Reduced-Complexity BCJR Algorithm for Turbo Equalization

Dario Fertonani, *Student Member, IEEE*, Alan Barbieri, and Giulio Colavolpe, *Member, IEEE*

***Abstract*—We propose novel techniques to reduce the complexity of the well-known Bahl, Cocke, Jelinek, and Raviv (BCJR) algorithm when it is employed as a detection algorithm in turbo equalization schemes. In particular, by also considering an alternative formulation of the BCJR algorithm, which is more suitable than the original one for deriving reduced-complexity techniques, we describe three reduced-complexity algorithms, each of them particularly effective over one of the three different classes of channels affected by intersymbol interference (minimum-phase, maximum-phase, and mixed-phase channels). The proposed algorithms do not explore all paths on the trellis describing the channel memory, but they work only on the most promising ones, chosen according to the maximum *a posteriori* criterion. Moreover, some optimization techniques improving the effectiveness of the proposed solutions are described. Finally, we report the results of computer simulations showing the impressive performance of the proposed algorithms, and we compare them with other solutions in the literature.**

***Index Terms*—Complexity reduction, intersymbol interference (ISI), maximum *a posteriori* (MAP) symbol detection, turbo equalization.**

I. Introduction

WE CONSIDER the algorithm by Bahl, Cocke, Jelinek, and Raviv (BCJR) [1], known to be optimal in implementing maximum *a posteriori* (MAP) *symbol* detection for channels with finite memory, and we propose new solutions to reduce its complexity. This paper focuses on turbo equalization schemes for channels affected by intersymbol interference (ISI) [2], [3], but the described techniques can also be adopted in other applications of the BCJR algorithm, such as the decoding of convolutional codes [1] and the MAP symbol detection over noncoherent channels [4] or flat fading channels [5].

The BCJR algorithm works on a trellis representing the finite-state machine that describes the channel, and its complexity is proportional to the number of trellis states. Since this number grows exponentially with the channel memory, it becomes often necessary to design low-complexity suboptimal algorithms ensuring a possibly negligible performance loss. The majority of the reduced-complexity algorithms in the literature maintains the three-stage structure of the BCJR algorithm (forward recursion, backward recursion, and completion stage), and obtains the complexity reduction by performing a simplified trellis search. The various suboptimal algorithms can be approximately classified based on two different rationales, which are described in the following.

In the first class, inspired by the reduced-state sequence detection (RSSD) [6]–[8] originally designed to reduce the complexity of the classical Viterbi algorithm (VA), we mention the reduced-state BCJR (RS-BCJR) algorithm [9] and the generalized reduced-state algorithms proposed in [10]. They exploit the fact that the forward and backward recursions of the BCJR algorithm reduce to the VA when the *max-log* approximation [11] is adopted. Hence, the concept of survivor results. Then, assuming that only a part of the information corresponding to the full state is embedded in a properly defined reduced state, they recover the missing information by decision feedback, in a way similar to the RSSD. The RS-BCJR algorithm is particularly effective on minimum-phase channels. This algorithm generally provides a high-quality soft output. For this reason, when employed in turbo equalization schemes, it ensures a good convergence of the iterative process. The algorithms presented in [10], which extend the concept of state reduction, achieve a good performance on a larger subset of ISI channels, but the corresponding performance loss with respect to the full-state BCJR algorithm increases when they are employed in turbo equalization schemes, because of the poor quality of the generated soft output.

The algorithms in the second class perform a reduced search on the original full-complexity trellis, instead of a full search on a reduced-state trellis. The M-BCJR algorithm [12] belongs to this class. It provides a good performance/complexity tradeoff on a large subset of ISI channels, but it is not effective in producing a high-quality soft output [13]. As a consequence of the fact that the M-BCJR algorithm gives a predominant role to the forward recursion, since only the trellis paths selected during this stage are explored in the backward recursion, it cannot cope with maximum-phase or mixed-phase channels. A couple of solutions to this problem, based on an independent trellis search in the backward recursion, were proposed in [14] and [15].

Among the reduced-complexity algorithms that do not belong to the described classes, we mention the algorithm described in [16], based on a confidence criterion used to detect reliable symbols early on during decoding, and the algorithm presented in [17], which addresses the particular case of sparse ISI channels, that is, channels with a number of nonzero interferers much lower than the channel memory.

We present three reduced-search algorithms, each of them particularly effective over one of the three different subsets into which the ISI channels can be partitioned, namely minimum-phase, maximum-phase, and mixed-phase channels

Paper approved by G. M. Vitetta, the Editor for Equalization and Fading Channels of the IEEE Communications Society. Manuscript received September 5, 2006; revised December 26, 2006 and March 7, 2007. This work was presented in part at the IEEE International Conference on Communications (ICC'06), Istanbul, Turkey, June 2006.

D. Fertonani A. Barbieri, and G. Colavolpe are with the Department of Information Engineering, University of Parma, 43100 Parma, Italy (e-mail: fertonani@tlc.unipr.it; barbieri@tlc.unipr.it; giulio@unipr.it).

Digital Object Identifier 10.1109/TCOMM.2007.910638

0090-6778/$25.00 © 2007 IEEE

[18]. In all the proposed solutions, while performing the reduced-complexity trellis searches, the paths to be explored are selected using the MAP criterion. We show that the classical formulation of the BCJR algorithm does not allow the MAP-based selection of the best paths during the backward recursion. Hence, we resort to an alternative formulation, showing that it is fundamental for deriving reduced-complexity algorithms for channels with mixed or maximum phase, whereas the traditional formulation is more effective for minimum-phase channels. The algorithms are further generalized by deriving a state-partitioning technique, which provides a better flexibility to different channels and modulation formats, and optimized by adopting a simple solution, thereby improving the quality of the produced extrinsic information [19].

The remainder of this paper is organized as follows. In Section II, we describe the system model and present two equivalent formulations of the BCJR algorithm. In Section III, the proposed reduced-complexity algorithms are described. In Section IV, we report the results of computer simulations comparing the performance of various detection algorithms. In Section V, some concluding remarks are drawn.

II. System Model

A. Turbo Equalization

In the considered transmission system, a sequence of M-ary complex-valued code symbols, obtained by the encoding of a sequence of information bits, is transmitted from epoch 0 to epoch $K-1$. These code symbols are permuted by a proper interleaver, and the resulting sequence $\mathbf{a} = \{a_k\}_{k=0}^{K-1}$ is linearly modulated and transmitted over an ISI channel that also introduces additive white Gaussian noise (AWGN). At the output of a whitened matched filter, assuming ideal synchronization, the received sample at time epoch k can be expressed as [20]

$$y_k = \sum_{i=0}^{L} f_i a_{k-i} + w_k \tag{1}$$

where $\{w_k\}_{k=0}^{K-1}$ are complex independent Gaussian random variables with mean zero and variance σ^2 per component, L is the channel memory, and $\boldsymbol{f} = \{f_i\}_{i=0}^{L}$ represents the discrete-time equivalent channel impulse response. We assume ideal knowledge on σ^2 and $\boldsymbol{f}$ (see [21], and references therein for the case of joint detection and channel estimation).

The receiver attempts to recover the information bits by means of the turbo equalization scheme depicted in Fig. 1. The key point of this scheme is the iterative exchange of information between the soft-input soft-output (SISO) modules forming the receiver [2], [3]. The focus of this paper is on the SISO detector, that is, a block that computes the *a posteriori* probability (APP) $P(a_k|\boldsymbol{y})$ of each modulation symbol a_k at each time epoch k, given the received sequence $\boldsymbol{y} = \{y_k\}_{k=0}^{K-1}$ and the *a priori* probabilities $\{P(a_k)\}_{k=0}^{K-1}$. With regard to each symbol a_k, the actual output of the detector is not $P(a_k|\boldsymbol{y})$, but the so-called extrinsic information $E_k(a_k) = P(a_k|\boldsymbol{y})/P(a_k)$. It is worth to notice that the presence of a proper interleaver allows the detector to consider the modulation symbols independent of each other, so that the factorization $P(\mathbf{a}) = \prod_{k=0}^{K-1} P(a_k)$ can be exploited to derive the detection algorithm. On the other hand, in this paper, the SISO decoder is simply considered as a block that, based on the extrinsic information provided by the detector, updates the probabilities of code symbols. These probabilities will be assumed as *a priori* probabilities by the SISO detector. To improve the reliability of the hard decisions on the information bits, the described process is iteratively repeated a number of times that strictly depends on the characteristics of both the code and the channel.

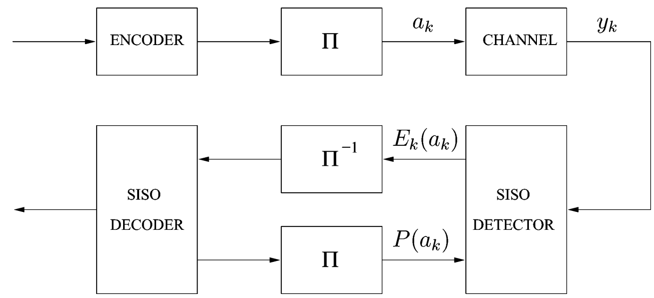

Fig. 1. Block diagram of a system employing turbo equalization.

B. BCJR Algorithm

An overview of the BCJR algorithm [1], which is known to be the optimal detection algorithm for the turbo equalization schemes, is given in the following. Together with the classical formulation of the algorithm, we also present an alternative formulation that will be exploited to derive the proposed low-complexity algorithms.

Let us define, at time epoch k, the state σ_k as

$$\sigma_k = (a_{k-L}, a_{k-L+1}, \ldots, a_{k-2}, a_{k-1}) \tag{2}$$

and the branch metric function as

$$F_k(a_k, \sigma_k) = P(a_k) \exp\left[-\frac{1}{2\sigma^2}\left|y_k - \sum_{i=0}^{L} f_i a_{k-i}\right|^2\right]. \tag{3}$$

The BCJR algorithm is characterized by the following forward and backward recursions:

$$\alpha_{k+1}(\sigma_{k+1}) = \sum_{a_k}\sum_{\sigma_k} T(a_k, \sigma_k, \sigma_{k+1}) F_k(a_k, \sigma_k) \alpha_k(\sigma_k) \tag{4}$$

$$\beta_k(\sigma_k) = \sum_{a_k}\sum_{\sigma_{k+1}} T(a_k, \sigma_k, \sigma_{k+1}) F_k(a_k, \sigma_k) \beta_{k+1}(\sigma_{k+1}) \tag{5}$$

where $T(a_k, \sigma_k, \sigma_{k+1})$ is the trellis indicator function equal to 1 if a_k, σ_k, and σ_{k+1} satisfy the trellis constraints and 0 otherwise. The state metrics $\alpha_k(\sigma_k)$ and $\beta_k(\sigma_k)$ have the following probabilistic meanings[1]:

$$\alpha_k(\sigma_k) \propto P(\sigma_k|\boldsymbol{y}_0^{k-1}) \tag{6}$$

$$\beta_k(\sigma_k) \propto p(\boldsymbol{y}_k^{K-1}|\sigma_k) \tag{7}$$

[1] In this paper, the proportionality symbol $\propto$ is used when two quantities differ for a positive multiplicative factor, irrelevant for the detection process.

where $\boldsymbol{y}_{k_1}^{k_2} = \{y_k\}_{k=k_1}^{k_2}$. The metrics $\alpha_0(\sigma_0)$ and $\beta_K(\sigma_K)$ are initialized according to the available information on the first state $\bar{\sigma}_0$ and the last state $\bar{\sigma}_K$. For example, when $\bar{\sigma}_0$ and $\bar{\sigma}_K$ are known to the receiver, $\alpha_0(\bar{\sigma}_0)$ and $\beta_K(\bar{\sigma}_K)$ can be set to 1 while the other metrics are initialized to 0. Finally, in the completion stage, the APP $P(a_k|\boldsymbol{y})$ is computed by marginalizing $P(\sigma_{k+1}|\boldsymbol{y})$ as

$$P(a_k|\boldsymbol{y}) = \sum_{\sigma_{k+1}} S(a_k, \sigma_{k+1}) P(\sigma_{k+1}|\boldsymbol{y}) \tag{8}$$

where the indicator function $S(a_k, \sigma_{k+1})$ is equal to 1 if σ_{k+1} is compatible with a_k and 0 otherwise. Simple manipulations of (6) and (7) prove that

$$P(\sigma_{k+1}|\boldsymbol{y}) \propto \alpha_{k+1}(\sigma_{k+1})\beta_{k+1}(\sigma_{k+1})\,. \tag{9}$$

Thus, the completion (8) becomes

$$P(a_k|\boldsymbol{y}) \propto \sum_{\sigma_{k+1}} S(a_k, \sigma_{k+1})\alpha_{k+1}(\sigma_{k+1})\beta_{k+1}(\sigma_{k+1})\,. \tag{10}$$

An alternative formulation of the BCJR algorithm can be derived by starting from a factorization of the APPs of the states different with respect to (9). If we give the following alternative definitions of the branch metric

$$\tilde{F}_k(a_k, \sigma_k) = P(a_{k-L}) \exp\left[-\frac{1}{2\sigma^2}\left|y_k - \sum_{i=0}^{L} f_i a_{k-i}\right|^2\right] \tag{11}$$

and of the recursions

$$\tilde{\alpha}_{k+1}(\sigma_{k+1}) = \sum_{a_k}\sum_{\sigma_k} T(a_k, \sigma_k, \sigma_{k+1})\tilde{F}_k(a_k, \sigma_k)\tilde{\alpha}_k(\sigma_k) \tag{12}$$

$$\tilde{\beta}_k(\sigma_k) = \sum_{a_k}\sum_{\sigma_{k+1}} T(a_k, \sigma_k, \sigma_{k+1})\tilde{F}_k(a_k, \sigma_k)\tilde{\beta}_{k+1}(\sigma_{k+1}) \tag{13}$$

it is straightforward to prove that, in this case, the state metrics have the following probabilistic meanings:

$$\tilde{\alpha}_k(\sigma_k) \propto p(\boldsymbol{y}_0^{k-1}|\sigma_k) \tag{14}$$

$$\tilde{\beta}_k(\sigma_k) \propto P(\sigma_k|\boldsymbol{y}_k^{K-1}) \tag{15}$$

so that

$$\tilde{\alpha}_k(\sigma_k)\tilde{\beta}_k(\sigma_k) \propto \alpha_k(\sigma_k)\beta_k(\sigma_k). \tag{16}$$

As in the classical BCJR algorithm, when $\bar{\sigma}_0$ and $\bar{\sigma}_K$ are known, the metrics $\tilde{\alpha}_0(\bar{\sigma}_0)$ and $\tilde{\beta}_K(\bar{\sigma}_K)$ are set to 1 while the others are initialized to 0. Finally, by combining (10) and (16), the completion can be written as

$$P(a_k|\boldsymbol{y}) \propto \sum_{\sigma_{k+1}} S(a_k, \sigma_{k+1})\tilde{\alpha}_{k+1}(\sigma_{k+1})\tilde{\beta}_{k+1}(\sigma_{k+1}). \tag{17}$$

By comparing (12) with (4), (13) with (5), and (17) with (10), it is clear that this alternative formulation does not modify, at all, the complexity of the algorithm, since the only difference is the presence of the term $P(a_{k-L})$ instead of $P(a_k)$ in the branch metric. Hence, the two formulations of the algorithm coincide only if the modulation symbols are equally likely.

The reduced-complexity algorithms described in the following exploit the probabilistic meanings of the state metrics; thus, it is worth to remark them. Let us focus on (6) and (7): the maximization of the metric $\alpha_k(\sigma_k)$ provides the selection, based on the first $k-1$ received samples, of the state σ_k according to the MAP criterion, while the maximization of the metric $\beta_k(\sigma_k)$ provides the selection, based on the remaining received samples, of the state σ_k according to the maximum-likelihood (ML) criterion. Vice versa, the relations (14) and (15) show that maximizing $\tilde{\alpha}_k(\sigma_k)$ provides an ML-based selection, while maximizing $\tilde{\beta}_k(\sigma_k)$ provides a MAP-based selection. This reverse symmetry motivates the choice for referring, hereafter, to the first formulation as *classical* and to the second as *reverse*.

We would like to mention that, although this paper focuses on a detection approach based on the whitened matched filter front end, this is not the only way to perform MAP *symbol* detection over ISI channels. The application of the proposed techniques for complexity reduction to a couple of alternative detection approaches, based on a different front end, is discussed in the Appendix.

III. Proposed Algorithms

A. Rationale

The number of trellis states can be considered as a measure of the complexity of the BCJR algorithm. Hence, for a fixed value of the frame length K, the complexity is proportional to M^L and grows exponentially with the channel memory. Long channel impulse responses and large modulation alphabets, thus, make the implementation of the BCJR algorithm impractical and motivate the search for suboptimal algorithms with a convenient performance/complexity tradeoff. To obtain reduced-complexity algorithms, we adopt the *reduced search* technique [12]: the algorithms still work on a full-state trellis, that is, no state reduction [9] is performed, but they explore only a subset of the possible paths on the trellis. Since the lower the metric (either forward or backward) of a state, the more negligible its contribution to the summations of the recursions and completion, it is natural to explore only the paths extending from the states with the largest metrics. Let us suppose to keep memory, at each time epoch, only of the S largest forward and the S largest backward metrics, and to explore only the paths extending from the related states while performing the recursions. In the state metrics computation, the contribution corresponding to unexplored paths is considered null. If we consider the number S of saved metrics as a measure of the complexity of the algorithms, the reduction factor with respect to the full-complexity BCJR is about M^L/S. More detailed complexity comparisons will be presented later.

By connecting the states whose forward metrics have been saved, a set of forward-selected paths (FSPs) is progressively built, and the same can be done in order to define a set of backward-selected paths (BSPs). As shown in Section II-B, the selection of the most promising paths can follow either a MAP or an ML criterion, depending on which formulation of the recursions is chosen between the classical and the reverse ones. These different criteria, which are equivalent only when the modulation symbols are equally likely, have a significant impact on the performance of the reduced-search algorithms since they generate potentially different sets of selected paths. It is intuitive to conjecture that the MAP approach is to be preferred, and extensive computer simulations confirm this fact. In order to combine the recursions providing MAP measures, that is, the classical forward recursion and the reverse backward recursion, the following completion can be adopted:

$$P(a_k|\boldsymbol{y}) \propto \sum_{\sigma_{k+1}} \frac{S(a_k, \sigma_{k+1})\alpha_{k+1}(\sigma_{k+1})\tilde{\beta}_{k+1}(\sigma_{k+1})}{P(\sigma_{k+1})} \quad (18)$$

where

$$P(\sigma_k) = \prod_{i=1}^{L} P(a_{k-i}) \; . \quad (19)$$

In this case, the computational complexity increases slightly with respect to traditional formulations as (10) or (17), because of the presence of the term $1/P(\sigma_{k+1})$ in the summations.

The values of the forward (respectively, backward) metrics related to states that do not belong to the set of FSPs (respectively, BSPs) are not available during the completion stage. The unavailable metrics are traditionally replaced by zero, thus neglecting the terms containing them while computing the summations. In this case, it is easy to prove that any state can give a nonzero contribution to the completion only if it belongs to both FSPs and BSPs, so that the effective trellis is given by the intersection of FSPs and BSPs. This heavily degrades the performance of the reduced-complexity algorithms when the sets of FSPs and BSPs are built independently of each other, since their intersection could be almost empty. For this reason, it is suggested in [12] not to build any set of BSPs and to perform the backward search over the set of FSPs, thus giving a predominant role to the forward recursion. In the next section, we will propose three different solutions, each of them being effective on a different subset of the whole set of channel types.

B. Description of the Algorithms

The ISI channels can be partitioned into three subsets, namely minimum-phase, maximum-phase, and mixed-phase channels, depending on the vector $\boldsymbol{f}$. The mathematical definition of the phase of a channel[2] implies that, in the case of minimum-phase channels, the channel energy is mainly located in the first taps [18]. As a consequence of this fact, the estimates of the APPs provided by the forward recursion are much more reliable than those provided by the backward recursion [15]. Hence, relatively to minimum-phase channels, the best choice is to perform the following sequence of steps:

1) classical forward recursion (4) selecting the FSPs;
2) classical backward recursion (5) over the FSPs;
3) classical completion (10) over the FSPs.

[2]The minimum-phase and the maximum-phase channels are defined in [18]. Here, all the remaining channels are referred to as the mixed-phase channels.

This solution, which will be referred to as *forward-trellis* (FT) algorithm, reduces to the M-BCJR algorithm [12] when none of the optimization techniques described in the next section is employed.

Vice versa, in the case of maximum-phase channels, the last taps contain the greatest part of the channel energy [18]; consequently, the most reliable estimates of the APPs are provided by the backward recursion [15]. Thus, it is convenient to perform the following sequence of steps:

1) reverse backward recursion (13) selecting the BSPs;
2) reverse forward recursion (12) over the BSPs;
3) reverse completion (17) over the BSPs.

This solution will be referred to as *backward-trellis* (BT) algorithm.

Finally, in the case of mixed-phase channels, no physical reason to privilege one recursion instead of the other exists, so that the best choice is to build the sets of FSPs and BSPs independently of each other. The resulting algorithm can be summarized by the following sequence of steps:

1) classical forward recursion (4) selecting the FSPs;
2) reverse backward recursion (13) selecting the BSPs;
3) completion (18) combining the FSPs and the BSPs.

In this case, the order of executing the recursions is irrelevant (they could be even implemented in parallel). This solution will be referred to as *double-trellis* (DT) algorithm. From a computational viewpoint, this algorithm causes a slight increase in complexity, since it requires to compare and sort the metrics in both the recursions, while the FT and BT algorithms require this procedure only in one recursion.

C. Optimization of the Algorithms

As stated before, a traditional completion replacing the unavailable metrics by null values works on a subset of paths given by the intersection of FSPs and BSPs. While in the case of the FT (respectively, BT) algorithm, the intersection coincides with the set of FSPs (respectively, BSPs), in the case of the DT algorithm, the intersection could even be empty, since the sets of FSPs and BSPs are built independently of each other. This issue is addressed in [14], where the authors propose a completion on a window of multiple trellis sections, thus implying a significant increase in the computational complexity of the completion stage. We propose a simpler solution allowing the completion stage to work not on the intersection but on the union of the sets of FSPs and BSPs, so that all the metrics saved during the recursions can give a contribution to the final result. As also discussed in [15] and [22], the way to do this consists of replacing the unavailable metrics in (18) by proper nonzero values. At every time epoch k, let α_k^{MIN} be the lowest metric saved during the classical forward recursion, and let $\tilde{\beta}_k^{\mathrm{MIN}}$ be the lowest metric saved during the reverse backward recursion. When a given state σ_{k+1} does not belong to the set of FSPs at time epoch $k+1$, any nonzero value lower than or equal to $\alpha_{k+1}^{\mathrm{MIN}}$ could be a reasonable choice for replacing the unavailable metric $\alpha_{k+1}(\sigma_{k+1})$ while performing the completion (18). Since we found, by means of extensive computer simulations, that overestimating the unavailable metrics provides a better performance than underestimating them, we choose the largest value in the allowed range. Hence, when the factor $\alpha_{k+1}(\sigma_{k+1})$ in (18) is not available, we replace it by $\alpha_{k+1}^{\mathrm{MIN}}$. Similarly, when $\tilde{\beta}_{k+1}(\sigma_{k+1})$ is not available, we replace it by $\tilde{\beta}_{k+1}^{\mathrm{MIN}}$. When both the factors are not available, we instead ignore the whole product $\alpha_{k+1}(\sigma_{k+1})\tilde{\beta}_{k+1}(\sigma_{k+1})$. In Section IV, it is shown that the increase in computational complexity due to this solution, which will be referred to as nonzero (NZ) completion, is not crucial.

Typically, even when they provide a hard output almost equal to the optimal one, the reduced-complexity algorithms estimate their hard decisions to be much more reliable than they really are. Extensive computer simulations show that even the proposed algorithms, especially the FT and BT ones, tend to overestimate the reliability of the decisions when the reduction factor M^L/S is large. Unfortunately, the fact that optimistic soft output is produced dramatically affects the convergence of turbo equalization. In [23], referring to the particular case of the soft-output VA (SOVA) [24], the authors conjecture that the low-quality soft output is due to the correlation between the intrinsic information $P(a_k)$ and the extrinsic information $E_k(a_k) = P(a_k|\boldsymbol{y})/P(a_k)$ generated by the SOVA, and propose to mitigate this effect by passing the extrinsic information through an adaptive attenuator. We adopt a much simpler solution that consists of saturating, for each time epoch k, the extrinsic information so that the ratio between the lowest extrinsic information and the largest one must be at least equal to γ, where γ is a proper parameter in the range $[0, 1]$. Hence, instead of the computed values $\{E_k(a_k)\}$, the modified values $\{\hat{E}_k(a_k)\}$ are fed to the SISO decoder, according to the definition

$$\hat{E}_k(a_k) = \max\{E_k(a_k), E_k^{\mathrm{S}}\} \tag{20}$$

where, for each time epoch k, the threshold value E_k^{S} is computed as

$$E_k^{\mathrm{S}} = \gamma \max_{a_k}\{E_k(a_k)\}\ . \tag{21}$$

The crucial point is the choice of the value of γ: when it is too low, the decoder is nearly forced to confirm the decisions of the detector, whereas, when it is too large, the information produced by the detection algorithm is practically destroyed. The optimization of the parameter γ, which can be performed by means of computer simulations, leads to values significantly depending on the signal-to-noise ratio and the reduction factor of interest. This *output saturation* (OS) practically does not increase the complexity of the algorithms, but, as shown in Section IV, it can provide a significant performance improvement.

Finally, we describe a generalization of the reduced search technique, which is based on a *state-partitioning* (SP) approach staying in the middle between the rationales of the M-BCJR algorithm, the RS-BCJR algorithm, and the algorithms based on decision-feedback sequence estimation (DFSE), as that proposed in [22]. This approach ensures a better flexibility to different channel types and can provide significant performance improvements, especially in the case of nonbinary modulation formats. While building the set of FSPs (that is, only in the case of the FT and DT algorithms), let us define

$$\sigma_k'' = (a_{k-Q_f}, a_{k-Q_f+1}, \ldots, a_{k-1}) \tag{22}$$

$$\sigma_k'' = (a_{k-L}, a_{k-L+1}, \ldots, a_{k-Q_f-1}) \tag{23}$$

Q_f being a nonnegative design parameter, such that $\sigma_k = (\sigma_k', \sigma_k'')$. We will adopt the notation *state of the recent symbols* for σ_k' and *state of the older symbols* for σ_k''. The set of FSPs is built by saving, for each possible value of σ_k', only the states σ_k'' that give the S'' best metrics $\alpha_k(\sigma_k)$. Hence, the saved metrics per time epoch are $S_f = M^{Q_f} S''$, and unlike the case of the basic algorithms presented before, they are not necessarily the S_f highest ones. By expanding the factor $P(\sigma_k|\boldsymbol{y}_0^{k-1})$ in (6) as

$$P(\sigma_k|\boldsymbol{y}_0^{k-1}) = P(\sigma_k'|\boldsymbol{y}_0^{k-1})P(\sigma_k''|\sigma_k', \boldsymbol{y}_0^{k-1}) \tag{24}$$

the rationale of the proposed process can be understood: conditioning to each combination of the recent symbols σ_k', we save only the S'' highest APPs of the older symbols σ_k''. Hence, the set of FSPs is defined as maintaining complete information on the state of the recent symbols and partial information on the state of the older symbols. While building the set of BSPs (that is, only in the case of the BT and DT algorithms), we adopt a different partitioning, and define the state of the recent symbols ω_k' and the state of the older symbols ω_k'' as

$$\omega_k' = (a_{k-L+Q_b}, a_{k-L+Q_b+1}, \ldots, a_{k-1}) \tag{25}$$

$$\omega_k'' = (a_{k-L}, a_{k-L+1}, \ldots, a_{k-L+Q_b-1}) \tag{26}$$

Q_b being a nonnegative design parameter, such that $\sigma_k = (\omega_k', \omega_k'')$. Conditioning to each combination of the older symbols ω_k'', we save only the S' highest APPs of the recent symbols ω_k', according to

$$P(\sigma_k|\boldsymbol{y}_k^{K-1}) = P(\omega_k''|\boldsymbol{y}_k^{K-1})P(\omega_k'|\omega_k'', \boldsymbol{y}_k^{K-1}). \tag{27}$$

The number of saved metrics per time epoch is, thus, $S_b = M^{Q_b} S'$. Symmetrically with respect to the forward recursion, the set of BSPs is defined as maintaining complete information on the state of the older symbols and partial information on the state of the recent symbols. We could not find a general rule to choose, for a fixed complexity, the values of parameters Q_f, Q_b, S', and S'' providing the best performance, but they can be easily optimized by means of computer simulations. The basic algorithms presented before can be obtained by setting $Q_f = Q_b = 0$ and $S_f = S_b = S$.

IV. Numerical Results

In this section, the performance of the proposed algorithms is assessed by means of computer simulations in terms of bit error rate (BER) versus E_b/N_0, E_b being the received signal energy per information bit and N_0 the one-sided power spectral density of the passband noise. For each considered scenario, a comparison with the performance of the BCJR, M-BCJR, and RS-BCJR algorithms is given. To ensure better numerical stability, all the algorithms are implemented in the logarithmic domain [11]. The channels considered in the computer simulations are reported in Table I.

TABLE I
CHARACTERISTICS OF THE CONSIDERED CHANNELS

Channel	L	$\boldsymbol{f}$	Phase
$\mathcal{C}_1$	6	$\frac{1}{\sqrt{140}}(7,6,5,4,3,2,1)$	minimum
$\mathcal{C}_2$	6	$\frac{1}{\sqrt{140}}(1,2,3,4,5,6,7)$	maximum
$\mathcal{C}_3$	5	$\frac{1}{\sqrt{6}}(1,1,1,1,1,1)$	mixed
$\mathcal{C}_4$	6	$\frac{1}{\sqrt{8}}(1,0,1,2,1,0,1)$	mixed

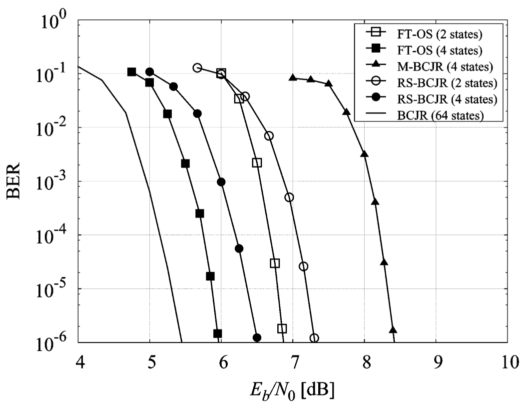

Fig. 2. Performance of different reduced-complexity algorithms for a coded BPSK transmission over the minimum-phase channel $\mathcal{C}_1$.

In Fig. 2, we consider a BPSK transmission over the minimum-phase channel $\mathcal{C}_1$. A $(3,6)$-regular LDPC code of rate $1/2$ and code word length of 4000 bits is used, and a detection instance is executed before each iteration of the SISO decoder, for a maximum of 40 iterations. The process also stops if, by checking the code syndrome, a valid code word is found before the 40th iteration. No interleaver is used because of the random nature of the LDPC code. As stated in the previous section, in such a system, it is convenient to adopt the FT algorithm optimized by saturating the extrinsic information to a proper minimum value. Fig. 2 shows that the proposed algorithm, implementing the OS technique with $\gamma = 1/10$, outperforms both the M-BCJR and RS-BCJR.[3] We remark that the OS technique, which in this case is the only difference between the FT-OS and the M-BCJR algorithms, provides a gain of about 2.5 dB. An alternative way to improve the performance of the M-BCJR algorithm consists of performing more than one iteration of the LDPC decoder for each iteration of the detector. In this scenario, we found that the M-BCJR algorithm gains about 0.5 dB of power efficiency when at least three consecutive decoding iterations are performed. Hence, the OS technique is the most convenient solution, since it is both simpler (the saturation is definitely less complex than performing additional decoding iterations) and more effective by about 2.0 dB.

In Fig. 3, we consider the maximum-phase channel $\mathcal{C}_2$. The coding scheme and the modulation are the same as adopted in the system of Fig. 2. The performance of the RS-BCJR and M-BCJR algorithms, as expected on maximum-phase channels, is absolutely unacceptable. The performance of the BT* algorithm, which are identical to the BT algorithm with the exception

[3]All the reported values of γ were chosen according to optimizations performed by means of computer simulations.

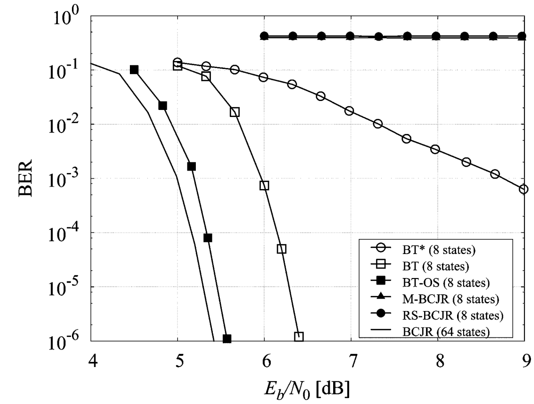

Fig. 3. Performance of different reduced-complexity algorithms for a coded BPSK transmission over the maximum-phase channel $\mathcal{C}_2$.

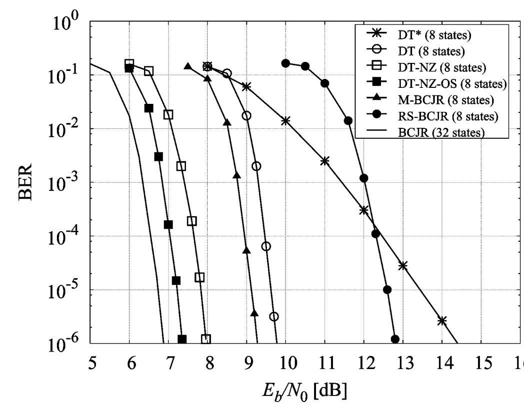

Fig. 4. Performance of different reduced-complexity algorithms for a coded BPSK transmission over the mixed-phase channel $\mathcal{C}_3$.

that it builds the set of BSPs on the basis of the classical backward recursion instead of the reverse one, is also reported. Fig. 3 proves that the proposed redefinition of the backward recursion ensures an impressive performance gain with respect to the traditional approach. A further gain of almost 1 dB is ensured by the OS technique, with $\gamma = 1/25$.

In Fig. 4, we consider the mixed-phase channel $\mathcal{C}_3$. Again, the coding scheme and the modulation are the same as adopted in the system of Fig. 2. With respect to the M-BCJR algorithm, the proposed DT algorithm loses about 0.5 dB if implemented with no optimization technique, while it gains about 1.5 dB when the NZ technique is adopted. Fig. 4 also confirms the effectiveness of the OS technique: in the considered case, a further gain of about 0.5 dB is obtained by saturating the extrinsic information evaluated by the algorithm and setting $\gamma = 1/30$. The curve related to the DT* algorithm, which is identical to the DT algorithm with the exception that it builds the set of BSPs on the basis of the classical backward recursion instead of the reverse one, gives a further proof of the need for a MAP-based selection of the set of BSPs. The M-BCJR algorithm loses about 2 dB from the DT-NZ-OS algorithm, while the RS-BCJR algorithm is ineffective on this mixed-phase channel. All the considered reduced-complexity algorithms work on eight states per trellis epoch, thus providing a reduction factor roughly equal to 4 with respect to the 32-state

TABLE II
REDUCTION FACTORS WITH RESPECT TO THE BCJR ALGORITHM

Algorithm	Forw. Rec.	Back. Rec.	Completion	Overall
DT	4	4	4.7*	4.2*
DT-NZ	4	4	3.5*	3.9*
DT-NZ-OS	4	4	3.5*	3.9*
M-BCJR	4	6.0*	4	4.6*
RS-BCJR	4	4	4	4

BCJR algorithm. More detailed measures of the reduction factors are given in Table II for each of the three stages of the algorithms, together with the overall reduction factors. These results refer to the number of nontrivial[4] multiply accumulate (MAC) operations only, while the operations of evaluating the branch metrics, sorting the state metrics, and saturating the output are not taken into account. In some cases, pointed out by means of an asterisk, the reduction factors cannot be evaluated in closed form, and the reported results are obtained by means of computer simulations related to a value of E_b/N_0 ensuring, for the corresponding algorithm, a BER of about 10^{-6}. As expected, the M-BCJR algorithm provides the highest reduction factor since the backward recursion is forced to explore only the set of FSPs. It is worth to remark that the slight decrease of the reduction factor due to the adoption of the NZ completion is largely compensated by the provided performance gain. Similar results have been obtained for other ISI channels and are not reported here.

We also investigated if the implementation of reduced-complexity algorithms requires a greater number of iterations to reach convergence with respect to the BCJR algorithm. For all scenarios discussed so far, we found that the mean number of iterations to reach convergence does not significantly depend on the detection algorithm when a target BER lower than 10^{-5} is considered—less than one additional iteration is required on average with respect to the optimal BCJR algorithm. Hence, the only performance degradation due to the implementation of reduced-complexity algorithms is the loss in terms of power efficiency.

In Fig. 5, we consider a BPSK transmission over the mixed-phase channel $\mathcal{C}_4$. A nonrecursive convolutional code with rate $1/2$ and generators $(5,7)_8$ is used, followed by a random interleaver. The considered code words have size 2000 bits, and a detection instance is executed before each iteration of the SISO decoder, performing 20 iterations. The curve related to perfect equalization (AWGN curve) is reported as reference. In this case, the DT-NZ algorithm reaches the optimal performance by keeping eight states, providing a significant gain with respect to the M-BCJR and RS-BCJR algorithms. Fig. 5 also shows the effectiveness of the SP technique: by keeping six states and setting the partitioning parameters $Q_f = 1$ and $Q_b = 0$, the DT-NZ-SP algorithm behaves as the full-complexity algorithm, whereas, by keeping six states without any partitioning (DT-NZ algorithm), the performance noticeably worsens.

[4]The terms giving a null contribution because of the unsatisfied indicator functions $T(a_k, \sigma_k, \sigma_{k+1})$ or $S(a_k, \sigma_k, \sigma_{k+1})$ are not considered.

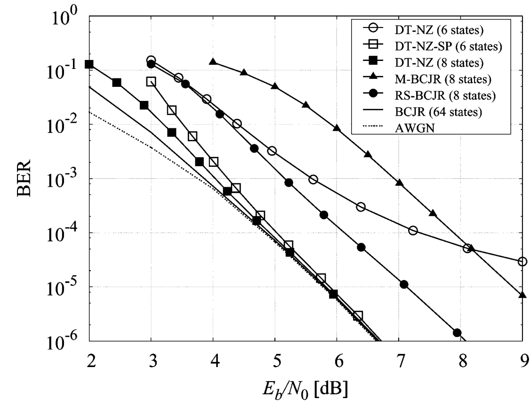

Fig. 5. Performance of different reduced-complexity algorithms for a coded BPSK transmission over the mixed-phase channel $\mathcal{C}_4$.

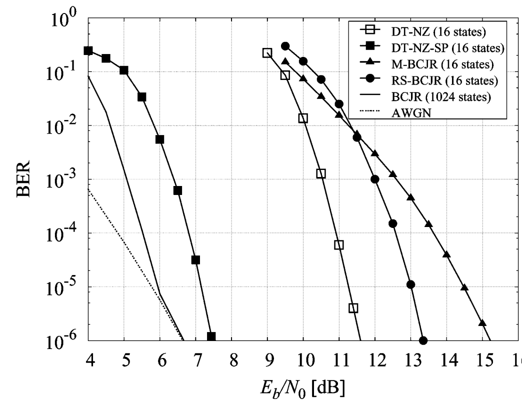

Fig. 6. Performance of different reduced-complexity algorithms for a coded QPSK transmission over the mixed-phase channel $\mathcal{C}_3$.

In Fig. 6, we finally consider a QPSK transmission over the mixed-phase channel $\mathcal{C}_3$, so that the full-complexity BCJR algorithm works on a 1024-state trellis. The coding scheme is the same as adopted in Fig. 5. While the DT-NZ algorithm is not particularly effective on this scenario, the implementation of the SP technique ensures an astonishing performance gain with respect to the other reduced-complexity algorithms, when they all work on 16 states per trellis epoch. As in the previous case, the state partitioning is implemented by setting $Q_f = 1$ and $Q_b = 0$. The resulting loss with respect to the BCJR algorithm is lower than 1 dB at a BER of 10^{-6}, with a complexity 64 times lower.

V. CONCLUSION

The design of reduced-complexity detection algorithms for turbo equalization schemes has been addressed. In particular, three low-complexity algorithms have been proposed, explaining how to choose among them based on the channel type. The proposed solutions have been derived from an alternative formulation of the BCJR algorithm, exactly equivalent to the original one when full-complexity implementations are considered, but much more suitable for deriving reduced-complexity algorithms. The MAP criterion has been adopted in order to select the most promising paths to be explored while performing the reduced-complexity searches on the trellis. Some optimization techniques providing a significant performance improvement have also been described. The proposed algorithms exhibit a very good flexibility to different coding structures and modulation formats, and impressively outperform the existing reduced-complexity algorithms.

APPENDIX

The complex envelope of the continuous-time received signal, in the case of linearly modulated transmissions over an ISI-AWGN channel, can be written as

$$r(t) = \sum_{m=0}^{K-1} a_m h(t - mT) + w(t) \tag{28}$$

where T is the signaling interval, $h(t)$ is the continuous-time channel impulse response, and $w(t)$ is a complex white Gaussian process with two-sided power spectral density N_0 per component. The pulse $h(t)$ has support for $t \in [0, (L+1)T]$, and its bandwidth B_h is assumed to be limited.[5] A couple of alternative detection approaches for this problem, namely the Forney approach [20], based on a whitened matched filter front end, whose output is given by (1), and the Ungerboeck approach [25], based on the matched filter front end, are known since the early 1970s. Both of them, although originally proposed for MAP *sequence* detection, have been extended to the case of the MAP *symbol* detection strategy. In fact, the original probabilistic derivation of the BCJR algorithm, as proposed in [1] and recalled in Section II, is based on the Forney observation model. On the other hand, although it does not seem possible to derive a MAP *symbol* detection extension of the Ungerboeck approach by means of probabilistic considerations, a solution has been recently found in [26] by applying the *sum–product algorithm* over a proper *factor graph* [27]. Unfortunately, although the resulting algorithm differs from the original BCJR algorithm in the expression of the branch metrics only, the fact that the forward and backward state metrics do not have a probabilistic meaning implies that the reduced-complexity techniques proposed in this paper cannot be extended to the Ungerboeck approach. In the following, we recall a further alternative scheme for MAP *symbol* detection over ISI channels derived from [28].

Let us assume to sample $r(t)$ at a rate N/T (where N is an integer), after filtering it by means of an ideal low-pass filter with bandwidth $B_{\rm LP} = N/(2T)$.[6] A set of sufficient statistics for detection can be obtained only if $N > 2TB_h$ [28]. In this case, the resulting samples can be written as

$$r_j = \sum_{m=0}^{K-1} a_m h\Big(j\frac{T}{N} - mT\Big) + w_j \tag{29}$$

[5]Since duration and bandwidth cannot both be limited, at least one of these assumptions must be an approximation.

[6]The following results can be generalized easily to the case of a more realistic low-pass filter, provided that its frequency response exhibits a vestigial symmetry with respect to $f = N/(2T)$ [28].

where $\{w_j\}$ are complex independent Gaussian random variables with mean zero and variance $\sigma_N^2 = N_0 N/T$ per component. By setting $j = kN + n$, with $n \in \{0, 1, \ldots, N-1\}$, $k \in \{0, 1, \ldots, K-1\}$, and defining

$$h_{i,n} = h\Big(iT + n\frac{T}{N}\Big) \tag{30}$$

we can write (29) as

$$r_{kN+n} = \sum_{i=0}^{L} h_{i,n} a_{k-i} + w_{kN+n}\,. \tag{31}$$

By following the probabilistic considerations in [1] and redefining the branch metrics as

$$F_k(a_k, \sigma_k) = P(a_k) \exp\left[-\frac{R_k(a_k, \sigma_k)}{2\sigma_N^2}\right] \tag{32}$$

$$\tilde{F}_k(a_k, \sigma_k) = P(a_{k-L}) \exp\left[-\frac{R_k(a_k, \sigma_k)}{2\sigma_N^2}\right] \tag{33}$$

where

$$R_k(a_k, \sigma_k) = \sum_{n=0}^{N-1} \left| r_{kN+n} - \sum_{i=0}^{L} h_{i,n} a_{k-i} \right|^2 \tag{34}$$

it is straightforward to extend the BCJR algorithm to the channel model (31). We found that all the recursive relations described in Section II-B still hold, with the following modifications of the probabilistic meanings

$$\alpha_k(\sigma_k) \propto P(\sigma_k | \boldsymbol{r}_0^{kN-1}) \tag{35}$$

$$\beta_k(\sigma_k) \propto p(\boldsymbol{r}_{kN}^{KN-1} | \sigma_k) \tag{36}$$

$$\tilde{\alpha}_k(\sigma_k) \propto p(\boldsymbol{r}_0^{kN-1} | \sigma_k) \tag{37}$$

$$\tilde{\beta}_k(\sigma_k) \propto P(\sigma_k | \boldsymbol{r}_{kN}^{KN-1})\,. \tag{38}$$

Finally, the APPs of the modulation symbols can still be computed according to

$$P(a_k | \boldsymbol{r}) \propto \sum_{\sigma_{k+1}} S(a_k, \sigma_{k+1}) \alpha_{k+1}(\sigma_{k+1}) \beta_{k+1}(\sigma_{k+1}) \tag{39}$$

$$\propto \sum_{\sigma_{k+1}} S(a_k, \sigma_{k+1}) \tilde{\alpha}_{k+1}(\sigma_{k+1}) \tilde{\beta}_{k+1}(\sigma_{k+1}). \tag{40}$$

It is worth to remark that this oversampling-based detection algorithm works on the same trellis as adopted by the original BCJR algorithm based on the Forney observation model, and that the only modifications are the branch metrics (32) and (33), which are slightly more complex than the original (3) and (11). Moreover, since the probabilistic meanings of the state metrics are the same described in Section II-B for the original BCJR algorithm, the reduced-complexity techniques proposed in this paper can be applied to such a detection scheme. Hence, this oversampling-based algorithm appears to be a convenient solution when the whitened matched filter, for any reason, cannot be implemented.

REFERENCES

[1] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inf. Theory*, vol. IT-20, no. 2, pp. 284–287, Mar. 1974.

[2] C. Douillard, M. Jezequel, C. Berrou, A. Picart, P. Didier, and A. Glavieux, "Iterative correction of intersymbol interference: Turbo-equalization," *Eur. Trans. Telecommun.*, vol. 6, pp. 507–511, Sep./Oct. 1995.

[3] M. Tüchler, R. Koetter, and A. C. Singer, "Turbo equalization: Principles and new results," *IEEE Trans. Commun.*, vol. 50, no. 5, pp. 754–767, May 2002.

[4] G. Colavolpe, G. Ferrari, and R. Raheli, "Noncoherent iterative (turbo) detection," *IEEE Trans. Commun.*, vol. 48, no. 9, pp. 1488–1498, Sep. 2000.

[5] P. Hoeher and J. Lodge, ""Turbo DPSK": Iterative differential PSK demodulation and channel decoding," *IEEE Trans. Commun.*, vol. 47, no. 6, pp. 837–843, Jun. 1999.

[6] M. V. Eyuboğlu and S. U. Qureshi, "Reduced-state sequence estimation with set partitioning and decision feedback," *IEEE Trans. Commun.*, vol. 36, no. 1, pp. 13–20, Jan. 1988.

[7] A. Duel Hallen and C. Heegard, "Delayed decision feedback estimation," *IEEE Trans. Commun.*, vol. 37, no. 5, pp. 428–436, May 1989.

[8] P. R. Chevillat and E. Eleftheriou, "Decoding of trellis-encoded signals in the presence of intersymbol interference and noise," *IEEE Trans. Commun.*, vol. 37, no. 7, pp. 669–676, Jul. 1989.

[9] G. Colavolpe, G. Ferrari, and R. Raheli, "Reduced-state BCJR-type algorithms," *IEEE J. Sel. Areas Commun.*, vol. 19, no. 5, pp. 848–859, May 2001.

[10] P. Thiennviboon, G. Ferrari, and K. M. Chugg, "Generalized trellis-based reduced-state soft-input/soft-output algorithms," in *Proc. IEEE Int. Conf. Commun.*, New York, Apr. 2002, pp. 1667–1671.

[11] P. Roberston, E. Villebrun, and P. Hoeher, "Optimal and sub-optimal maximum a posteriori algorithms suitable for turbo decoding," *Eur. Trans. Telecommun.*, vol. 8, pp. 119–125, Mar./Apr. 1997.

[12] V. Franz and J. B. Anderson, "Concatenated decoding with a reduced-search BCJR algorithm," *IEEE J. Sel. Areas Commun.*, vol. 16, no. 2, pp. 186–195, Feb. 1998.

[13] M. Sikora and D. J. Costello Jr., "A new SISO algorithm with application to turbo equalization," in *Proc. IEEE Int. Symp. Inf. Theory*, Adelaide, Australia, Sep. 2005, pp. 2031–2035.

[14] D. Bokolamulla, A. Hansson, and T. Aulin, "Low-complexity iterative detection based on bi-directional trellis search," in *Proc. IEEE Int. Symp. Inf. Theory*, Yokohama, Japan, Jun./Jul. 2003, p. 396.

[15] C. Fragouli, N. Seshadri, and W. Turin, "On the reduced trellis equalization using the M-BCJR algorithm," AT&T Labs-Res., Florham Park, NJ, Tech. Rep. TR 99.15.1, Nov. 1999.

[16] B. Frey and F. Kschischang, "Early detection and trellis splicing: Reduced complexity iterative decoding," *IEEE J. Sel. Areas Commun.*, vol. 16, no. 2, pp. 153–159, Feb. 1998.

[17] G. Colavolpe and G. Germi, "On the application of factor graphs and the sum–product algorithm to ISI channels," *IEEE Trans. Commun.*, vol. 53, no. 5, pp. 818–825, May 2005.

[18] J. G. Proakis and D. G. Manolakis, *Digital Signal Processing: Principles, Algorithms, and Applications*. 3rd ed. Upper Saddle River, NJ: Prentice-Hall, 1996.

[19] G. Colavolpe, G. Ferrari, and R. Raheli, "Extrinsic information in iterative decoding: A unified view," *IEEE Trans. Commun.*, vol. 49, no. 12, pp. 2088–2094, Dec. 2001.

[20] G. D. Forney Jr., "Maximum-likelihood sequence estimation of digital sequences in the presence of intersymbol interference," *IEEE Trans. Inf. Theory*, vol. 18, no. 3, pp. 363–378, May 1972.

[21] B. L. Yeap, C. H. Wong, and L. Hanzo, "Reduced complexity in-phase/quadrature-phase M-QAM turbo equalization using iterative channel estimation," *IEEE Trans. Wireless Commun.*, vol. 2, no. 1, pp. 2–10, Jan. 2003.

[22] N. Nangare, F. Neeser, and K. Narayanan, "A novel reduced complexity forward–backward equalization algorithm," presented at the Annu. Allerton Conf. Commun., Control Comput.. Monticello, IL, Oct. 2004.

[23] L. Papke, P. Robertson, and E. Villebrun, "Improved decoding with the SOVA in a parallel concatenated (turbo-code) scheme," in *Proc. IEEE Int. Conf. Commun.*, Dallas, TX, Jun.1996, pp. 102–106.

[24] J. Hagenauer and P. Hoeher, "A Viterbi algorithm with soft-decision outputs and its applications," in *Proc. IEEE Global Telecommun. Conf.*, Dallas, TX, Nov. 1989, pp. 1680–1686.

[25] G. Ungerboeck, "Adaptive maximum likelihood receiver for carrier-modulated data-transmission systems," *IEEE Trans. Commun.*, vol. COM-22, no. 5, pp. 624–636, May 1974.

[26] G. Colavolpe and A. Barbieri, "On MAP symbol detection for ISI channels using the Ungerboeck observation model," *IEEE Commun. Lett.*, vol. 9, no. 8, pp. 720–722, Aug. 2005.

[27] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum–product algorithm," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.

[28] H. Meyr, M. Oerder, and A. Polydoros, "On sampling rate, analog prefiltering, and sufficient statistics for digital receivers," *IEEE Trans. Commun.*, vol. 42, no. 12, pp. 3208–3214, Dec. 1994.

Dario Fertonani (S'06) was born in Mantua, Italy, in 1980. He received the Dr.Ing. degree (*cum laude*) in telecommunications engineering in 2004 from the University of Parma, Parma, Italy, where he is currently working toward the Ph.D. degree at the Department of Information Engineering.

His current research interests include various topics in digital communications and information theory, with particular interest on design issues related to iterative detection/decoding over channels with memory and channels with impulse noise. He participates in several research projects funded by the European Space Agency—European Space Research and Technology Centre (ESA-ESTEC), in collaboration with other important research groups and telecommunications companies.

Alan Barbieri was born in Parma, Italy, in 1979. He received the Dr.Ing. degree (*cum laude*) in telecommunications engineering and the Ph.D. degree in information technology from the University of Parma, Parma, in 2003 and 2007, respectively.

From March 2007 to July 2007, he was with the Department of Electrical Engineering, University of Southern California, Los Angeles, CA, as a Postdoctoral Visiting Scholar. He is currently a Research Associate in the Dipartimento di Ingegneria dell'Informazione (DII), University of Parma. His current research interests include digital transmission theory and information theory, with particular emphasis on channel coding, iterative joint detection and decoding algorithms, estimation of unknown parameters, and algorithms for synchronization. He participates in several research projects funded by the European Space Agency—European Space Research and Technology Centre (ESA-ESTEC) and important telecommunications companies.

Dr. Barbieri was the recipient of the Premio Conti for the year 2003, as the best graduate in information engineering at the University of Parma in 2003.

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

Since 1997, he has been with the University of Parma, where he is currently an Associate Professor of telecommunications. During 2000, he was a Visiting Scientist at the Institut Eurécom, Valbonne, France. His current research interests include digital transmission theory, adaptive signal processing, channel coding, and information theory. He is the author or coauthor of numerous scientific articles published in leading international journals and conference proceedings. He also holds a few industrial patents. He is a coauthor of the book Detection Algorithms for *Wireless Communications, With Applications to Wired and Storage Systems* (New York: Wiley, 2004).

Dr. Colavolpe received the Best Paper Award at the 13th International Conference on Software, Telecommunications and Computer Networks (SoftCOM 2005).