

Reduced-State BCJR-Type Algorithms

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

Abstract—In this paper, we propose a technique to reduce the number of trellis states in BCJR-type algorithms, i.e., algorithms with a structure similar to that of the well-known algorithm by Bahl, Cocke, Jelinek, and Raviv (BCJR). This work is inspired by reduced-state sequence detection (RSSD). The key idea is the construction, during one of the recursions in the reduced-state trellis, of a “survivor map” to be used in the other recursion. In a more general setting, two distinct survivor maps could be determined in the two recursions and used jointly to approximate the *a posteriori* probabilities. Three examples of application to iterative decoding are shown: 1) coherent detection for intersymbol interference (ISI) channels; 2) noncoherent detection based on an algorithm recently proposed by the authors; and 3) detection based on linear prediction for Rayleigh fading channels. As in classical RSSD, the proposed algorithm allows significant state-complexity reduction with limited performance degradation.

Index Terms—Error correcting codes, iterative decoding, soft-input soft-output, turbo codes.

I. INTRODUCTION

IN CURRENT digital communication systems based on iterative detection/decoding, it is necessary to determine the likelihood that a particular symbol has been transmitted. Soft-output algorithms [1]–[6] have been considered with renewed interest, the recent most famous application of them being the iterative decoding of interleaved concatenated codes [7], [8]. The Bahl, Cocke, Jelinek, and Raviv (BCJR) algorithm [2] is based on a symbol-wise maximum *a posteriori* probability (MAP) criterion and proves to be optimal for estimating the states or outputs of a Markov chain observed in white noise. This algorithm is rather complex to implement because of the following

- 1) Problems of numerical representation of very low probability values;
- 2) Use of nonlinear functions;
- 3) Numerous mixed multiplications and additions

In order to reduce these problems, realizations of this algorithm in the logarithmic domain have been proposed, which result in useful simplifications [9]–[11].

Whenever the overall transmission system can be considered as finite memory, the receiver has to take into account the overall memory in a trellis diagram. For example, this is the case for a convolutional code in an additive white Gaussian noise (AWGN) channel and/or a finite intersymbol interference (ISI) channel. A frequency nonselective fading channel is not rigorously finite

memory [12] but it is generally assumed as being so [13]. Once the overall system memory has been identified, a relevant trellis diagram can be defined and soft-output algorithms, such as the BCJR algorithm, can be employed. If the overall memory is large, the complexity of a BCJR algorithm, even with the cited logarithmic simplifications, may be unacceptable because of the trellis size. Various solutions have appeared in the literature to reduce the complexity of the decoder trellis diagram. The so-called T-algorithm and M-algorithm reduce the number of paths which are searched in the trellis diagram [14]. Trellis splicing based on a confidence criterion may be used to detect reliable information symbols early on during decoding [15]. When the channel memory is not finite, as for a “noncoherent channel” [16], it is possible to utilize soft-output algorithms with a structure similar to that of the BCJR algorithm, which try to partially take into account this memory by means of an augmented trellis [6]. This approach can be generalized to other channels with infinite memory by means of suitable algorithms which will be referred to as *BCJR-type*.

In this paper, we propose an extension of reduced-state sequence detection (RSSD) [17]–[19] to a general BCJR-type algorithm. Further generalizations toward the application of per-survivor processing (PSP) techniques [20] in order to dynamically estimate unknown channel parameters are also possible, as for example in [21]–[23].

The application of the principle of RSSD to the complexity reduction of soft-input soft-output (SISO) algorithms was previously addressed. As an example, RSSD may be straightforwardly applied to a soft-output Viterbi algorithm as in [24] (see also [6] for a recent application). Other applications were brought to the authors’ attention during the review process [25]–[29]. Most of the algorithms proposed in these papers deal with ISI channels and none considers iterative decoding. Reference [25], mentions the use of RSSD to simplify the forward and backward recursions of a BCJR algorithm employed as a soft-output equalizer in a frequency-selective channel. In [26], a soft-output equalizer with soft-decision feedback is derived from a modified version of the Lee algorithm [30] for a minimum-phase channel impulse response. It is also shown that the algorithm there proposed may be given an equivalent formulation by introducing a backward recursion, making it similar to the BCJR algorithm. In [27], a family of Bayesian conditional decision feedback estimators are proposed, the T-algorithm is applied to reduce the complexity, and the possibility of applying RSSD techniques is mentioned. This idea is further developed in [28], where reduced-complexity versions of the fixed-lag soft-output algorithm introduced in [5] are proposed based on the application of the T-algorithm and RSSD, although an explicit introduction of the concept of survivor is not given. A similar detection strategy is used in [29] to detect continuous phase modulations.

Manuscript received April 26, 2000; revised November 22, 2000. This work was supported by Ministero dell’Università e della Ricerca Scientifica e Tecnologica (MURST), Italy. This work was presented in part at the *IEEE International Conference on Communications*, (ICC’00), New Orleans, LA, USA, June 2000.

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

Publisher Item Identifier S 0733-8716(01)03902-6.

A reduced-state BCJR algorithm similar to that here presented and its application to iterative decoding was proposed independently in [31] at the same time of the first presentation of this work [32].

The paper is organized as follows. In Section II, we give a general definition of BCJR-type algorithms. In Section III, we propose a reduced-state BCJR-type algorithm in general terms. In Section IV, we consider examples of application of the proposed technique for ISI channels, noncoherent channels, and Rayleigh flat-fading channels. Numerical results are presented in Section V. Conclusions are drawn in Section VI.

II. BCJR-TYPE ALGORITHMS

An information source emits a sequence of independent and identically distributed information symbols a_k which is possibly encoded and transmitted through a channel. We assume that a soft-output algorithm can be devised to compute the *a posteriori* probabilities of the transmitted information symbols or good approximations of them as in [6]. The calculation of these probabilities is generally based on the observation of a suitable sequence of samples at the input of the receiver. We denote this sequence by $\mathbf{x}_1^K \triangleq \{\mathbf{x}_k\}_{k=1}^K$, where K is the transmission length and vector \mathbf{x}_k is the observation at the k th signaling interval. This vector notation accounts for possible multiple observations at the k th signaling interval.

Let ζ be the number of properly defined receiver trellis states, which depends on the modulation format, the presence of coding, the channel model and possible approximations. Let S_k denote the trellis state at epoch k . For simplicity, we assume that a single information symbol a_k is associated with a transition from state S_k to state S_{k+1} and there are no parallel transitions. The extension to more general cases, such as convolutional codes with rate k/n and $k > 1$ or trellis coded modulations, can be easily obtained by a suitable notation. To simplify the notation, we also assume that $a_k \in \{0, \dots, M-1\}$, although in general a_k could belong to any M -ary set. Therefore, a trellis section is characterized by ζM branches, each of them being uniquely identified by the couple of states it connects. We denote the branch which connects $S_k = m'$ to $S_{k+1} = m$ by $e_k(m', m)$ and the information symbol which drives this transition by $a(e_k)$.

With these definitions, we assume that a good approximation of the *a posteriori* probability of symbol a_k can be determined on the basis of the considered observations. We denote this value by $P\{a_k|\mathbf{x}_1^K\}$ and assume that it can be expressed, in analogy with a generalized version of the BCJR algorithm [2], [6] as¹

$$P\{a_k|\mathbf{x}_1^K\} = P\{a_k\} \sum_{e_k:a(e_k)=a_k} \gamma_k(e_k)\alpha_k(e_k)\beta_k(e_k) \cdot P\{S^-(e_k)\} \quad a_k = 0, \dots, M-1 \quad k = 1, \dots, K. \quad (1)$$

The sum in (1) is extended over all transitions of epoch k associated with information symbol a_k . The quantities $\gamma_k(e_k)$, $\alpha_k(e_k)$

and $\beta_k(e_k)$ are proportional to suitable probability density functions. They depend on the particular trellis branch and their expressions can be specified for the transmission system under consideration. $P\{a_k\}$ and $P\{S^-(e_k)\}$ represent the *a priori* probabilities of information symbol a_k and beginning state of branch e_k , respectively. If the symbols are *a priori* equiprobable, these probabilities reduce to constants. On the contrary, if the algorithm is used in an iterative decoding process, these probabilities in general are not constant in successive iterations, and must be explicitly considered.

A proper normalization of $\gamma_k(e_k)$, $\alpha_k(e_k)$, and $\beta_k(e_k)$ is necessary in order to correctly express the *a posteriori* probabilities in (1). However, an arbitrary normalization of these quantities would scale the resulting *a posteriori* probabilities by a factor which would not affect the correctness of the algorithms under consideration. As a consequence, we take the notational liberty of denoting by $P\{a_k|\mathbf{x}_1^K\}$ a *scaled version* of the *a posteriori* probabilities and referring to them as *a posteriori* probabilities for simplicity. Similarly, we refer to $\gamma_k(e_k)$, $\alpha_k(e_k)$ and $\beta_k(e_k)$ as probability density functions, although strictly speaking they may be so except for a normalization constant. If necessary, we may further relax the terminology by referring to probabilities and probability density functions and using the relevant notation even if an approximation may be involved.

Similarly to the BCJR algorithm, we assume that we can compute the probability density functions $\alpha_k(e_k)$ and $\beta_k(e_k)$ by means of forward and backward recursions [2], [6], [7]. Defining $S^+(e_k)$ as the final state of transition e_k we may write

$$\alpha_k(e_k) = \sum_{e_{k-1}:S^+(e_{k-1})=S^-(e_k)} \psi_k(e_{k-1}, e_k)\alpha_{k-1}(e_{k-1}) \cdot P\{\overset{\circ}{a}(e_{k-1})\} \quad (2)$$

$$\beta_k(e_k) = \sum_{e_{k+1}:S^+(e_k)=S^-(e_{k+1})} \phi_{k+1}(e_k, e_{k+1})\beta_{k+1}(e_{k+1}) \cdot P\{a(e_{k+1})\} \quad (3)$$

where $\psi_k(e_{k-1}, e_k)$ and $\phi_{k+1}(e_k, e_{k+1})$ are suitable, possibly scaled, probability density functions that, in general, may depend on two consecutive branches. By $\overset{\circ}{a}(e_{k-1})$ we denote the information symbol “lost” in transition e_{k-1} , i.e., the oldest symbol in the initial state $S^-(e_{k-1})$. The couple $(S^+(e_{k-1}), \overset{\circ}{a}(e_{k-1}))$ uniquely identifies $S^-(e_{k-1})$. In (2), 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 (3), relative to the trellis section at epoch $k+1$, may be interpreted similarly.

We may also define reliability values of each symbol a_k , with respect to a reference symbol $a_k = 0$, by considering logarithmic likelihood ratios of the following type

$$L(a_k = i) \triangleq \ln \frac{P\{a_k = i|\mathbf{x}_1^K\}}{P\{a_k = 0|\mathbf{x}_1^K\}} = \ln P\{a_k = i|\mathbf{x}_1^K\} - \ln P\{a_k = 0|\mathbf{x}_1^K\} = \ln \frac{P\{a_k = i\}}{P\{a_k = 0\}} + \ln \frac{\sum_{e_k:a(e_k)=i} \gamma_k(e_k)\alpha_k(e_k)\beta_k(e_k)P\{S^-(e_k)\}}{\sum_{e_k:a(e_k)=0} \gamma_k(e_k)\alpha_k(e_k)\beta_k(e_k)P\{S^-(e_k)\}} \quad i = 1, 2, \dots, M-1. \quad (4)$$

¹As customary, we do not use a specific notation to distinguish between random quantities and their hypothetical values. Specifically, $P\{a_k\}$ denotes the *a priori* probability that the information symbol at the k th epoch takes on the hypothetical value a_k . A consistent notation is used throughout and the correct interpretation should be clear from the context.

The second term in (4) represents the generated extrinsic information, that is the component of the likelihood values to be fed back in an iterative decoding process (as an example, see [7], [8]). Equations (1)–(3) can be reformulated in the logarithmic domain [9]–[11].

We remark that the algorithm described by (1)–(3) reduces to the classical BCJR algorithm [2] in the special case of strictly finite memory. In this case, the probability density functions α_k and β_k depend on transition e_k through a single state S_k . Moreover, ψ_k and ϕ_{k+1} depend on a single transition e_k and reduce to γ_{k-1} and γ_{k+1} , respectively.

The proposed formulation of BCJR-type algorithms generalizes the classical one [2] and may be suitable for specific detection problems characterized by strictly unlimited memory. As an example, the noncoherent BCJR-type algorithm proposed in [6] is obtained by truncating the infinite memory of the considered channel and falls within the considered formulation. In fact in [6], the expression of the *a posteriori* probability involves a probability density function γ_k which depends on a single transition, the forward and backward recursions involve probability density functions ψ_k and ϕ_k which depend on two successive transitions, and α_k and β_k depend on a transition (not on a state). As shown in the Appendix, two alternative formulations of a BCJR-type algorithm may be conceived but none reduces exactly to the classical BCJR algorithm. The choice of the formulation previously described has several motivations. First, the alternative formulations described in the Appendix involve augmented trellis diagrams with larger number of states. As this paper deals with reduced-state algorithms, it would be contradictory to begin with augmented trellis diagrams. Second, the described formulation is easily reduced to the classical BCJR algorithm when the overall memory is finite, as opposed to the alternative formulations (see the Appendix). Finally, the described formulation is consistent with that adopted in [6].

III. REDUCED-STATE BCJR-TYPE ALGORITHMS

A BCJR-type algorithm formulated as in Section II requires the calculation of essentially two kinds of probability density functions: γ_k , which depends on a single transition, and ψ_k and ϕ_k , which may depend on two consecutive transitions. Let us assume that η code symbols $(c_k^{(1)}, \dots, c_k^{(\eta)})$ are generated by a single information symbol a_k . By using a vector notation, we denote this sequence of code symbols by $\mathbf{c}_k = (c_k^{(1)}, \dots, c_k^{(\eta)})$ and refer to \mathbf{c}_k as the code (vector) symbol at epoch k . In general, a single transition can be related to N code symbols, that is $e_k \equiv (\mathbf{c}_{k-N+1}, \dots, \mathbf{c}_k)$, where the symbol \equiv denotes a one-to-one correspondence between transition e_k and the sequence of N code symbols. The parameter N is related to the channel and/or the detection strategy. As an example, for an AWGN channel, $N = 1$ in the case of coherent detection, whereas N is the assumed phase memory for noncoherent detection based on the algorithm proposed in [6]. A single branch can be related to V information symbols, that is $e_k \equiv (a_{k-V+1}, \dots, a_k)$. The relation between integers V (relative to information symbols) and N (relative to code symbols) depends on the coding structure.² We assume that the receiver observes,

²If the considered code is recursive, as it is usually for turbo codes, one should consider some encoder state variables together with the information symbols in the definition of S_k and e_k [6].

at each epoch, a (vector) sample $\mathbf{x}_k = (x_k^{(1)}, \dots, x_k^{(\eta)})$ at the output of the channel. Generalizations to the case of oversampling or multiple decoder inputs are straightforward. Denoting by L the length of the discrete-time overall impulse response, each sample \mathbf{x}_k will depend on L code symbols $(\mathbf{c}_{k-L+1}, \dots, \mathbf{c}_k)$.

We consider a trellis diagram with a reduced number of states $\zeta' < \zeta$ and denote its generic state by s_k . Following the principle of RSSD [17]–[19], we define the state reduction by assuming that a transition $e_k \equiv (s_k, s_{k+1})$ in the reduced trellis diagram is equivalent to a sequence (a_{k-Q+1}, \dots, a_k) of information symbols, with $Q < V$.³ The central point is that, even with this reduction of states, knowledge of $V + 1$ information symbols is necessary in order to specify $(\mathbf{c}_{k-N}, \dots, \mathbf{c}_k)$, which is required to compute γ_k , ψ_k , and ϕ_k . Hence, for a given transition e_k , we need to estimate symbols a_{k-i} for $i \geq Q$ in order to compute the metric. According to RSSD, this estimate could be obtained by tracking the survivor of each state in the reduced-state trellis. However, how could we define a survivor in this case?

A “full-state” BCJR-type algorithm runs first a forward recursion to compute $\alpha_k(e_k)$ for each transition e_k and epochs $k = 1, 2, \dots, K$. As shown in (1) and (2), α_k is associated with a transition e_k , instead of a single state S_k as in a classical case. Hence, a “survivor” associated with a single transition has to be defined. By considering the algorithm in the logarithmic domain, the forward recursion of α_k ((2)) can be expressed as

$$\begin{aligned} \bar{\alpha}_k(e_k) \triangleq \ln \alpha_k(e_k) = \ln & \sum_{e_{k-1}: S^+(e_{k-1})=S^-(e_k)} \\ & \cdot \exp\{\bar{\psi}_k(e_{k-1}, e_k) + \bar{\alpha}_{k-1}(e_{k-1}) \\ & + \ln P\{\hat{a}(e_{k-1})\}\} \end{aligned} \quad (5)$$

where $\bar{\psi}_k(e_{k-1}, e_k) \triangleq \ln \psi_k(e_{k-1}, e_k)$. We can approximate this recursion by the “max-log” approximation as follows [9]

$$\begin{aligned} \bar{\alpha}_k(e_k) \simeq & \max_{e_{k-1}: S^+(e_{k-1})=S^-(e_k)} \{\bar{\psi}_k(e_{k-1}, e_k) \\ & + \bar{\alpha}_{k-1}(e_{k-1}) + \ln P\{\hat{a}(e_{k-1})\}\}. \end{aligned} \quad (6)$$

In (6), it is intuitive to interpret the term to be maximized as a “metric” associated with a path ending with transition e_k , as in a classical Viterbi algorithm.

Considering the reduced-state trellis and assuming we know the survivors of each transition e_{k-1} , we now show how the survivors can be extended to time instant k by using the forward recursion. We may define by $E_{k-j}^{(i)}(e_{k-1})$ the sequence of i transitions reaching epoch $k - j$ along the survivor of transition e_{k-1} , i.e., $E_{k-j}^{(i)}(e_{k-1}) = (\hat{e}_{k-j-i+1}, \dots, \hat{e}_{k-j}) \equiv (\hat{a}_{k-j-i-Q+2}, \dots, \hat{a}_{k-j-Q+1})$. Each transition \hat{e}_{k-l} , for $l \in \{j, \dots, j+i-1\}$, in $E_{k-j}^{(i)}(e_{k-1})$ and information symbol \hat{a}_{k-j-h} , for $h \in \{Q-1, \dots, Q+i-2\}$, depend on the transition e_{k-1} . Hence, the couple $(E_{k-2}^{(V-Q)}(e_{k-1}), e_{k-1})$ uniquely identifies the sequence of information symbols $(\hat{a}_{k-V}, \dots, \hat{a}_{k-Q-1}, a_{k-Q}, \dots, a_{k-1})$ where $\{\hat{a}_{k-i}\}_{Q+1}^V$ represent the information symbols in the path history of transition

³More complex techniques based on set partitioning may also be employed [17], [19].

ϵ_{k-1} . Similarly, we can extend a couple of transitions $(\epsilon_{k-1}, \epsilon_k)$ in order to get $V+1$ information symbols. With these definitions, it is easy to extend the survivors by a step, i.e., to epoch k . In the reduced-state trellis, taking into account paths $E_{k-1}^{(i)}(\epsilon_{k-1})$ associated with transitions at epoch $k-1$, (6) reduces to

$$\bar{\alpha}_k(\epsilon_k) \simeq \max_{\epsilon_{k-1}: s^+(\epsilon_{k-1})=s^-(\epsilon_k)} \{ \bar{\psi}_k(E_{k-2}^{(V-Q)}(\epsilon_{k-1}), \epsilon_{k-1}, \epsilon_k) + \bar{\alpha}_{k-1}(\epsilon_{k-1}) + \ln P\{\hat{a}(\epsilon_{k-1})\} \}. \quad (7)$$

For each transition ϵ_k , the branch ϵ_{k-1}^{\max} that maximizes (7) should be stored.

During the forward recursion, a “full-state” BCJR-type algorithm does not necessarily make a selection between the metrics of the paths terminating with a given transition ϵ_k as shown in (6) but rather determines a new metric $\alpha_k(\epsilon_k)$ as an average of these path metrics as in (5). Similarly, in the reduced-state trellis equation (7) does not necessarily have to be used to compute $\alpha_k(\epsilon_k)$ but only to determine the survivor associated with transition ϵ_k .

We note that the concept of survivor associated with a transition can be reconciled with the classical one by properly defining an augmented trellis diagram, to be used in the forward and backward recursions, and reformulating the described BCJR-type algorithm. The Appendix describes this possible formulation.

In the reduced-state trellis, it is necessary to keep track of the survivors associated with each transition, at each epoch, in the forward recursion of α_k only. In fact, these survivors may be used in the backward recursion of β_k and in the calculation of the probability $P\{a_k | \mathbf{x}_1^K\}$, by evaluating the probability density function γ_k in (1). We will refer to the ensemble of the survivors generated in the forward recursion as “survivor map.” The basic structural idea can be further generalized. In fact, it is not necessarily the forward recursion which allows the construction of a survivor map to be used in the backward recursion. One could build a survivor map during the backward recursion (run first) and use it in the forward recursion. Finally, one could build two distinct survivor maps during the forward and backward recursion, and consider an extended γ_k in the combination (1). In the rest of this section, we will refer for simplicity to the case of a survivor map built during the forward recursion but generalizations of the proposed formulation to the other cases are possible.

Given the above definitions, we can reformulate (1)–(3), obtaining

$$P\{a_k | \mathbf{x}_1^K\} = P\{a_k\} \sum_{\epsilon_k: a(\epsilon_k)=a_k} \gamma_k(E_{k-1}^{(V-Q)}(\epsilon_k), \epsilon_k) \cdot \alpha_k(\epsilon_k) \beta_k(\epsilon_k) P\{s^-(\epsilon_k)\} \quad (8)$$

$$\alpha_k(\epsilon_k) = \sum_{\epsilon_{k-1}: s^+(\epsilon_{k-1})=s^-(\epsilon_k)} \cdot \psi_k(E_{k-2}^{(V-Q)}(\epsilon_{k-1}), \epsilon_{k-1}, \epsilon_k) \cdot \alpha_{k-1}(\epsilon_{k-1}) P\{\hat{a}(\epsilon_{k-1})\} \quad (9)$$

$$\beta_k(\epsilon_k) = \sum_{\epsilon_{k+1}: s^+(\epsilon_k)=s^-(\epsilon_{k+1})} \cdot \phi_{k+1}(E_{k-1}^{(V-Q)}(\epsilon_k), \epsilon_k, \epsilon_{k+1}) \cdot \beta_{k+1}(\epsilon_{k+1}) P\{a(\epsilon_{k+1})\} \quad (10)$$

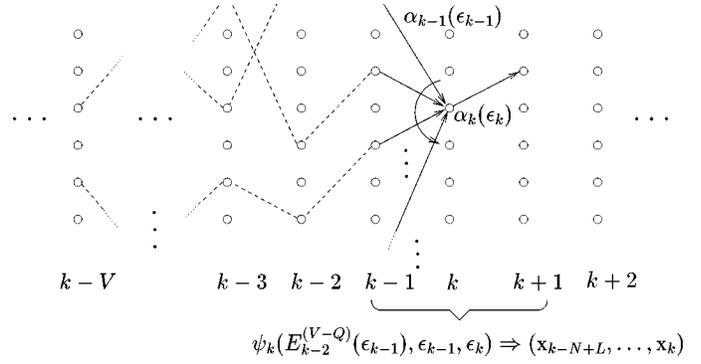


Fig. 1. Forward recursion of the probability density function $\alpha_k(\epsilon_k)$ for a general reduced-state BCJR-type algorithm.

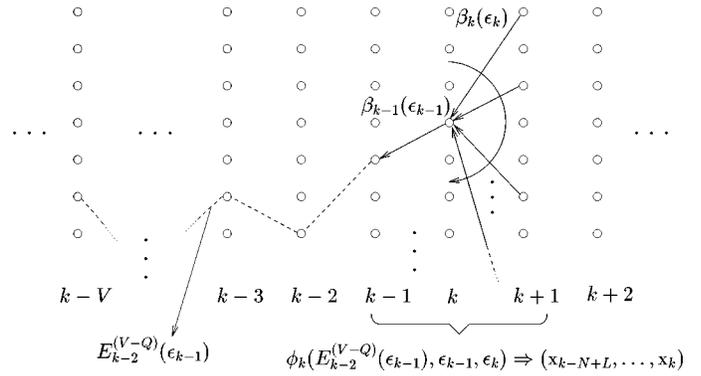


Fig. 2. Backward recursion of the probability density function $\beta_k(\epsilon_k)$ for a general reduced-state BCJR-type algorithm. The metric ϕ_k is calculated using the survivor map previously built in the forward recursion.

where $s^-(\epsilon_k)$ and $s^+(\epsilon_{k+1})$ are the initial and final states of transitions ϵ_k and ϵ_{k+1} , respectively. In Fig. 1, it is shown how the forward recursion proceeds in the reduced-state trellis according to (9). In order to compute $\alpha_k(\epsilon_k)$, one should consider the M probability density functions $\{\alpha_{k-1}(\epsilon_{k-1})\}$ such that $s^+(\epsilon_{k-1}) = s^-(\epsilon_k)$ and, for each of them, compute the branch metric ψ_k by considering the symbols associated with the survivor of transition ϵ_{k-1} . The backward recursion proceeds similarly using the survivor map generated during the forward recursion, as shown in Fig. 2.

Two logarithmic simplified versions of a reduced-state BCJR-type algorithm may be considered. A first one is derived by applying the max-log approximation [9], [10] to (8)–(10). By defining $\bar{\gamma}_k \triangleq \ln \gamma_k$ and $\bar{\psi}_k \triangleq \ln \psi_k$, (9) can be approximated as follows

$$\bar{\alpha}_k(\epsilon_k) \simeq \max_{\epsilon_{k-1}: s^+(\epsilon_{k-1})=s^-(\epsilon_k)} \{ \bar{\psi}_k(E_{k-2}^{(V-Q)}(\epsilon_{k-1}), \epsilon_{k-1}, \epsilon_k) + \bar{\alpha}_{k-1}(\epsilon_{k-1}) + \ln P\{\hat{a}(\epsilon_{k-1})\} \}. \quad (11)$$

Similar logarithmic extensions hold for (8) and (10), by defining $\bar{\beta}_k \triangleq \ln \beta_k$ and $\bar{\phi}_k \triangleq \ln \phi_k$. This version of the algorithm would be motivated, besides complexity reduction, by the fact that the forward recursion of $\bar{\alpha}_k$ allows to determine the survivors without additional operations. In a second logarithmic simplified version, the max-log approximation for the forward and backward recursions is maintained but a “full” combination is considered in order to compute $\ln P\{a_k | \mathbf{x}_1^K\}$. Hence, we

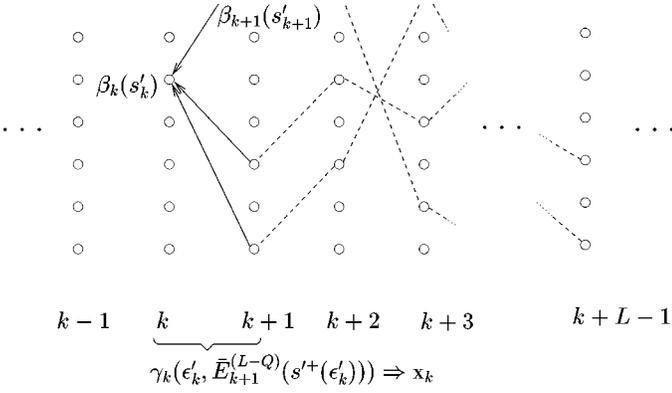


Fig. 4. Backward recursion of the probability density function $\beta_k(s_k)$ for the reduced-state BCJR algorithm in the case of coherent detection over ISI channels. The survivor map is built during this recursion.

and (12) reduces to

$$\begin{aligned} \ln P\{a_k | \mathbf{x}_1^K\} &= \ln P\{a_k\} + \ln \sum_{\epsilon_k: a(\epsilon_k)=i} \\ &\cdot \exp \left\{ \bar{\gamma}_k(E_{k-1}^{(L-Q)}(s^-(\epsilon_k)), \epsilon_k) \right. \\ &+ \bar{\alpha}_k(s^-(\epsilon_k)) + \bar{\beta}_k(s^+(\epsilon_k)) \\ &\left. + \sum_{i=1}^{Q-1} \ln P\{a_{k-i}\} \right\}. \end{aligned} \quad (22)$$

If the overall impulse response $\{f_i\}_{i=0}^{L-1}$ is maximum phase,⁴ efficient definitions of trellis state and transition may be $s'_k = (a_{k-L+1}, \dots, a_{k-L+Q-1})$ and $\epsilon'_{k+1} = (a_{k-L+1}, \dots, a_{k-L+Q})$, respectively. As mentioned in Section III, a new version of a reduced-state BCJR-type algorithm can be conceived, which starts with the backward recursion, builds a “survivor map” compatible with the new reduced state s'_k , and runs the forward recursion. Related reverse-time processing structures, suited to impulse responses with energy concentrated toward the end, are considered in [37], [38]. In this “specular” version of the previously introduced algorithm, during the backward recursion the information symbol a_{k-L+1} is relative to the transition from state s'_{k+1} to state s'_k and symbol a_{k-L+Q} is discarded. The formulation is a straightforward extension of that previously introduced, the only modification being a termination of the reduced-state trellis necessary to better initialize the backward recursion. Fig. 4 schematically shows how the survivor map can be built during the backward recursion. The survivor may be denoted by $\bar{E}_{k+1}^{(L-Q)}(s'^+(\epsilon'_{k+1})) = (\hat{a}_{k-L+Q+1}, \dots, \hat{a}_k)$, where $s'^+(\epsilon'_{k+1})$ indicates the final state s'^+_{k+2} of transition ϵ'_{k+1} .

This example leads to a more general conclusion on the possible applications of the proposed technique. Survivor maps can be built during both the forward and backward recursions, depending on the overall channel impulse response and, consequently, on the structure of the received signal samples. As an example, a proper use of survivor maps built during both recursions may prove useful if the overall impulse response is “mixed phase.” A specific investigation of this case was not performed

⁴For example, if a causal whitening filter is selected.

because beyond the scope of the paper. Related aspects are discussed in [22].

B. Noncoherent Channels

In the case of noncoherent decoding, we refer to the BCJR-type algorithm proposed in [6]. We may express this algorithm in terms of the general formulation given above with $L = 1$ and $N > 1$ (hence, the receiver considers a “window” of N consecutive samples at each epoch). Denoting by μ_k the encoder state, the “decoder state” is defined by $S_k = (a_{k-1}, \dots, a_{k-N+1}, \mu_{k-N+1})$ [6]; hence, $e_k \equiv (\mathbf{c}_k, \dots, \mathbf{c}_{k-N+1})$. In this case, the probability density functions γ_k , ψ_k and ϕ_k appearing in (1)–(3) specialize as follows

$$\begin{aligned} \gamma_k(e_k) &= p(\mathbf{x}_{k-N+1}^k | e_k) \\ &\propto \exp \left\{ -\frac{1}{2\sigma^2} \sum_{i=0}^{N-1} [|\mathbf{x}_{k-i}|^2 + |\mathbf{c}_{k-i}|^2] \right\} \\ &\cdot \text{I}_0 \left(\frac{1}{\sigma^2} \left| \sum_{i=0}^{N-1} \mathbf{x}_{k-i} \mathbf{c}_{k-i}^H \right| \right) \end{aligned} \quad (23)$$

$$\begin{aligned} \psi_k(e_{k-1}, e_k) &= p(\mathbf{x}_{k-N} | \mathbf{x}_{k-N+1}^k, e_{k-1}, e_k) \\ &\propto \exp \left\{ -\frac{[|\mathbf{x}_{k-N}|^2 + |\mathbf{c}_{k-N}|^2]}{2\sigma^2} \right\} \\ &\cdot \frac{\text{I}_0 \left(\frac{1}{\sigma^2} \left| \sum_{i=0}^N \mathbf{x}_{k-i} \mathbf{c}_{k-i}^H \right| \right)}{\text{I}_0 \left(\frac{1}{\sigma^2} \left| \sum_{j=0}^{N-1} \mathbf{x}_{k-j} \mathbf{c}_{k-j}^H \right| \right)} \end{aligned} \quad (24)$$

$$\begin{aligned} \phi_k(e_{k-1}, e_k) &= p(\mathbf{x}_k | \mathbf{x}_{k-N}^{k-1}, e_{k-1}, e_k) \\ &\propto \exp \left\{ -\frac{[|\mathbf{x}_k|^2 + |\mathbf{c}_k|^2]}{2\sigma^2} \right\} \\ &\cdot \frac{\text{I}_0 \left(\frac{1}{\sigma^2} \left| \sum_{i=0}^N \mathbf{x}_{k-i} \mathbf{c}_{k-i}^H \right| \right)}{\text{I}_0 \left(\frac{1}{\sigma^2} \left| \sum_{j=1}^N \mathbf{x}_{k-j} \mathbf{c}_{k-j}^H \right| \right)} \end{aligned} \quad (25)$$

where $[\cdot]^H$ is the transpose conjugate operator.

By defining a reduced state as $s_k = (a_{k-1}, \dots, a_{k-Q+1}, \mu_{k-Q+1})$, with $Q < N$, a transition in the reduced-state trellis becomes $\epsilon_k \equiv (\mathbf{c}_{k-Q+1}, \dots, \mathbf{c}_k)$. Hence, the probability density function γ_k can be expressed as

$$\begin{aligned} \gamma_k(E_{k-1}^{(N-Q)}(\epsilon_k), \epsilon_k) &= p(\mathbf{x}_{k-N+1}^k | E_{k-1}^{(N-Q)}(\epsilon_k), \epsilon_k) \\ &\propto \exp \left\{ -\frac{1}{2\sigma^2} \sum_{i=0}^{Q-1} [|\mathbf{x}_{k-i}|^2 + |\mathbf{c}_{k-i}|^2] \right\} \\ &\cdot \exp \left\{ -\frac{1}{2\sigma^2} \sum_{j=Q}^{N-1} [|\mathbf{x}_{k-j}|^2 + |\hat{\mathbf{c}}_{k-j}|^2] \right\} \\ &\cdot \text{I}_0 \left(\frac{1}{\sigma^2} \left| \sum_{i=0}^{Q-1} \mathbf{x}_{k-i} \mathbf{c}_{k-i}^H + \sum_{j=Q}^{N-1} \mathbf{x}_{k-j} \hat{\mathbf{c}}_{k-j}^H \right| \right) \end{aligned} \quad (26)$$

and (8) specializes as follows

$$P\{a_k | \mathbf{x}_1^K\} = P\{a_k\} \sum_{\epsilon_k: a(\epsilon_k)=a_k} \gamma_k(E_{k-1}^{(N-Q)}(\epsilon_k), \epsilon_k) \cdot \alpha_k(\epsilon_k) \beta_k(\epsilon_k) \prod_{i=1}^{Q-1} P\{a_{k-i}\}. \quad (27)$$

The survivor map is built during the forward recursion and employed in the backward recursion and to evaluate γ_k in (26). Referring to the original formulation proposed in [6], the extension of the previously introduced general recursions is not immediate. Noting that $a^\circ(\epsilon_{k-1}) = a_{k-Q}$, the forward recursion on $\bar{\alpha}_k = \ln \alpha_k$ can be expressed as follows

$$\bar{\alpha}_k(\epsilon_k) \simeq \max_{a_{k-Q}} \{\bar{\alpha}_{k-1}(\epsilon_{k-1}) + \ln p(\mathbf{x}_{k-N}^k | a_{k-Q}, \epsilon_k) + \ln P\{a_{k-Q}\}\} - \ln p(\mathbf{x}_{k-N+1}^k | \epsilon_k). \quad (28)$$

Since at epoch k the survivor of each transition ϵ_{k-1} is known and $(a_{k-Q}, \epsilon_k) \equiv (\epsilon_{k-1}, \epsilon_k)$, we replace $p(\mathbf{x}_{k-N}^k | a_{k-Q}, \epsilon_k)$ with $p(\mathbf{x}_{k-N}^k | E_{k-2}^{(N-Q)}(\epsilon_{k-1}, \epsilon_{k-1}, \epsilon_k))$. The choice of the survivor associated with ϵ_k (that is, ϵ_{k-1}^{\max}) is based on the max operation. The term $\ln p(\mathbf{x}_{k-N+1}^k | \epsilon_k)$ affects the exact value of $\bar{\alpha}_k(\epsilon_k)$ but it does not affect the survivor selection. Instead of computing $\ln p(\mathbf{x}_{k-N+1}^k | \epsilon_k)$ we propose to utilize a_{k-Q}^{\max} determined through the max operation and replace it with

$$\begin{aligned} & \ln p(\mathbf{x}_{k-N+1}^k | E_{k-2}^{(N-Q-1)}(\epsilon_{k-1}^{\max}, \epsilon_{k-1}^{\max}, \epsilon_k)) \\ & \sim -\frac{1}{2\sigma^2} \sum_{i=0}^{Q-2} [|\mathbf{x}_{k-i}|^2 + |\mathbf{c}_{k-i}|^2] \\ & \quad + \frac{1}{2\sigma^2} \sum_{j=Q-1}^{N-1} [|\mathbf{x}_{k-j}|^2 + |\hat{\mathbf{c}}_{k-j}|^2] \\ & \quad + \ln I_0 \left(\frac{1}{\sigma^2} \left| \sum_{i=0}^{Q-2} \mathbf{x}_{k-i} \mathbf{c}_{k-i}^H + \sum_{j=Q-1}^{N-1} \mathbf{x}_{k-j} \hat{\mathbf{c}}_{k-j}^H \right| \right) \end{aligned} \quad (29)$$

where the expression $x \sim y$ denotes that x and y are monotonically related quantities. The forward recursion finally reduces to

$$\begin{aligned} \bar{\alpha}_k(\epsilon_k) & \simeq \bar{\alpha}_{k-1}(\epsilon_{k-1}^{\max}) \\ & \quad + \ln p(\mathbf{x}_{k-N}^k | E_{k-2}^{(N-Q)}(\epsilon_{k-1}^{\max}, \epsilon_{k-1}^{\max}, \epsilon_k)) \\ & \quad - \ln p(\mathbf{x}_{k-N+1}^k | E_{k-2}^{(N-Q-1)}(\epsilon_{k-1}^{\max}, \epsilon_{k-1}^{\max}, \epsilon_k)) \\ & \quad + \ln P\{a_{k-Q}\} \\ & \sim \bar{\alpha}_{k-1}(\epsilon_{k-1}^{\max}) + \frac{[|\mathbf{x}_{k-N}|^2 + |\hat{\mathbf{c}}_{k-N}|^2]}{2\sigma^2} \\ & \quad + \ln I_0 \left(\frac{1}{\sigma^2} \left| \sum_{i=0}^{Q-2} \mathbf{x}_{k-i} \mathbf{c}_{k-i}^H + \sum_{j=Q-1}^N \mathbf{x}_{k-j} \hat{\mathbf{c}}_{k-j}^H \right| \right) \\ & \quad - \ln I_0 \left(\frac{1}{\sigma^2} \left| \sum_{i=0}^{Q-2} \mathbf{x}_{k-i} \mathbf{c}_{k-i}^H + \sum_{j=Q-1}^{N-1} \mathbf{x}_{k-j} \hat{\mathbf{c}}_{k-j}^H \right| \right) \\ & \quad + \ln P\{a_{k-Q}\}. \end{aligned} \quad (30)$$

Once the survivor map has been determined during the forward recursion, the probability density function ϕ_k , used in the backward recursion, can be calculated

$$\begin{aligned} & \phi_k(E_{k-2}^{(N-Q)}(\epsilon_{k-1}, \epsilon_{k-1}, \epsilon_k)) \\ & = p(\mathbf{x}_k | \mathbf{x}_{k-N}^{k-1}, E_{k-2}^{(N-Q)}(\epsilon_{k-1}, \epsilon_{k-1}, \epsilon_k)) \\ & \propto \exp \left\{ -\frac{[|\mathbf{x}_k|^2 + |\mathbf{c}_k|^2]}{2\sigma^2} \right\} \\ & \quad \cdot \frac{I_0 \left(\frac{1}{\sigma^2} \left| \sum_{i=0}^{Q-1} \mathbf{x}_{k-i} \mathbf{c}_{k-i}^H + \sum_{j=Q}^N \mathbf{x}_{k-j} \hat{\mathbf{c}}_{k-j}^H \right| \right)}{I_0 \left(\frac{1}{\sigma^2} \left| \sum_{i=1}^{Q-1} \mathbf{x}_{k-i} \mathbf{c}_{k-i}^H + \sum_{j=Q}^N \mathbf{x}_{k-j} \hat{\mathbf{c}}_{k-j}^H \right| \right)}. \end{aligned} \quad (31)$$

C. Fading Channels

We consider the transmission of differentially encoded M -ary phase shift keying (M -PSK) over a Rayleigh flat-fading channel. We refer to the transmission system considered in [39] and denote by $f_D T$ the normalized fading rate of the channel. We assume that each information symbol a_k corresponds to a group of $\log_2 M = \tau$ bits, i.e., $a_k = (a_k^{(1)}, \dots, a_k^{(\tau)})$. The M -PSK symbol a_k is obtained by Gray-mapping the bits $(a_k^{(1)}, \dots, a_k^{(\tau)})$. The differentially encoded M -ary symbols c_k are defined by the rule $c_k = a_k c_{k-1}$, with $c_0 = 1$ (c_0 acts as a reference symbol). The corresponding received signal at the matched filter output can be written as $x_k = h_k c_k + w_k$, where the channel coefficients $\{h_k\}$ are complex random variables whose quadrature components are independent, Gaussian with zero mean, and $\{w_k\}$ are samples of a zero mean complex-valued white Gaussian noise process. The autocorrelation of the fading process is assumed to follow the classical isotropic scattering model [40].

Assuming that the information bits are independent within each symbol, we can consider $P\{a_k\} = P\{a_k^{(1)}\} \dots P\{a_k^{(\tau)}\}$.⁵ The proposed formulation gives the *a posteriori* probability of the M -PSK information symbols $\{a_k\}$. If the interleaver works bit-wise, as in [39], *a posteriori* probabilities of bits $\{a_k^{(i)}\}$ have to be calculated. To obtain these probabilities (1) can be modified as follows

$$\begin{aligned} & P\{a_k^{(i)} = j | \mathbf{x}_1^K\} \\ & = P\{a_k^{(i)} = j\} \sum_{\epsilon_k: a(\epsilon_k)^{(i)}=j} \gamma_k(\epsilon_k) \alpha_k(S^-(\epsilon_k)) \\ & \quad \cdot \beta_k(S^+(\epsilon_k)) P\{S^-(\epsilon_k)\} \prod_{l \neq i} P\{a_k^{(l)}\} \end{aligned} \quad (32)$$

for $j \in \{0, 1\}$. Since the channel coefficients $\{h_k\}$ are unknown in practice, linear prediction [41], [42] can be used. The branch metric $\bar{\gamma}_k(\epsilon_k) = \ln \gamma_k(\epsilon_k)$ becomes [39]

$$\begin{aligned} \bar{\gamma}_k(\epsilon_k) & \sim -\frac{|x_k - \hat{h}_k c_k|^2}{2\sigma^2} \\ & = -\frac{\left| x_k - c_k \sum_{n=1}^{\nu} p_n x_{k-n} c_{k-n}^* \right|^2}{2\sigma_{\text{pre}}^2} \end{aligned} \quad (33)$$

⁵In the case of an iterative decoding process, where $P\{a_k^{(i)}\}$ are derived from input extrinsic information, this is an approximation.

where \hat{h}_k is the fading gain estimate, ν is the prediction order, $\{p_n\}$ are the prediction coefficients and σ_{pre}^2 is the variance of the prediction error, which can be computed as shown in [43]. We may observe that $c_k c_{k-i}^* = \prod_{j=0}^{i-1} a_{k-j}$. By defining $S_k = (a_{k-\nu+1}, \dots, a_{k-1})$, the corresponding trellis consists of $M^{\nu-1}$ states. Assuming that the autocorrelation coefficients of the channel fading process are known, the optimal prediction coefficients p_n , $1 \leq n \leq \nu$ are obtained by solving the Wiener-Hopf equations [39]. Hence

$$\begin{aligned} \bar{\gamma}_k(e_k) &\sim -\frac{\left| x_k - \sum_{n=1}^{\nu} p_n x_{k-n} \prod_{j=0}^{n-1} a_{k-j} \right|^2}{2\sigma_{\text{pre}}^2} \\ &= -\frac{|x_k|^2 + |a_k|^2 \left| \sum_{n=1}^{\nu} p_n x_{k-n} \prod_{j=1}^{n-1} a_{k-j} \right|^2}{2\sigma_{\text{pre}}^2} \\ &\quad + \frac{\Re \left\{ x_k^* a_k \sum_{n=1}^{\nu} p_n x_{k-n} \prod_{j=1}^{n-1} a_{k-j} \right\}}{\sigma_{\text{pre}}^2}. \end{aligned} \quad (34)$$

Noting that $|a_k|^2$ and $|x_k|^2$ do not depend on e_k and defining

$$g(S_k) \triangleq \sum_{n=1}^{\nu} p_n x_{k-n} \prod_{j=1}^{n-1} a_{k-j}$$

we have

$$\bar{\gamma}_k(e_k) \sim -\frac{|g(S^-(e_k))|^2}{2\sigma_{\text{pre}}^2} + \frac{\Re\{x_k^* a_k g(S^-(e_k))\}}{\sigma_{\text{pre}}^2}. \quad (35)$$

The first term in (35) depends on $S^-(e_k)$ and should not be neglected, unless one assumes perfect channel estimation.

To reduce the number of trellis states, we proceed as in the case of an ISI channel. We may define a reduced state $s_k = (a_{k-Q+1}, \dots, a_{k-1})$, with $Q < \nu$, obtaining (the formalization is identical to that used in Section IV-A)

$$\begin{aligned} &g(E_{k-1}^{(\nu-Q)}(s^-(\epsilon_k)), s^-(\epsilon_k)) \\ &= \sum_{n=1}^{\nu} p_n x_{k-n} \prod_{j=1}^{n-1} a_{k-j} + \sum_{n=Q+1}^{\nu} p_n x_{k-n} \prod_{j=1}^{n-1} \hat{a}_{k-j} \\ \bar{\gamma}_k(E_{k-1}^{(\nu-Q)}(s^-(\epsilon_k)), \epsilon_k) \\ &\sim -\frac{|g(E_{k-1}^{(\nu-Q)}(s^-(\epsilon_k)), s^-(\epsilon_k))|^2}{2\sigma_{\text{pre}}^2} \\ &\quad + \frac{\Re\{x_k^* a_k g(E_{k-1}^{(\nu-Q)}(s^-(\epsilon_k)), s^-(\epsilon_k))\}}{\sigma_{\text{pre}}^2} \\ &\quad P\{s_k\} = \prod_{i=1}^{Q-1} \prod_{j=1}^{\tau} P\{a_{k-i}^{(j)}\}. \end{aligned} \quad (36)$$

V. NUMERICAL RESULTS

The performance of receivers based on reduced-state BCJR-type algorithms in the three cases considered in Section IV is assessed by means of computer simulations in terms of bit error

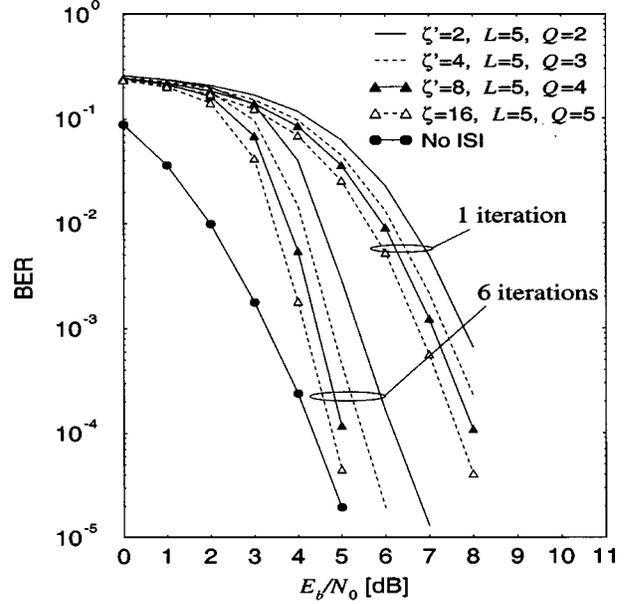


Fig. 5. Application of the proposed technique to iterative decoding/detection over an ISI channel. Receivers with various levels of complexity are considered and compared with the full-state receiver ($\zeta = 16$). The considered numbers of iterations are one and six in all cases. The performance in the case of coded transmission over an AWGN channel, without ISI, is also shown (solid lines with circles).

rate (BER) versus E_b/N_0 , E_b being the received signal energy per information bit and N_0 being the one-sided noise power spectral density. In any component decoder we consider the second simplified logarithmic version of the BCJR-type algorithm proposed in Section III (hybrid method).

In the case of coherent detection over an ISI channel, we consider the scheme of turbo detection in [35]. More precisely, we consider a binary ($M = 2$) transmission system characterized by a rate 1/2 16-state recursive systematic convolutional (RSC) encoder with generators $G_1 = 23$, $G_2 = 35$ (octal notation), followed by a 64×64 nonuniform interleaver. The bits at the output of the interleaver are sent through the channel by a binary PSK (BPSK) modulation. The minimum phase discrete-time channel impulse response is identified by the following coefficients: $f_0 = \sqrt{0.45}$, $f_1 = \sqrt{0.25}$, $f_2 = \sqrt{0.15}$, $f_3 = \sqrt{0.10}$ and $f_4 = \sqrt{0.05}$ [35]. The receiver is based on a serial concatenation of a detector, which uses the reduced-state BCJR algorithm with metrics proposed in Section IV-A, and a decoder which is a SISO module [36]. The extrinsic information is used according to the heuristic method proposed in [44]. By trial and error, we found that a good performance is obtained when the extrinsic information generated by the inner detector is weighted (i.e., multiplied) by a parameter equal to 0.3 and the extrinsic information generated by the outer decoder is weighted by a parameter equal to 0.5. The state reduction technique is applied to the inner detector. In Fig. 5, the performance of the full-state receiver (inner detector with $\zeta = 16$ states) is compared with the performance of the receiver with reduced complexity (inner detector with $\zeta' = 8, 4$, or 2 states). In all cases, we consider one and six decoding iterations. At six decoding iterations, the performance loss for a detector with $\zeta' = 4$ states with respect to

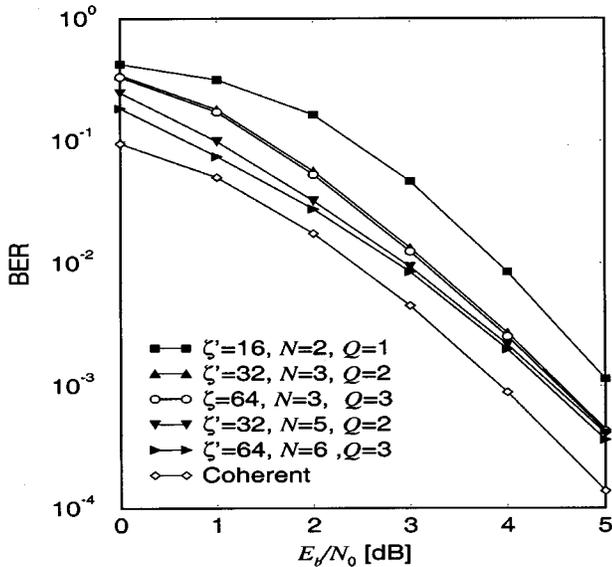


Fig. 6. Application of the proposed technique to noncoherent decoding of an RSC code. Receivers with various levels of complexity are considered and compared with a full-state receiver (with $N = 3$) and the coherent receiver.

the receiver without state reduction is only 0.75 dB at a BER of 10^{-4} , and it reduces to 0.25 dB for a detector with $\zeta' = 8$ states. For comparison, the performance in the absence of ISI, that is for coded transmissions over an AWGN channel, is also shown. In this case, the receiver reduces to the outer decoder of the RSC code considered above.

In the case of noncoherent decoding, we first consider a single 16-state RSC binary code with generators $G_1 = 37$, $G_2 = 21$ and rate $2/3$ obtained by means of puncturing, used as component of the turbo code presented in [7]. The modulation format is BPSK. In Fig. 6 the performance of the noncoherent decoder using the BCJR-type algorithm with metrics described in Section IV-B is assessed for various levels of state reduction and compared to that of a coherent receiver. With respect to a decoder with $N = 3$ and $\zeta = 64$ states (full complexity), for $N = 6$ and $\zeta' = 64$ states the performance is appreciably improved at low signal-to-noise ratios (SNRs). For $N = 2$ and $\zeta' = 16$, the performance loss with respect to the full-state receiver with $N = 3$ is less than one dB for every SNR, and reduces to only 0.5 dB at SNRs above five dB.

We also consider noncoherent decoding of a turbo code having as component codes those described in the previous paragraph [7]. The two component encoders are parallelly concatenated by a 32×32 nonuniform interleaver. At the receiver, each noncoherent component decoder uses the reduced-state BCJR-type algorithm described in Section IV-B. In this case too, the extrinsic information generated by each component decoder is passed to the other one by following the heuristic method proposed in [44]. By trial and error, we found that the best performance is obtained when the extrinsic informations generated by both the component decoders are weighted by a parameter equal to 0.3. Fig. 7 shows the performance of the receiver for various levels of state reduction, specified by the values of N and ζ' , and compares it to that of a coherent receiver and a noncoherent receiver with $N = Q = 3$ and $\zeta = 64$

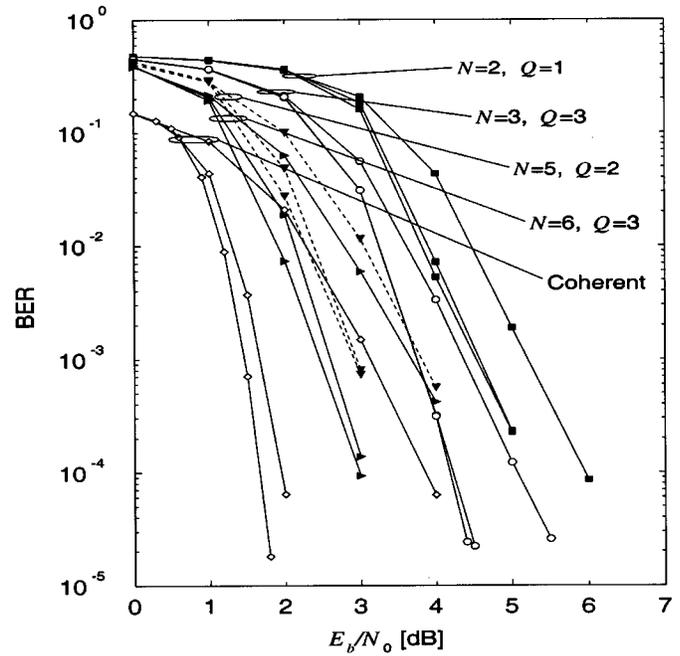


Fig. 7. Application of the proposed technique to noncoherent decoding of a turbo code. Receivers with various levels of complexity are considered and compared with a full-state receiver (with $N = 3$) and a coherent receiver. The considered numbers of iterations are 1, 3, and 6 in all cases.

(full number of states). In all cases, the considered numbers of iterations are 1, 3, and 6. The reduced-state BCJR-type algorithm with $N = 2$ and $\zeta' = 16$ ($Q = 1$) exhibits a performance loss of about 3.2 dB at six decoding iterations with respect to the coherent receiver. For $N = 6$ and $\zeta' = 64$ ($Q = 3$), at a BER of 10^{-4} a performance gain of about 1.2 dB is obtained with respect to the full-state case with $N = Q = 3$ and the same number of states ($\zeta = 64$). We conclude that the use of the proposed state reduction technique makes the algorithm proposed in [6] applicable, even for large values of phase memory N .

Finally, we consider transmission of differentially encoded quaternary PSK (QPSK) signals over a flat-fading channel, as in [39]. The outer code is a 64-state nonrecursive convolutional code with generators $G_1 = 133$ and $G_2 = 171$. This code is concatenated, through a 64×64 nonuniform bit interleaver, with an inner differential encoder. In fact, bit interleaving is an appropriate means to combat the effects of fading [45], [46]. The normalized fading rate is $f_D T = 0.01$. The differential inner detector uses linear prediction and the state reduction technique. In Fig. 8 the performance of the full-state receiver, with prediction order $\nu = Q = 3$ and 4, respectively, is compared with the performance of a receiver with various levels of complexity specified by the couple of parameters (ν, Q) . In all cases, we consider one and six decoding iterations. At six decoding iterations, a detector with $\nu = 3$ and $Q = 2$ exhibits a performance loss of only 0.2 dB at a BER of 10^{-4} , with respect to a full-state one ($\nu = Q = 3$). For $\nu = 5$ and $Q = 3$, the performance gain with respect to a full-state receiver with $\nu = 3$ is about 0.6 dB at a BER of 10^{-4} . For comparison, the performance curve in the case of perfect knowledge of the fading coefficients (coherent decoding) is also shown.

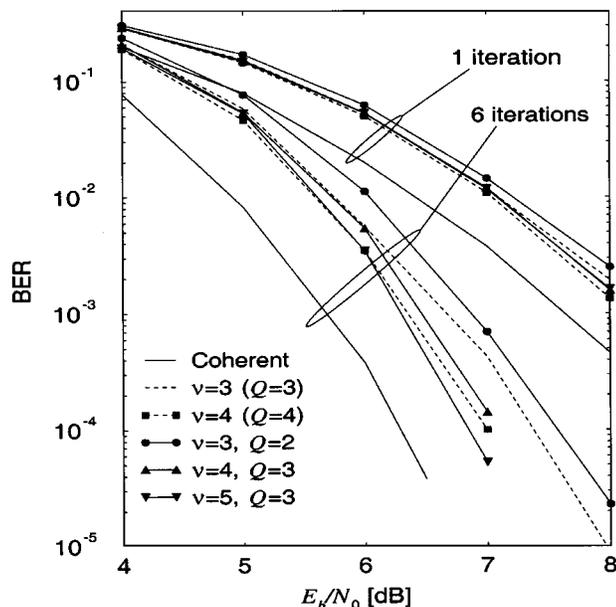


Fig. 8. Application of the proposed technique to iterative decoding/detection, through linear prediction, for flat-fading channel. Receivers with various levels of complexity (in terms of prediction order ν and reduced state parameter Q) are shown. The considered numbers of iterations are one and six in all cases. The performance in the case of decoding with perfect knowledge of the fading coefficients is also shown (solid lines).

VI. CONCLUSION

A class of BCJR-type algorithms has been defined as an extension of the well-known BCJR algorithm, whenever the channel memory is not finite or partially taken into account. Similarly to the BCJR algorithm, the proposed algorithms run a forward and a backward recursion. Techniques for state-complexity reduction for these algorithms have been introduced based on reduced-state sequence detection. The structure of the proposed BCJR-type algorithms leads to the definition of a survivor associated with a trellis branch (instead of a state). Depending on the structure of the system memory, a survivor map can be determined during one of the recursions and used in the other one. Alternatively, two survivor maps can be computed (one for each recursion) and suitably combined when generating *a posteriori* probabilities. A reduced-state BCJR-type algorithm is well suited for iterative processing applications. In the considered examples, the proposed reduced-state BCJR-type algorithms are effective for appreciably limiting the receiver complexity with minor performance degradation or improving performance for a given level of complexity.

APPENDIX

The considered formulation of a BCJR-type algorithm differs from that of the classical BCJR algorithm [2], although it reduces to the latter when the channel is strictly finite-memory. In this Appendix, we show that two alternative equivalent formulations of a BCJR-type algorithm exist, but none coincides exactly with the classical one.

Let us define an augmented “super-state” \check{S}_k in a one-to-one correspondence with a transition e_k . The number of states of this augmented trellis diagram is M times that of the original

one because information symbol a_k now appears in the state definition. In this augmented trellis, we may identify a transition as $\check{e}_k = (\check{S}_{k-1}, \check{S}_k)$. With these definitions, we may rewrite recursion (2) as

$$\alpha_k(\check{S}_k) = \sum_{\check{e}_k: S^+(\check{e}_k)=S_k} \psi_k(\check{e}_k) \alpha_{k-1}(\check{S}^-(\check{e}_k)) P\{\check{a}(\check{e}_k)\} \quad (37)$$

where $\check{S}^-(\check{e}_k)$ and $\check{S}^+(\check{e}_k)$ are the beginning and ending states of transition \check{e}_k , respectively, and the sum is extended over all transitions ending in S_k . The forward recursion (37) now reduces to that of the classical BCJR algorithm, except for the more general use of the probability density function $\psi_k(\check{e}_k)$. A similar conclusion also holds for the backward recursion. However, let us consider the “central” combination (1), which gives the *a posteriori* probability of an information symbol a_k . Unlike a classical BCJR algorithm, if this combination is expressed in terms of states and transitions in the augmented trellis, the following consequences arise: 1) the probability density functions α_k and β_k depend on states relative to the same time instant and 2) the sum must be extended over $1/M$ of the possible super-states $\{\check{S}_k\}$ at epoch k , that is over all states characterized by the considered symbol a_k (and not over all states). Therefore, even if this alternative formulation would entail a forward recursion in a form similar to that of a usual BCJR algorithm, it would still exhibit differences in this central combination.

In a second alternative formulation, forward and backward recursions may be computed on the augmented trellis, whereas the central combination may be computed on the original trellis. In this case, the BCJR-type algorithm is similar to a classical BCJR but the two recursions and the central combination must be referred to two different trellis representations in order to consider a single step as involving a single transition. Specifically, in the forward and backward recursions an augmented trellis must be used with M times the number of states of the trellis relative to the central combination.

ACKNOWLEDGMENT

References [24]–[29], [31], and [33] were brought to the authors’ attention during the review process. Three anonymous reviewers and W. H. Gerstacker are gratefully acknowledged.

REFERENCES

- [1] K. Abend and B. Fritchman, “Statistical detection for communications channels with intersymbol interference,” *Proc. IEEE*, vol. 58, pp. 779–785, May 1970.
- [2] L. R. Bahl, J. Cocke, F. Jelinek, and R. Raviv, “Optimal decoding of linear codes for minimizing symbol error rate,” *IEEE Trans. Inform. Theory*, vol. 20, pp. 284–284, Mar. 1974.
- [3] G. Battail, “Pondération des symboles décodé par l’algorithme de Viterbi,” *Ann. Télécommun., Fr.*, vol. 42, no. 1–2, pp. 31–38, Jan. 1987.
- [4] J. Hagenauer and P. Hoher, “A Viterbi algorithm with soft-decision outputs and its applications,” in *Proc. IEEE Global Telecom. Conf.*, Dallas, TX, Nov. 1989, pp. 1680–1686.
- [5] Y. Li, B. Vucetic, and Y. Sato, “Optimum soft-output detection for channels with intersymbol interference,” *IEEE Trans. Inform. Theory*, vol. 41, pp. 704–713, May 1995.
- [6] G. Colavolpe, G. Ferrari, and R. Raheli, “Noncoherent iterative (turbo) decoding,” *IEEE Trans. Commun.*, vol. 48, pp. 1488–1498, Sept. 2000.
- [7] C. Berrou and A. Glavieux, “Near optimum error correcting coding and decoding: Turbo-codes,” *IEEE Trans. Commun.*, vol. 44, pp. 1261–1271, Oct. 1996.

- [8] 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.
- [9] P. Robertson, E. Villerbrun, and P. Hoeher, "A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain," in *Proc. Int. Conf. Communications*, Seattle, WA, June 1995, pp. 1009–1013.
- [10] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "Soft-output algorithms for continuous decoding of parallel concatenated convolutional codes," in *Proc. Int. Conf. Communications*, Dallas, TX, June 1996, pp. 112–127.
- [11] K. M. Chugg and X. Chen, "Efficient architectures for soft-output algorithms," in *Proc. Int. Conf. Communications*, Atlanta, GA, June 1998, pp. 126–130.
- [12] K. M. Chugg, "The condition for applicability of the Viterbi algorithm with implications for fading channel MLSD," *IEEE Trans. Commun.*, vol. 46, pp. 1112–1116, Sept. 1998.
- [13] X. Yu and S. Pasupathy, "Innovations-based MLSE for Rayleigh fading channels," *IEEE Trans. Commun.*, vol. 43, pp. 1534–1544, Feb. 1995.
- [14] 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.
- [15] B. J. Frey and F. R. Kschischang, "Early detection and trellis splicing: Reduced-complexity iterative decoding," *IEEE J. Select. Areas Commun.*, vol. 16, pp. 153–159, Feb. 1998.
- [16] G. Colavolpe and R. Raheli, "Noncoherent sequence detection," *IEEE Trans. Commun.*, vol. 47, pp. 1376–1385, Sept. 1999.
- [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, Jan. 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] R. Raheli, A. Polydoros, and C.-K. Tzou, "Per-survivor processing: A general approach to MLSE in uncertain environments," *IEEE Trans. Commun.*, vol. 43, pp. 354–364, Feb. 1995.
- [21] A. Anastasopoulos and A. Polydoros, "Adaptive soft-decision algorithms for mobile fading channels," *Eur. Trans. Telecommun.*, vol. 9, pp. 183–190, Mar. 1998.
- [22] A. Anastasopoulos and K. M. Chugg, "Adaptive SISO algorithms for iterative detection with parametric uncertainty," in *Proc. ICC Communications Theory Mini-Conf.*, Vancouver, BC, Canada, June 1999, pp. 177–181.
- [23] A. Anastasopoulos and K. M. Chugg, "Adaptive iterative detection for turbo codes with carrier phase uncertainty," in *Proc. Global Telecommunications Conf.*, Rio de Janeiro, Brazil, Dec. 1999, pp. 2369–2374.
- [24] W. Koch and A. Baier, "Optimum and sub-optimum detection of coded data disturbed by time-varying intersymbol interference," in *Proc. Global Telecommunications Conf.*, San Diego, CA, Dec. 1990, pp. 1679–1684.
- [25] P. Hoeher, "TCM on frequency-selective fading channels: A comparison of soft-output probabilistic equalizers," in *Proc. Global Telecommunications Conf.*, San Diego, CA, Dec. 1990, pp. 1679–1684.
- [26] S. H. Müller, W. H. Gerstacker, and J. B. Huber, "Reduced-state soft-output trellis-equalization incorporating soft feedback," in *Proc. Global Telecommunications Conf.*, London, U.K., Dec. 1996, pp. 95–100.
- [27] G.-K. Lee, S. B. Gelfand, and M. P. Fitz, "Bayesian decision feedback techniques for deconvolution," in *Proc. Global Telecommunications Conf.*, San Francisco, CA, Dec. 1994, pp. 248–252.
- [28] M. P. Fitz and S. B. Gelfand, "Reduced complexity symbol-by-symbol soft output algorithms," in *Proc. Global Telecommunications Conf.*, London, U.K., Dec. 1996, pp. 21–25.
- [29] R. Balasubramanian, M. P. Fitz, and J. K. Krogmeier, "Optimal and suboptimal symbol-by-symbol demodulation of continuous phase modulated signals," *IEEE Trans. Commun.*, vol. 46, pp. 1662–1668, Dec. 1998.
- [30] L.-N. Lee, "Real-time minimal-bit-error probability decoding of convolutional codes," *IEEE Trans. Commun.*, vol. 22, pp. 146–151, Feb. 1974.
- [31] X. Chen and K. M. Chugg, "Reduced-state soft-input/soft-output algorithms for complexity reduction in iterative and noniterative data detection," in *Proc. Int. Conf. Communications*, New Orleans, LA, June 2000, pp. 6–10.
- [32] G. Colavolpe, G. Ferrari, and R. Raheli, "Reduced-state BCJR-type algorithms," in *Proc. IEEE Int. Conf. Communications*, New Orleans, LA, June 2000, pp. 460–464.
- [33] J. Heo, K. M. Chugg, and A. Anastasopoulos, "A comparison of forward-only and bi-directional fixed-lag adaptive SISOs," in *Proc. Int. Conf. Communications*, New Orleans, LA, June 2000, pp. 1660–1664.
- [34] G. D. Forney, Jr., "Maximum-likelihood sequence estimation of digital sequence in the presence of intersymbol interference," *IEEE Trans. Inform. Theory*, vol. IT-18, pp. 363–378, May 1972.
- [35] C. Douillard, M. Jesequel, C. Berrou, A. Picart, P. Didier, and A. Glavieux, "Iterative correction of intersymbol interference: Turbo-equalization," *Eur. Trans. Telecommun.*, vol. 6, pp. 507–511, Sept. 1995.
- [36] 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.
- [37] S. Ariyavisitakul, "A decision feedback equalizer with time-reversal structure," *IEEE J. Select. Areas Commun.*, vol. 10, pp. 599–613, Apr. 1992.
- [38] N. C. McGinty and R. A. Kennedy, "Reduced-state sequence estimator with reverse-time structure," *IEEE J. Select. Areas Commun.*, vol. 45, pp. 265–268, Mar. 1997.
- [39] P. Hoeher and J. Lodge, "'Turbo-DPSK': Iterative differential PSK demodulation and channel decoding," *IEEE Trans. Commun.*, vol. 47, pp. 837–843, June 1999.
- [40] R. Clarke, "A statistical theory of mobile-radio reception," *Bell Syst. Tech. J.*, vol. 47, pp. 957–1000, July 1968.
- [41] J. Makhoul, "Linear prediction: A tutorial review," *Proc. IEEE*, vol. 63, pp. 561–580, Apr. 1975.
- [42] D. Makrakis, P. Mathiopoulos, and D. Bouras, "Optimal decoding of coded PSK and QAM signals in correlated fast fading channels and AWGN: A combined envelope, multiple differential and coherent detection approach," *IEEE Trans. Commun.*, vol. 42, pp. 837–843, Jan. 1994.
- [43] G. Colavolpe, P. Castoldi, and R. Raheli, "Linear predictive receivers for fading channels," *IEE Electron. Lett.*, vol. 34, pp. 1289–1290, June 1998.
- [44] G. Colavolpe, G. Ferrari, and R. Raheli, "Extrinsic information in turbo decoding: A unified view," in *Proc. Global Telecommunications Conf.*, Rio de Janeiro, Brazil, Dec. 1999, pp. 505–509.
- [45] G. Caire, G. Taricco, and E. Biglieri, "Bit-interleaved coded modulation," *IEEE Trans. Inform. Theory*, vol. 44, pp. 927–946, May 1999.
- [46] X. Li and J. A. Ritcey, "Bit-interleaved coded modulation with iterative decoding," *IEEE Commun. Lett.*, vol. 1, pp. 169–171, Nov. 1997.



Giulio Colavolpe (S'97–A'01) was born in Cosenza, Italy, in 1969. He received the Dr.Eng. degree in telecommunications engineering (cum laude) from the University of Pisa, Italy, in 1994 and Ph.D. degree in information technology from the University of Parma, Italy, in 1998.

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



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

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

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



Riccardo Raheli (M'87) received the Dr.Ing. degree (Laurea) in electrical engineering (summa cum laude) from the University of Pisa, Italy, in 1983, the M. Sci. degree in electrical and computer engineering from the University of Massachusetts, Amherst, in 1986, and the Ph.D. 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 with Siemens Telecomunicazioni, Cassina de' Pecchi (Milan), Italy. From 1988 to 1991, he was a Research Professor with the Scuola Superiore di Studi Universitari e di Perfezionamento S. Anna, Pisa, Italy. In 1990, he was a Visiting Assistant with the University of Southern California, Los Angeles. Since 1991, he has been with the University of Parma, Italy, where he is currently an Associate Professor of Telecommunications. His scientific interests are in the general area of statistical communication theory, with special attention toward digital transmission systems, data-sequence detection techniques, digital signal processing and adaptive algorithms for telecommunications. His research activity has led to numerous scientific publications in leading international journals and conference proceedings and a few industrial patents.

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