}^{max}, \epsilon_k \right.) + \ln P\{a_{k-Q}^{max}\} \\
&\sim \bar{\alpha}_{k-1}(\epsilon_{k-1}^{max}) + \frac{|x_{k-N}|^2 + |\hat{c}_{k-N}|^2}{2\sigma^2} + \ln \mathrm{I}_0 \left(\frac{1}{\sigma^2} \left| \sum_{i=0}^{Q-2} x_{k-i} c_{k-i}^* + \sum_{j=Q-1}^{N} x_{k-j} \hat{c}_{k-j}^* \right| \right) \\
&\quad - \ln \mathrm{I}_0 \left(\frac{1}{\sigma^2} \left| \sum_{i=0}^{Q-2} x_{k-i} c_{k-i}^* + \sum_{j=Q-1}^{N-1} x_{k-j} \hat{c}_{k-j}^* \right| \right) + \ln P\{a_{k-Q}^{max}\}. \qquad (23)
\end{aligned}$$

The obtained forward recursion in the reduced-state trellis exhibits some analogy with the corresponding forward recursion in the full-state trellis [10]. This indirectly confirms the validity of the proposed intuitive approximations. The backward recursion can be similarly obtained with the further simplification that the survivor map is now already available because previously determined during the forward recursion. More precisely, remarking that (ϵ_k, a_k) uniquely identifies ϵ_{k+1}, the backward recursion may be written as follows

$$\begin{aligned}
\bar{\beta}_k(\epsilon_k) &= \max_{a_{k+1}} \left\{ \bar{\beta}_{k+1}(\epsilon_{k+1}) + \ln p(\mathbf{x}_{k-N+1}^{k+1} \left| E_{k-1}^{(N-Q)}(\epsilon_k), \epsilon_k, \epsilon_{k+1} \right.) + \ln P\{a_{k+1}\} \right\} \\
&\quad - \ln p(\mathbf{x}_{k-N+1}^k \left| E_{k-1}^{(N-Q)}(\epsilon_k), \epsilon_k \right.) \\
&= \bar{\beta}_{k+1}(\epsilon_{k+1}^{max}) + \ln p(\mathbf{x}_{k-N+1}^{k+1} \left| E_{k-1}^{(N-Q)}(\epsilon_k), \epsilon_k, \epsilon_{k+1}^{max} \right.) + \ln P\{a_{k+1}^{max}\} \\
&\quad - \ln p(\mathbf{x}_{k-N+1}^k \left| E_{k-1}^{(N-Q)}(\epsilon_k), \epsilon_k \right.) \\
&\sim \bar{\beta}_{k+1}(\epsilon_{k+1}^{max}) + \frac{|x_{k+1}|^2 + |c_{k+1}|^2}{2\sigma^2} + \ln \mathrm{I}_0 \left(\frac{1}{\sigma^2} \left| \sum_{i=0}^{Q-2} x_{k+1-i} c_{k+1-i}^* + \sum_{j=Q-1}^{N} x_{k+1-j} \hat{c}_{k+1-j}^* \right| \right) \\
&\quad - \ln \mathrm{I}_0 \left(\frac{1}{\sigma^2} \left| \sum_{i=1}^{Q-2} x_{k+1-i} c_{k+1-i}^* + \sum_{j=Q-1}^{N-1} x_{k+1-j} \hat{c}_{k+1-j}^* \right| \right) + \ln P\{a_{k+1}^{max}\}. \qquad (24)
\end{aligned}$$

A problem connected with trellis coded modulations (especially when the code is recursive and $M > 2$) is the initialization of the recursion in the reduced-state trellis. Even if this aspect may be neglected when considering continuous transmissions, it is very important in packet transmissions, since interleaving operates on the entire packet, and hence it is not allowed to discard the first decoded symbols. The survivor map is built during the forward recursion, but the survivors should be already available at the very first steps of this recursion. Hence, an initial transient period for the forward recursion may be considered, where a fictitious phase memory parameter is increased by 1 at each step to reach the final value N as detailed in Appendix A. A valid alternative is considering a sequence of N pilot symbols at the beginning of the transmission, in order to correctly initialize the forward recursion. The transmission efficiency is not appreciably reduced as the overhead is less than 1% with the packet lengths considered in the numerical results.

4 Separate Detection and Decoding

The first considered class of spectrally efficient schemes uses coding structures based on the concatenation of a T-TCM block followed by an inner differential encoder. At the receiver side, a noncoherent differential detector computes a posteriori bit probabilities which are passed to the following coherent turbo decoder as logarithmic likelihood ratios. The introduction of the inner differential encoding allows to obtain noncoherently non-catastrophic coding schemes [15, 20].

The scheme proposed in [4] is basically a systematic turbo code of rate 1/3 followed by a puncturer and a mapper. An immediate extension of this scheme to noncoherent decoding is shown in Fig. 1, where a sequence of independent bits $\{u_k\}$ undergoes systematic turbo encoding. The code bits $\{b_k\}$ at the output of the turbo encoder are punctured according to some puncturing pattern [7]. The systematic and code bits, after being serialized, are interleaved. After interleaving they are grouped into $m = \log_2 M$ bits and mapped into $M-$ary complex symbols, undergoing differential encoding. In all block diagrams describing the proposed schemes, we associate solid lines with binary symbols and dashed lines with complex symbols. Furthermore, for notational consistency with Section 2, we use the symbols a_k and c_k to denote the input and output symbols, respectively, of the component encoders which are noncoherently decoded according to the described algorithm. Note that the symbols a_k

are rendered independent by the interleaver, as required by the algorithm in Section 2.

A similar scheme derived from one of the structures proposed in [7] is shown in Fig. 2. This scheme is basically composed of two parallelly concatenated Ungerboeck codes, and puncturing on information bits is considered before mapping. In this figure, we consider a sequence of couples of information bits $(u_k^{(1)}, u_k^{(2)})$. Both encoders receive this sequence and generate two sequences of coded bits $(b_k^{(1)}, b_k^{(2)})$, but the systematic bits are punctured symmetrically in the two codes, as shown in Fig. 2. We simply consider differential encoding after mapping. Strictly speaking, symbols a_k are not independent as assumed in the derivation of the noncoherent decoding algorithm. However, we observed by simulation that breaking this dependence by means of an interleaver (both bit-wise before mapping or symbol-wise after mapping) does not yield substantial performance improvement. This behavior may be related to the implicit puncturing considered in the outer turbo code, which, in a certain sense, decorrelates the bits carried by a modulated symbol.

5 Joint Detection and Decoding

In this case, we consider coding structures which do not employ differential encoding. The proposed schemes perform noticeably well in the case of ideal coherent decoding, i.e., assuming perfectly known phase at the receiver side.

Serially concatenated codes [2] have been proven to have remarkable performance (even better than that of turbo codes) with very simple component codes. However, this performance is obtained at the expense of the spectral efficiency of the code. For example, with rate 1/2 inner and outer convolutional codes, the overall rate is 1/4. In order to increase the efficiency of the serial code, we consider an inner Ungerboeck code, as shown in Fig. 3. A similar structure was also considered in [12], where an outer Reed-Solomon code and an inner Ungerboeck code were used. Various combinations of serial codes are considered, where the outer convolutional code is a simple non-recursive code [2, 28], whereas the inner Ungerboeck code may be a recursive systematic code [14] or a non-recursive one [15]. It is worth noting that interleaving is bit-wise. In fact, the coded bits generated by the outer encoder are serialized and then interleaved. Fig. 3 refers to the case of an outer rate 1/2 code and inner rate 2/3 code. After interleaving, the bits feed the inner encoder in groups of two. The receiver is based on an inner noncoherent decoder of the inner Ungerboeck code, which gives a posteriori probabilities of the systematic bits of each modulated symbol (bits $a_k^{(1)}$ and $a_k^{(2)}$ in Fig. 3) by using the proposed reduced-state noncoherent algorithm. These soft-outputs are passed, as logarithmic likelihood ratios, to the outer coherent decoder, which acts as a soft-input soft-output module [21]. Obviously, the overall serial code is noncoherently non-catastrophic depending on the characteristics of the inner Ungerboeck code. Hence, particular care has to be taken in choosing this code as a noncoherently non-catastrophic code [15, 20].

Besides serially concatenated coding structures, it is interesting to explore the possibility of deriving parallelly concatenated coding structures suitable to combined noncoherent detection and decoding. The scheme proposed in [6], employing PSK as modulation format at the output of each encoder, cannot be used when considering a noncoherent decoding strategy. In fact, because of puncturing, the proposed BCJR-type noncoherent decoding algorithm fails, since the metrics (9) and (10) reduce to 1 every other time epoch. Hence, every other transition in the decoder trellis the forward and backward recursions cannot be correctly extended. This problem obviously affects the reduced-state version of the algorithm described in Section 3 as well. On the contrary, the scheme proposed in [7] may be directly employed for transmissions over noncoherent channels, provided that the punctured component Ungerboeck codes are noncoherently non-catastrophic. With respect to the scheme proposed in [7], the only proposed modification consists of considering a single bit interleaver between the two Ungerboeck codes, instead of considering a different bit interleaver for each bit stream, as shown in Fig. 2. The input bit streams are serialized in a single bit stream before being interleaved. The interleaved bit stream is then parallelized and undergoes trellis encoding. We noticed that using a single interleaver instead of separate interleavers for each bit stream improves the performance, at least at high signal-to-noise ratios [16]. This is intuitively related to the fact that low reliability values associated with the couple of bits embedded in the same symbol may be better spread over the whole bit sequence. Hence, the receiver has a structure similar to that of a turbo decoder, where each component decoder uses the reduced-state noncoherent soft-output decoding algorithm previously introduced. This scheme may be considered as a direct extension to spectrally efficient modulations of the noncoherent schemes proposed in [10] for binary modulations.

In Fig. 4, we consider, for simplicity, the case of a turbo trellis encoder where each of the component Ungerboeck encoders receives a sequence of couples of information bits $(a_k^{(1)}, a_k^{(2)})$ and generates a parity bit ($c_k^{(0)}$ in the upper encoder and $d_k^{(0)}$ in the lower encoder). Puncturing may be considered on one of the two information bits (symmetrically in the two encoders): in the upper encoder the systematic bit $c_k^{(1)} = a_k^{(1)}$ is transmitted, whereas in the lower encoder the bit $d_k^{(1)} = a_{i_k}^{(2)}$ is transmitted.[2] As shown in Fig. 4, after interleaving the two original bit streams have to be separated in order to consider proper puncturing on $a_{i_k}^{(2)}$. This is possible if the single interleaver is odd-odd, i.e., if it maps the bits stored in odd positions (bits $\{a_k^{(1)}\}$) in odd positions, so that they can be recovered after interleaving. In this case, the single odd-odd interleaver is equivalent to two separate interleavers. A QPSK symbol is generated at the output of each component encoder. The spectral efficiency in this case is 1 bit per channel use.

Although the above scheme with QPSK has remarkable performance with coherent decoding, i.e., with an AWGN channel, we observed that the performance noticeably degrades when considering noncoherent decoding, because of the catastrophicity of the code. This motivates the following modification. The spectral efficiency remains the same by eliminating puncturing, hence transmitting an 8-PSK symbol at the output of each component

[2]The time instant of the second encoded bit is denoted by i_k because of the presence of interleaving.

encoder. In this case, both systematic bits at the input of each encoder are mapped to the corresponding generated complex symbol (in Fig. 4 we indicate by dotted lines the supplementary connections which must be considered). This means adding redundancy, at the cost of decreasing the robustness of the modulation constellation. In the coherent case, the performance worsens, whereas in the noncoherent case it improves. Combining modulation and coding when dealing with a noncoherent channel cannot be carried out as in the case of an AWGN channel, because the noncoherent catastrophicity must be taken into account. Moreover, based on an exhaustive search using different constellation mappings, we noticed that the receiver performance in the noncoherent case does not seem to be appreciably influenced by the particular mapping rule (Gray, reordered, etc. [7]).

The last considered parallel scheme deserves some remarks about its noncoherent catastrophicity. By reducing the modulation constellation from 8-PSK to QPSK, the code properties, in terms of modulated output symbols, may change. Hence, a code may not be simultaneously noncoherently non-catastrophic with and without puncturing. An open problem, currently under study, is the design of a good code for such a trasmitter structure when considering puncturing and QPSK. An important aspect to be considered is the rotational invariance of the component codes, taking into account puncturing and mapping. The methods proposed in [22]-[24] may be considered. A relevant analysis concerning the rotational invariance of T-TCM schemes is addressed in [25].

6 Numerical Results

The performance of the receivers considered in Section 4 and Section 5 is assessed by means of computer simulations in terms of BER versus E_b/N_0, E_b being the received signal energy per information bit and N_0 the monolateral noise power spectral density. All the BCJR-type algorithms (noncoherent and coherent) considered in the proposed schemes apply the max-log approximation [26]. The generated extrinsic information is weighted by a coefficient as described in [27]. The value of this coefficient, obtained by trial and error, is about 0.3 in all schemes.

In Fig. 5 the performance in the case of the code shown in Fig. 1 is presented. The code is that proposed in [4], with internal random 32×32 interleaver. The component RSC codes have generators $G_1 = 37$ and $G_2 = 21$.[3] The turbo code has rate 1/2: every 2 information bits ($\tilde{m} = 2$) two code bits ($m - \tilde{m} = 2$) are retained, with the puncturing pattern considered in [4]. After random bit interleaving, groups of $m = 4$ bits are mapped into a 16-QAM symbol. It is important to observe that the particular chosen mapping (Gray, reordered, natural, etc.) does not seem to noticeably influence the performance of the noncoherent system. This may be due to the presence of bit interleaving followed by differential encoding. The spectral efficiency of this system is 2 bits per channel use. The inner noncoherent differential detector at the receiver side applies the reduced-state noncoherent decoding algorithm proposed in Section 2 by reducing the number of states to 16. The phase parameter N is set equal to 4 or 6. For comparison, we also show the performance of the equivalent coherent system (i.e., considering differential encoding after the turbo code). In all cases the iterations are carried out in the outer coherent turbo decoder, and the numbers of considered iterations are 1, 3 and 5. It is evident that there is a slight improvement in the performance of the noncoherent system by increasing N from 4 to 6, and the loss, with respect to the noncoherent decoding, is about 1 dB at BER below 10^{-4}.

In Fig. 6 we show the performance in the case of noncoherent decoding of the code proposed in Fig. 2. The component 16-state recursive Ungerboeck codes of the turbo code have generators $h_0 = 23$, $h_1 = 16$ and $h_2 = 27$ [7], and there are two different 32×32 random bit-interleavers. We consider a 16-QAM modulation format. The system has an efficiency of 2 bits per channel use. As for the previous scheme, in this case also we consider the inner

[3] In the case of binary codes, for example RSC codes, we refer to the generators of the code as $\{G_i\}$, following the octal notation in [1, 28]. When referring to Ungerboeck codes, we indicate the generators of the code as $\{h_i\}$, following the octal notation in [7].

noncoherent detector with the number of states reduced to 16 and phase parameter N equal to 4 or 6, respectively. For comparison, we also show the performance of the equivalent coherent system. The numbers of iterations are 1, 3 and 6 in all cases. The performance loss of the noncoherent system with $N = 6$ with respect to the coherent system is about 1 dB.

In Fig. 7, the performance in the case of the serial scheme shown in Fig. 3 is presented. The outer code is a non-recursive non-systematic convolutional code, with generators $G_1 = 7$ and $G_2 = 5$ and rate 1/2. The inner Ungerboeck code is recursive and systematic, with generators $h_0 = 23$, $h_1 = 16$ and $h_2 = 27$ [7]. The inner interleaver is a 32×32 pseudorandom bit-interleaver. The bits at the output of the inner code are mapped to an 8-PSK symbol, considering reordered mapping [7]. The spectral efficiency of this system is 1 bit per channel use. The inner noncoherent decoder at the receiver side applies the reduced-state noncoherent decoding algorithm proposed in Section 2. Various complexity reduction levels, denoted by the couple (N, Q), are considered. The phase parameter N ranges from 4 to 16, while Q is kept fixed to 2 (64 states). For comparison, we also show the performance of the equivalent coherent system, i.e., assuming perfect knowledge of the channel phase at the receiver side. In all cases, the number of considered iterations is 10. As one can see, for increasing values of the phase parameter N the performance of the noncoherent scheme approaches that of the coherent scheme. For $N = 16$ the performance loss at a BER of 10^{-5} is around 1 dB.

In Fig. 8, we consider again a coding structure as given in Fig. 3, with the same inner Ungerboeck code of Fig. 7 but considering an outer non-recursive non-systematic convolutional code, with generators $G_1 = 15$ and $G_2 = 13$ and rate 1/2. Hence, we replaced an outer 8-state code with a 16-state code. As in the previous case, the noncoherent inner decoder is identified by the couple (N, Q). The phase parameter N ranges from 4 to 16, and $Q = 2$. The numbers of considered iterations for both the coherent and noncoherent systems are 10. For $N = 16$ the performance loss of the noncoherent scheme with respect to that of the coherent scheme is only 0.5 dB at a BER of 10^{-4}.

In Fig. 9, we consider a serial structure as in Fig. 3 given by an outer rate 1/3 non-recursive code with 16 states and generators $G_1 = 17$, $G_2 = 06$ and $G_3 = 15$ [28] and an inner rate 3/4 non-recursive code with 8 states and generators $G_1 = 040$, $G_2 = 402$, $G_3 = 240$ and $G_4 = 100$ [15]. The inner random interleaver is bit-wise, with length 1536. The spectral efficiency is 2 bits per channel and we consider a 16-QAM modulation format at the output of the inner code. The inner noncoherent decoder at the receiver side applies the reduced-state noncoherent decoding algorithm proposed in Section 2.[4] Various complexity reduction levels, denoted by the couple (N, Q), are considered. The numbers of iterations are 1, 5 and 10 in all cases, and a comparison with the equivalent coherent system is made.

In Fig. 10, we show the performance in the case of noncoherent decoding of the code proposed in Fig. 4. The component 16-state recursive Ungerboeck codes of the proposed scheme have generators $h_0 = 23$, $h_1 = 16$ and $h_2 = 27$ [7] and there is a single 64×64 pseudorandom bit-interleaver [1]. At the output of each component encoder both the systematic bits are retained and mapped, together with the parity bit, to an 8-PSK symbol. Reordered mapping is considered in this case as well. The system efficiency is 1 bit per channel use. The two component noncoherent decoders have a number of states reduced to 64 and phase parameter N equal to 4 and 6, respectively. For comparison, we also show the performance of the equivalent coherent system. The numbers of iterations are 1, 3 and 6 in all cases. Considering $N = 6$ and 6 decoding iterations, the performance loss of the noncoherent scheme with respect to the coherent scheme is about 1.5 dB.

We now compare the performance of the considered schemes under the same spectral efficiency. In fact, schemes with spectral efficiency of both 1 and 2 bits per channel use have been analyzed. As it appears from Figures 5, 6, and 9, for schemes with spectral efficiency of 2 bits per channel use, the coherent receivers show a BER of 10^{-4} at a signal to noise

[4] The derivation carried out in Section 2 in the case of a recursive code may be easily extended observing that in this case $\mathring{a}(e_{k-1}) = (a_{k-2}^{(1)}, a_{k-3}^{(2)}, a_{k-1}^{(3)})$. Hence, in this case the symbol $\mathring{a}(e_{k-1})$ is not an information symbol, but it is composed by bits coming from information symbols relative to different time instants.

ratio between 5 and 6 dB. The correspoding noncoherent schemes exhibit a performance degradation of about 1 dB. As shown in Figures 7, 8, and 10, for schemes with spectral efficiency of 1 bit per channel use, the performance of coherent receivers is between 3 and 4 dB, whereas the noncoherent schemes exhibit a performance loss of less than 1 dB. As one can see, the performance of each scheme is strictly related to its spectral efficiency and is roughly independent of the specific detection strategy (separate or joint). Taking into consideration the performance/complexity trade-off, it turns out that the simple schemes with separate detection and decoding may offer a good solution.

7 Conclusions

In this paper, we presented possible solutions for noncoherent decoding of concatenated codes with spectrally efficient modulations. We proposed a soft-output noncoherent decoding algorithm and showed that in the case of high order constellations it is essential to apply complexity reduction techniques in order to obtain implementable systems. A state-reduction technique suited to BCJR-type algorithms was successfully applied.

We considered a first class of schemes given by the concatenation of a parallel concatenated scheme with a differential encoder, and a second class constituted by serially concatenated schemes and a parallelly concatenated coding structure without differential encoding. In the first case we considered separate detection and decoding, and in the second case we considered joint detection and decoding. We demonstrated the performance for various values of phase memory parameter N, number of trellis states and length of transmitted bit packets. In all cases, the performance of the noncoherent scheme approaches that of the equivalent coherent scheme for increasing value of the parameter N.

The described separate and joint decoding schemes offer different levels of performance and complexity. The schemes based on separate detection and decoding have a low complexity, since the inner noncoherent detector accounts for differential encoding only. In particular, these schemes show a lower complexity with respect to the schemes where joint detection and decoding of trellis codes is considered. In terms of the performance/complexity trade-off, it turns out that the simple schemes with separate detection and decoding may offer a good solution in many situations.

Appendix A

In this appendix we show how the considered soft-output noncoherent algorithm presented in Section 3 has to be modified in the initial transient period, i.e., for $k \leq N$. The superscript $(\cdot)^t$ is used in the following to denote the value of previously introduced quantities during this initial transient period.

For $k < Q$, we may write[5]

$$P\{a_k|\mathbf{x}_1^K\} = P\{a_k\} \sum_{\epsilon_k^t : a(\epsilon_k^t) = a_k} \gamma_k^t(\epsilon_k^t)\beta_k^t(\epsilon_k^t) P\{s^{t-}(\epsilon_k^t)\} \tag{25}$$

where $\epsilon_k^t = s_{k+1}^t = (a_1, \ldots, a_k)$, $s^{t-}(\epsilon_k^t) = (a_1, \ldots, a_{k-1})$ and

$$\begin{aligned} \gamma_k^t(\epsilon_k^t) &\triangleq p\left(\mathbf{x}_1^k \middle| \epsilon_k^t\right) \\ &= \exp\left\{-\frac{1}{2\sigma^2}\sum_{i=0}^{k-1}\left[|x_{k-i}|^2 + |c_{k-i}|^2\right]\right\} \mathrm{I}_0\left(\frac{1}{\sigma^2}\left|\sum_{i=0}^{k-1} x_{k-i}c_{k-i}^*\right|\right). \end{aligned} \tag{26}$$

The probability density function β_k^t may be computed by means of a simplified backward recursion:

$$\begin{aligned} \beta_k^t(\epsilon_k^t) &\triangleq p\left(\mathbf{x}_{k+1}^K \middle| \mathbf{x}_1^k, \epsilon_k^t\right) \\ &= \sum_{a_{k+1}} \beta_{k+1}^{t+}(\epsilon_{k+1}^t)\frac{\gamma_{k+1}^t(\epsilon_{k+1}^{t+})}{\gamma_k^t(\epsilon_k^t)} P\{a_{k+1}\}. \end{aligned} \tag{27}$$

As it can be noticed from the derivation above, for $k \in \{1, Q-1\}$ the probability density function α_k does not appear in the a posteriori probability (25) (exactly computed without

[5]For simplicity ,we consider the formulation relative to the a posteriori symbol probability. The extension for a posteriori bit probability is straightforward.

approximations) and survivors are not needed. In fact, according to the definition of α_k, a correct initialization at epoch Q (in the logarithmic domain) for the forward recursion is

$$\bar{\alpha}_Q = \begin{cases} 0 & \text{if } s_Q \text{ such that } \mu_1 = 0 \\ -\infty & \text{if } s_Q \text{ such that } \mu_1 \neq 0. \end{cases} \tag{28}$$

Hence, for $k \in \{Q+1, \ldots, N\}$, the forward recursion may be written as

$$\begin{aligned} \bar{\alpha}_k(\epsilon_k) &= \bar{\alpha}_{k-1}(\epsilon_{k-1}^{max}) + \ln p(\mathbf{x}_1^k \left| E_{k-2}^{(k-Q)}(\epsilon_{k-1}^{max}), \epsilon_{k-1}^{max}, \epsilon_k \right.) \\ &\quad - \ln p(\mathbf{x}_2^k \left| E_{k-2}^{(k-Q-1)}(\epsilon_{k-1}^{max}), \epsilon_{k-1}^{max}, \epsilon_k \right.) + \ln P\{a_{k-Q}^{max}\}, \end{aligned} \tag{29}$$

and the a posteriori symbol probability is obtained as follows:

$$P\{a_k | \mathbf{x}_1^K\} = \sum_{\epsilon_k : a(\epsilon_k) = a_k} \gamma_k((E_{k-1}^{(k-Q)}(\epsilon_k), \epsilon_k)) \alpha_k(\epsilon_k) \beta_k(\epsilon_k) \prod_{i=1}^{Q} P\{a_{k-i}\} \,. \tag{30}$$

The backward recursion is easily extended in a similar fashion, based on the survivor map built during the forward recursion. More precisely, the backward recursion may be approximated as follows:

$$\begin{aligned} \bar{\beta}_k(\epsilon_k) &\simeq \max_{a_{k+1}} \left\{ \bar{\beta}_{k+1}(\epsilon_{k+1}) + \ln p(\mathbf{x}_1^{k+1} \left| E_{k-1}^{(k+1-Q)}(\epsilon_k), \epsilon_k, \epsilon_{k+1} \right.) + \ln P\{a_{k+1}\} \right\} \\ &\quad - \ln p(\mathbf{x}_1^k \left| E_{k-1}^{(k+1-Q)}(\epsilon_k), \epsilon_k \right.), \end{aligned} \tag{31}$$

where (a_k, ϵ_k) uniquely identifies ϵ_{k+1}.

For $k > N$ the general formulation previously introduced holds.

References

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

[2] 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.

[3] G. Ungerboeck, "Channel coding with multilevel phase signaling," *IEEE Trans. Inform. Theory,* vol. IT-25, pp. 55-67, January 1982.

[4] S. LeGoff, A. Glavieux and C. Berrou, "Turbo codes and high spectral efficiency modulation," in *Proc. IEEE Intern. Conf. Commun.* (ICC '94), New Orleans, LA, U.S.A., May 1994, pp. 645-649.

[5] B. E. Wahlen and C. Y. Mai,"Turbo coding applied to pragmatic trellis-coded modulation," *IEEE Commun. Letters,* vol. 4, no. 2, pp. 65-67, February 2000.

[6] P. Roberston and T. Worz, "A novel bandwidth efficient coding scheme employing turbo codes," *IEEE J. on Sel. Areas in Comm.,* vol. 16, pp. 206-218, February 1998.

[7] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "Parallel concatenated trellis coded modulation," in *IEEE Proc. Intern. Conf. Commun.* (ICC '96), June 1996, pp. 974-978.

[8] T. Minowa and H. Ogiwara, "Application of soft-in soft-out Viterbi algorithm to turbo trellis-coded modulation," *IEICE Trans. Fundam. of Elect., Commun. and Comp. Sciences,* vol. E81-A, no. 10, pp. 2047-2054, October 1998.

[9] S. Benedetto and G. Montorsi, "Versatile bandwidth-efficient parallel and serial turbo-trellis-coded modulation," in *Proc. Intern. Symp. on Turbo Codes & Relat. Topics* (ISTC '00), Brest, France, September 2000, pp. 201-208.

[10] G. Colavolpe, G. Ferrari and R. Raheli, "Noncoherent iterative (turbo) decoding," *IEEE Trans. Commun.*, vol. 48, no. 9, pp. 1488-1498, September 2000.

[11] G. Colavolpe and R. Raheli, "The capacity of the noncoherent channel," to be published in *European Transactions on Telecommunications*, 2001. Also in *Proc. IEEE Intern. Conf. Commun.* (ICC '99), Vancouver, Canada, June 1999, pp. 1182-1186.

[12] R. H. Deng and D. J. Costello, Jr., "High rate concatenated coding systems using bandwidth efficient trellis inner codes," *IEEE Trans. Commun.*, vol. 37, no. 5, pp. 420-427, May 1989.

[13] G. Colavolpe, G. Ferrari, R. Raheli, "Reduced-state BCJR-type algorithms," *IEEE J. Sel. Areas Comm.,* May 2001.

[14] E. Biglieri, D. Divsalar, P.J. McLane and M. K. Simon, "Introduction to trellis-coded modulation with applications," Macmillan Publishing Company, 1991.

[15] G. Colavolpe and R. Raheli, "Noncoherent sequence detection," *IEEE Trans. Commun.*, vol. 47, no. 9, pp. 1376-1385, September 1999.

[16] C. Fragouli and R. D. Wesel, "Symbol interleaved parallel concatenated trellis coded modulation", in *Proc. IEEE Intern. Conf. Commun.—Comm. Theory Min-Conf.,* (ICC—CTMC '99), Vancouver, Canada, June 1999, pp. 42-46.

[17] 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, January 1988.

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

[19] 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.

[20] D. Raphaeli, "Noncoherent coded modulation," *IEEE Trans. Commun.*, vol. 44, no. 2, pp. 172-183, February 1996.

[21] S. Benedetto, D. Divsalar, G. Montorsi and F. Pollara, "A soft-input soft-output APP module for iterative decoding of concatenated codes," *IEEE Communications Letters*, vol. 1, pp. 22-24, January 1997.

[22] L.-F. Wei, "Rotationally invariant convolutional channel coding with expanded signal space - Part I: 180°", *IEEE J. on Sel. Areas in Comm.,* vol. 2, no. 5, pp. 659-671, September 1984.

[23] L.-F. Wei, "Rotationally invariant convolutional channel coding with expanded signal space - Part II: nonlinear codes", *IEEE J. on Sel. Areas in Comm.,* vol. 2, no. 5, pp. 672-686, September 1984.

[24] S. S. Pietrobon, G. Ungerboeck, L. C. Perez and D. J. Costello,Jr., "Rotationally invariant nonlinear trellis codes for two-dimensional modulation," *IEEE Trans. Inform. Theory,* vol. 40, pp. 1773-1791, November 1994.

[25] W. Liu and S. G. Wilson, "Rotationally-invariant concatenated (turbo) TCM codes," *Conference Record of Thirty-Third Asilomar Conference on Signals, Systems and Computers,* Monterey, California, October 1999, pp. 32-36.

[26] P. Robertson, E. Villerbrun, and P. Hoeher "Optimal and sub-optimal maximum a posteriori algorithms suitable for turbo decoding," *Europ. Trans. on Telecommun.,* (ETT), vol. 8, No. 2, pp. 119-125, March-April 1997.

[27] G. Colavolpe, G. Ferrari and R. Raheli, "Extrinsic information in turbo decoding: a unified view," in *Proc. IEEE Global Telecom. Conf.* (GLOBECOM '99), Rio de Janeiro, Brazil, December 1999, pp. 505-509.

[28] J. G. Proakis, "Digital communications," McGraw-Hill International Editions, 1995.

Gianluigi Ferrari was born in Parma, Italy, in 1974. He received the Dr. Ing. degree (Laurea) in electrical engineering (cum laude) from the University of Parma, Italy, in 1998. Since November 1998 he has been a Ph.D. student at the University of Parma. Since July 2000 he has been a Visiting Scholar at the Communication Sciences Institute, University of Southern California, Los Angeles, U.S.A.. His main research interests include digital transmission and detection theory, channel coding and iterative decoding techniques.

Giulio Colavolpe was born in Cosenza, Italy, in 1969. He received the Dr. Ing. degree in Telecommunications Engineering (cum laude) from the University of Pisa, Italy, in 1994 and Doctoral degree in Information Technology from the University of Parma, Italy, in 1998. Since November 1999, he has been a Research Professor at the University of Parma. In 2000, he was Visiting Scientist at the Insitut Eurècom, Valbonne, France. His main research interests include digital transmission theory, channel coding and signal processing.

Riccardo Raheli received the Dr. Ing. degree (Laurea) in Electrical Engineering "summa cum laude" from the University of Pisa, Italy, in 1983, the Master of Science in Electrical and Computer Engineering from the University of Massachusetts at Amherst, USA, in 1986, and the Doctoral degree (Perfezionamento) in Electrical Engineering "summa cum laude" from the Scuola Superiore di Studi Universitari e di Perfezionamento (now "S. Anna"), Pisa, Italy, in 1987. From 1986 to 1988 he was a Project Engineer at Siemens Telecomunicazioni, Cassina de' Pecchi (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, U.S.A.. Since 1991, he has been at the University of Parma, Italy, where he is currently an Associate Professor of Telecommunications. Since 1999, he has served on the Editorial Board of the international scientific journal "IEEE Transactions on Communications" as an Editor for Detection, Equalization and Coding.

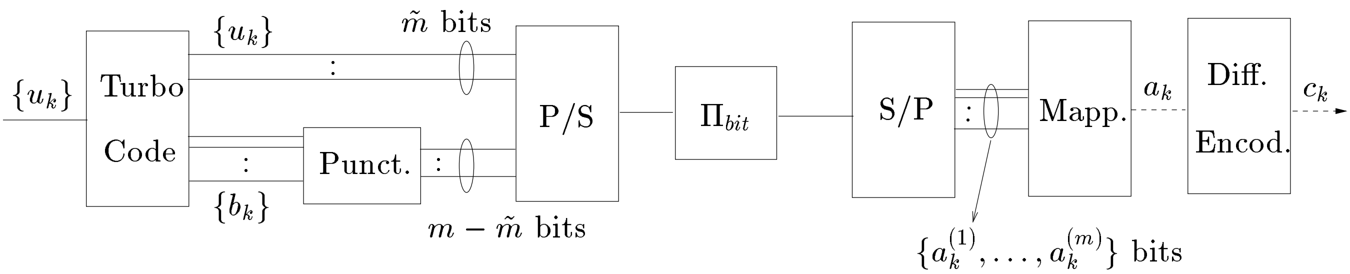

Figure 1: Berrou-type turbo code followed by differential encoding on the modulated symbols.

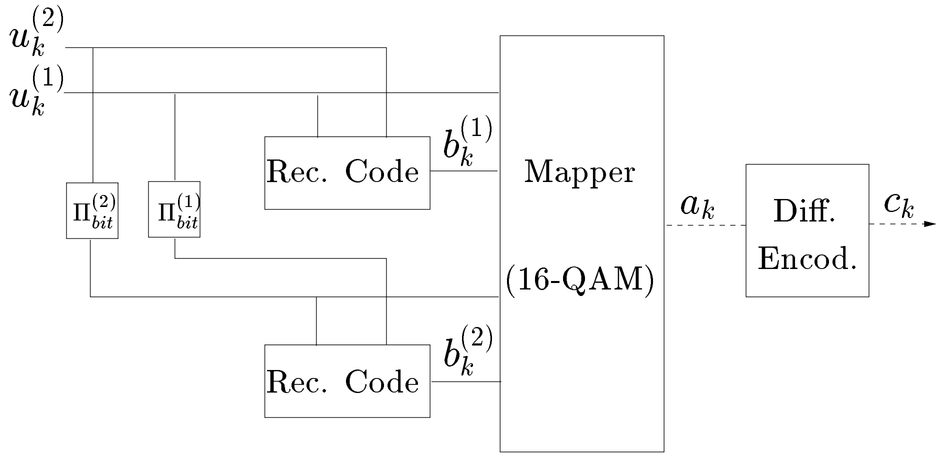

Figure 2: Benedetto-type turbo code followed by differential encoding on the modulated symbols.

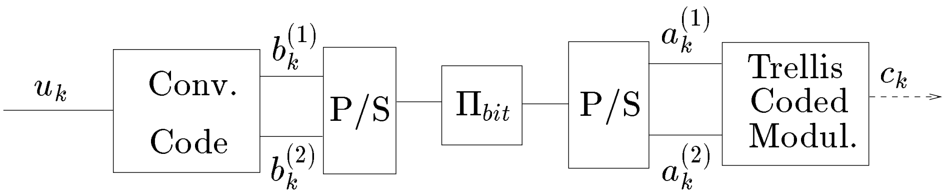

Figure 3: Serial concatenated code constituted by an outer convolutional code and an inner Ungerboeck code.

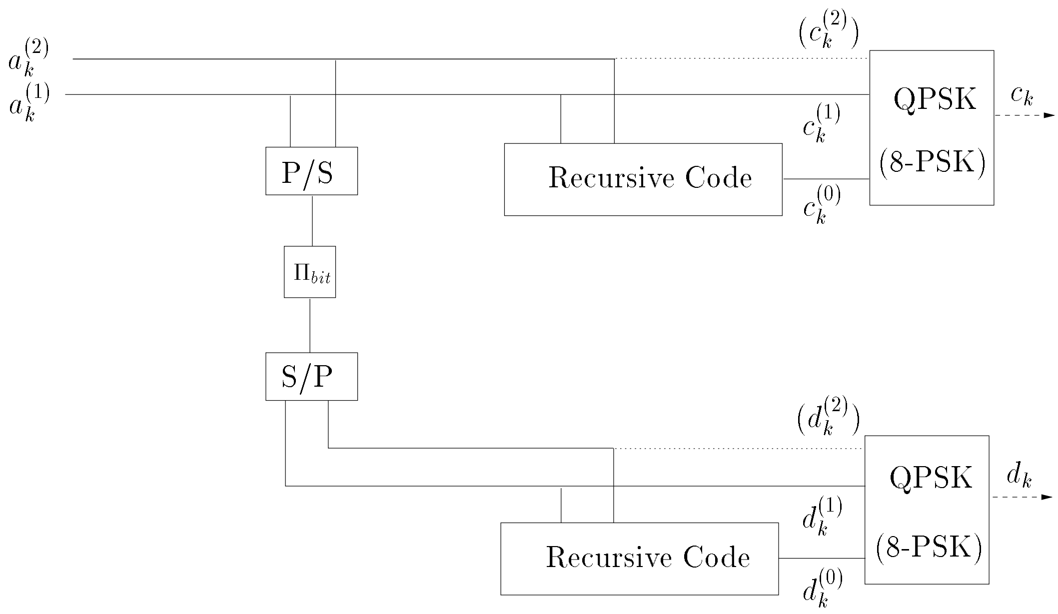

Figure 4: Benedetto et al. turbo trellis coded scheme with 8-PSK modulation. Puncturing may be embedded in the component Ungerboeck codes to consider QPSK modulation.

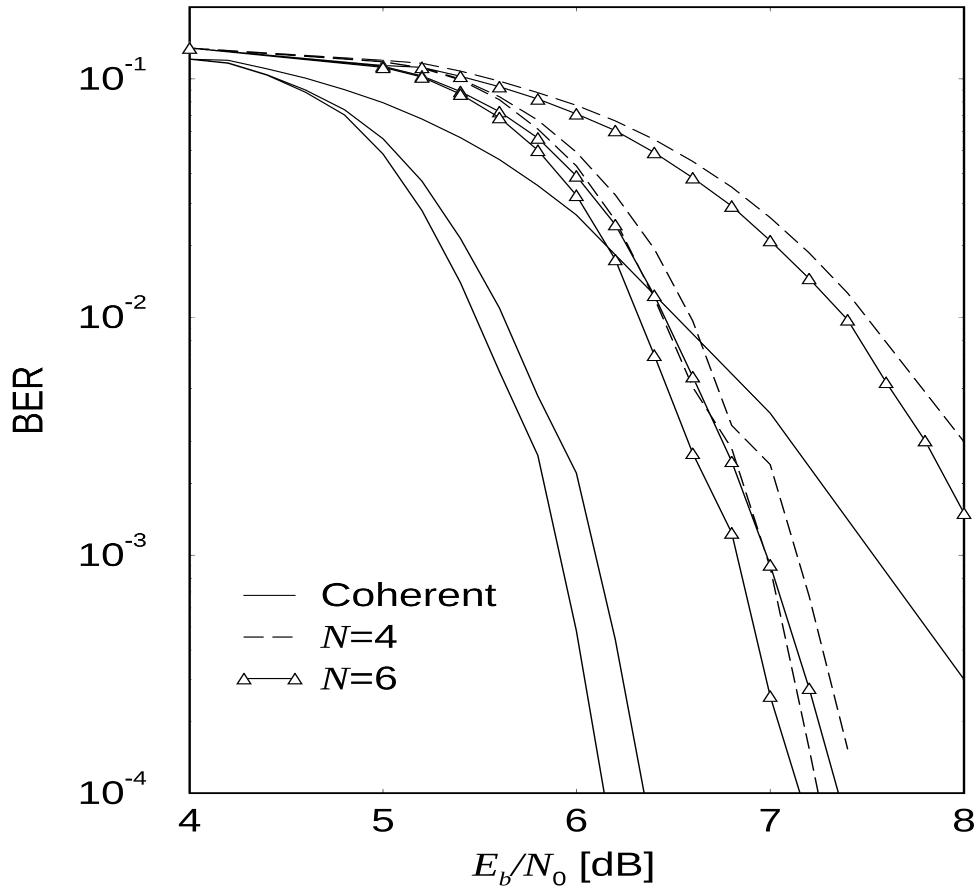

Figure 5: Performance of the system proposed in Fig. 1. The considered numbers of iterations are 1, 3 and 5 in all cases.

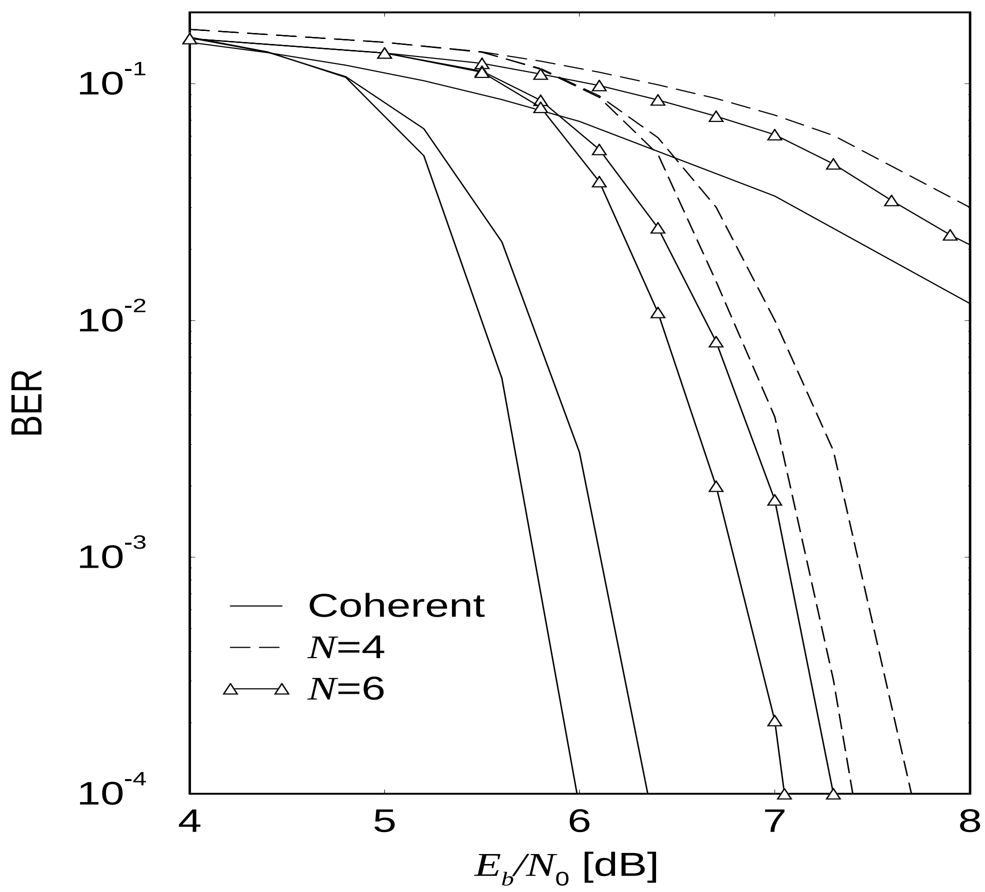

Figure 6: Performance of the system proposed in Fig. 2. The considered numbers of iterations are 1, 3 and 6 in all cases.

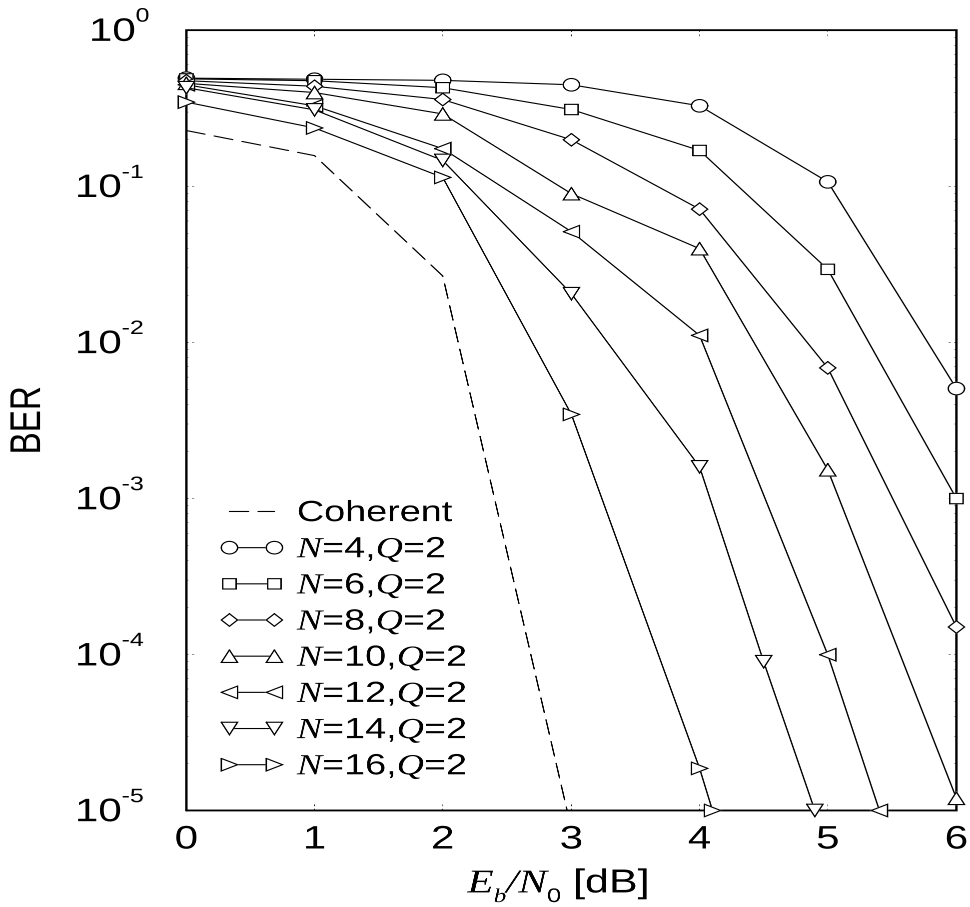

Figure 7: Performance of the system proposed in Fig. 3. The outer code has 8 states and the number of iterations is 10 in all cases.

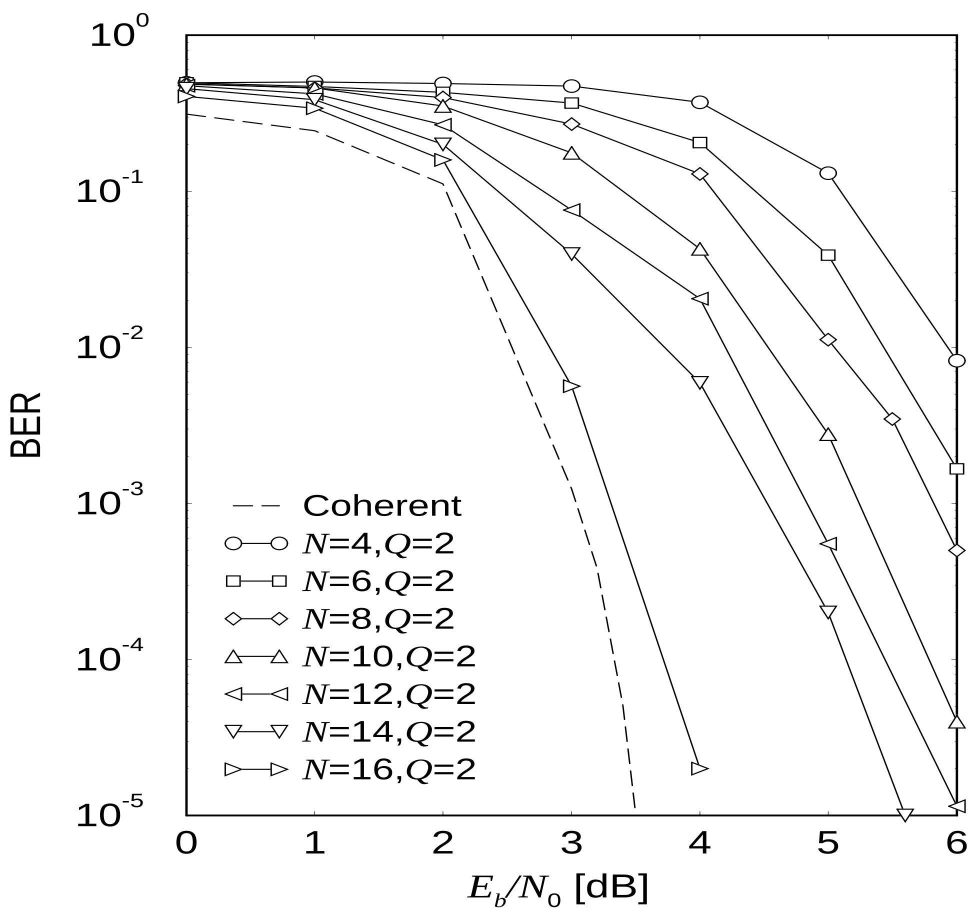

Figure 8: Performance of the system proposed in Fig. 3. The outer code has 16 states and the number of iterations is 10 in all cases.

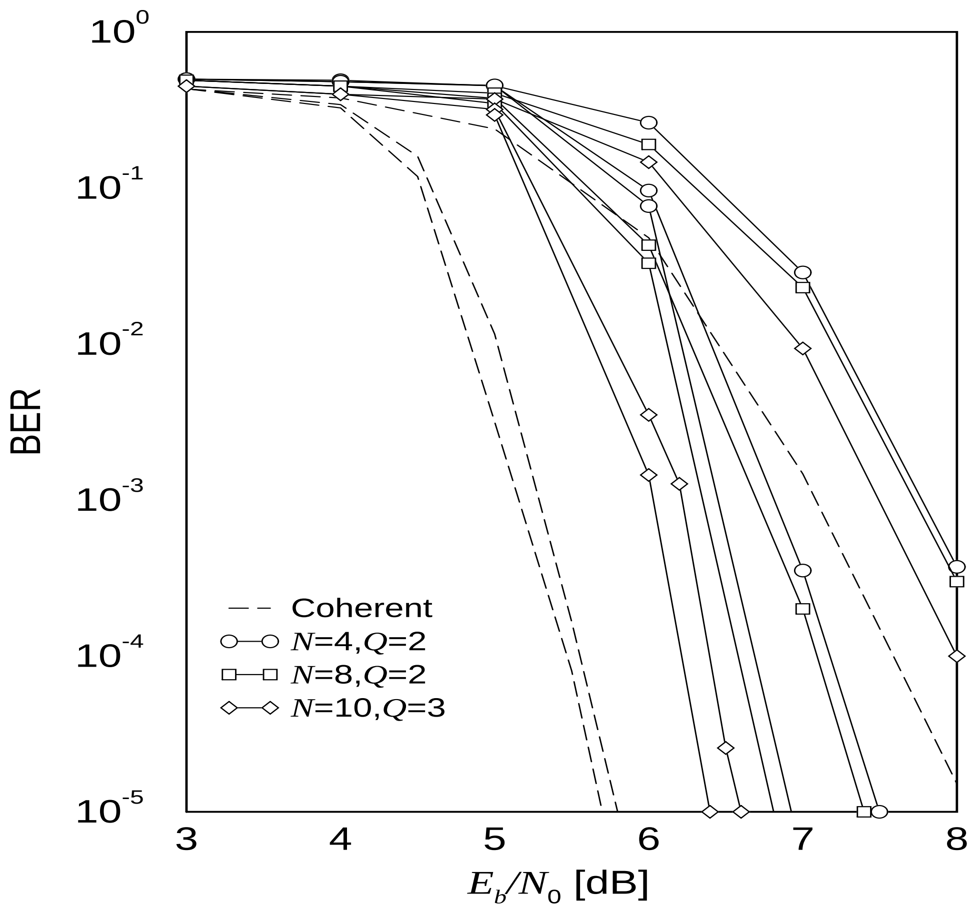

Figure 9: Performance of the system proposed in Fig. 3. The modulation format is 16-QAM and the number of iterations is 1, 5 and 10 in all cases.

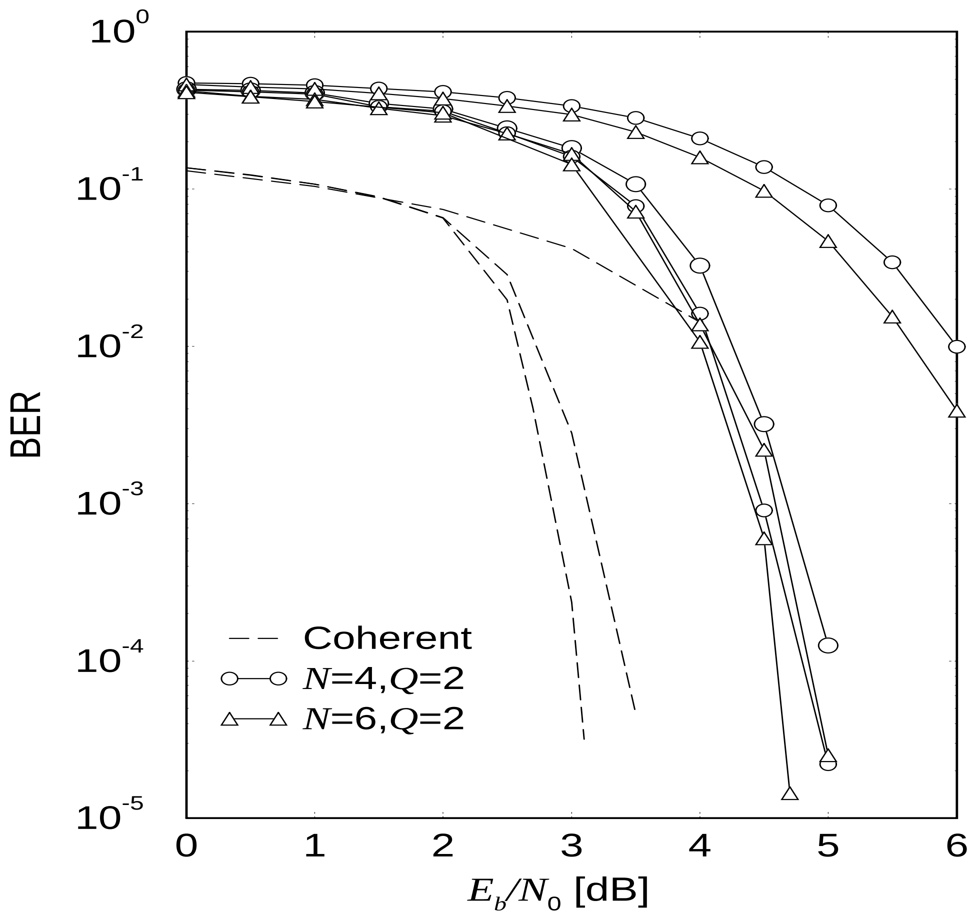

Figure 10: Performance of the system proposed in Fig. 4. The modulation format is 8-PSK and the number of iterations is 1, 3 and 6 in all cases.