COMBINED BLIND EQUALIZATION AND TURBO DECODING

Javier Garcia-Frias and John D. Villasenor
Electrical Engineering Department
University of California, Los Angeles

Abstract— **We describe parallel concatenated codes for communication in the presence of intersymbol interference (ISI). Two iterative decoder structures which combine channel equalization and turbo decoding are presented. The first one combines the trellis representing each one of the constituent encoders with the ISI trellis and performs slightly better that the second structure, which treats the ISI as another constituent decoder. We show that for both methods it is possible to perform the equalization blindly, since the parameters of the channel can be estimated jointly with the combined turbo decoding/equalization algorithm. These methods involve little or no sacrifice in performance relative to the case where the channel taps are known by the receiver.**

I. Introduction

Many practical channels in communications, as well as in magnetic recording, present intersymbol interference (ISI). In this paper, we describe two different methods to modify a turbo decoder [1] in order to take into account this effect. We assume a channel modeled as a discrete time filter with coefficients $\{h_n\}$. The output sequence $\{v_k\}$ can be represented as:

$$v_k = \sum_{n=0}^{L} h_n x_{k-n} + n_k, \qquad (1)$$

where $\{x_k\}$ is the input sequence to the channel (which can take values ± 1) corresponding to the coded bits, and $\{n_k\}$ is a white gaussian noise sequence with zero mean and variance σ^2. $L+1$ is the ISI length. We assume i.i.d. symmetric sources.

The problem of convolutional decoding for ISI channels using the "turbo principle" [2] has been considered in several papers [2], [3]. The basic idea is to consider two different decoding blocks: one for the equalizer and the other for the convolutional code. Extrinsic information is calculated in each one of the blocks and passed to the other block in a similar way as it is done in turbo codes. Previous approaches to the problem of combined equalization (but without considering blind equalization) and turbo coding can be found in [4], [5] and in [6] in the context of magnetic recording. The decoder structure in [4], [5] is similar to the first of our methods, although the way in which the extrinsic information is managed is very different in both approaches. Our second method, based on building supertrellises combining the ISI and each of the constituent decoders (following the idea in [7] for convolutional codes), was partially proposed in [6], although

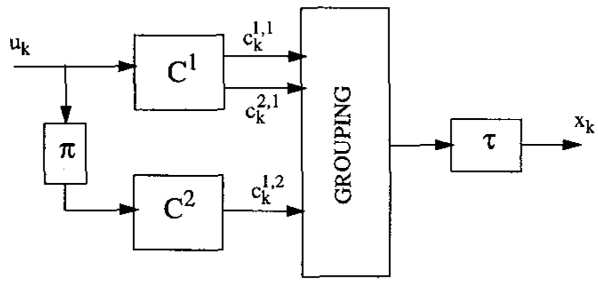

Fig. 1. Encoder structure used for the method 1 proposed in section II. π represents the turbo encoder interleaver (length $M = 16384$). τ represents the channel interleaver, with length, J, depending on M and the code rate. Both interleavers are generated pseudorandomly.

the supertrellis was only constructed for the first (non-interleaved) constituent encoder. As we will show in the paper, the improvement obtained by building supertrellises using all constituent encoders as opposed to only the first constituent coder, is significant. For both methods, we show that it is possible to perform the equalization blindly without degrading the decoding performance.

II. Method 1: Separate trellises for equalization and decoding

We consider the case of a parallel concatenated coder with a single interleaver and two constituent convolutional encoders. At the receiver, we refer to the portion of the decoder operating on the observations associated with the non-interleaved input sequence as the "non-interleaved decoder"; this contrasts with the "interleaved decoder" that operates on the observations corresponding to the interleaved input. The kth input bit (before interleaving) is denoted by u_k (with $k = 1 \ldots M$, where M is the interleaver length) and can take on values i, $i \in \{0, 1\}$. After the bits have been turbo coded, they are grouped and interleaved (forming the sequence $\{x_k\}$) and sent through the equivalent channel (1), as shown in Fig. 1. As indicated before, the received bits are denoted by $\{v_k\}$. Notice that there are two different interleavers, the one corresponding to the turbo encoder (of length M) and the channel interleaver (of length J). The relation between M and J is fixed for a given turbo code.

We use $O_1 \ldots O_K$ to represent the coded sequence at the encoder (not directly observable in the receiver), with

0-7803-5653-5/99/$10.00 © 1999 IEEE

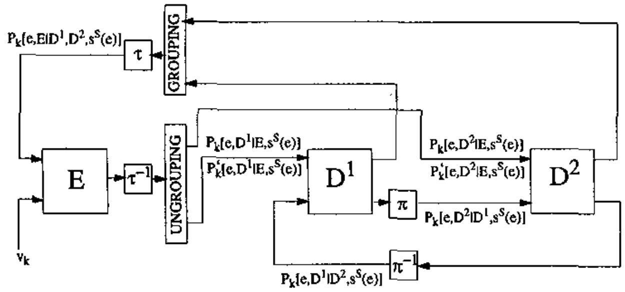

Fig. 2. Decoder structure for the method 1. The figure outlines the information flow as defined in section II. Notice that there is no observation sequence available for the constituent turbo decoders (D^1 and D^2). $\{v_k\}$ is the received sequence as defined by equation (1).

each O_k a vector whose elements are the coded bits associated with the trellis transition k. Notice that since the convolutional coder is forced to end in state 0, $K > M$. Some elements of O_k are produced using interleaved input; the rest are produced from the non-interleaved input. To make this distinction clearer during the decoding iterations in which one performs processing in the non-interleaved and interleaved decoders in alternation, we will denote by $O_k^p = [c_k^{1,p} \dots c_k^{n_p,p}]$ the subset of elements of O_k associated with the $1/n_p$ rate "present" decoder (non-interleaved or interleaved); i.e. the one in which processing is occurring. $O_k^f = [c_k^{1,f} \dots c_k^{n_f,f}]$ is used to denote the elements of O_k associated with the other, or "former" decoder (interleaved or non-interleaved) of rate $1/n_f$. For example, if the present constituent encoder has rate 1/2, and the other encoder has rate 1, $O_k^p = [c_k^{1,p}, c_k^{2,p}]$ and $O_k^f = [c_k^{1,f}]$. Obviously, there is a one to one relation (given by the channel interleaver) between $\{c_k^{1,p}, c_k^{2,p}, c_k^{1,f}\}$, $k \in [1, K]$ and $\{x_k\}$, $k \in [1, J]$. As in [8], we use e to symbolize the trellis edges, or branches, with the starting and ending state associated with a particular edge e given by $s^S(e)$ and $s^E(e)$ respectively. The input bit corresponding to branch e will be denoted by $u(e)$ and the output (coded) bits by $c(e) = [c(e)^{1,p} \dots c(e)^{n_p,p}]$.

The basic idea of this method is to treat the trellis describing the ISI in the channel (which has 2^L states, where $L+1$ is the length of the ISI channel) as another constituent decoder. Then, the decoding equations of the decoder in which processing is occurring are modified in such a way that the factor $P(e|s^S(e))$ (from the forward/backward equations [9]) is substituted by $P_k(e, D^p|s^S(e), D^{f_p})$, where D^{f_p} denotes all the other constituent decoders (including the equalizer block) and D^p represents the present decoder. In other words, we obtain an estimation of the transition probability of going through branch e by using the information available from the other constituent decoders. However, this substitution has to be done in such a way that positive feedback to the other decoders is avoided (i.e., passing only the so called extrinsic information). In order to make this clearer, we give the equations separately for the block corresponding to the equalizer (denoted by E) and for the two constituent convolutional decoders (denoted by D^1 and D^2). We also use the notation D^p and D^f to denote the present and the former constituent decoder, respectively. The decoder structure is shown in Fig. 2.

A. Equalizer block

The term $\alpha_k(s)$ represents, for the forward recursion, $P(v_1 \dots v_k, s_k = s)$, the probability of the received observation and that the equalizer trellis is in state s after the kth transition. $\beta_k(s)$ represents the probability $P(v_{k+1} \dots v_J | s_k = s)$ as calculated using the backward recursion. The output and input bits associated with a branch e are denoted by $v(e)$ and $x(e)$ respectively. The resulting equations are:

$$\alpha_k(s) = \sum_{e:s^E(e)=s} \alpha_{k-1}\left[s^S(e)\right] P_k\left[e, E|D^1, D^2, s^S(e)\right] \times P\left[v_k|e\right], \quad 1 \le k \le J+L \qquad (2)$$

$$\beta_k(s) = \sum_{e:s^S(e)=s} \beta_{k+1}\left[s^E(e)\right] P_{k+1}\left[e, E|D^1, D^2, s^S(e)\right] \times P[v_{k+1}|e], \quad J+L-1 \ge k \ge 1 \qquad (3)$$

$$P(x_k = i|E) = \frac{1}{P(v_1 \dots v_J)} \sum_{e:x(e)=i} \alpha_{k-1}\left[s^S(e)\right] P\left[v_k|e\right] \times \beta_k\left[s^E(e)\right], \quad 1 \le k \le J, \qquad (4)$$

where $P_k\left[e, E|D^1, D^2, s^S(e)\right]$ is calculated as indicated in section II.C and $P[v_k|e] = \frac{1}{\sigma\sqrt{2\pi}} exp(-\frac{(v_k - v(e))^2}{2\sigma^2})$.

B. Constituent decoder blocks

In order to derive the equations in this case, it is important to note that the observation sequences corresponding to these constituent decoder blocks are not available. In other words, the observation sequence in this case would be the encoded bits plus the noise, but the only sequence available at the receiver is $\{v_k\}$, which has been processed by the ISI channel. However, we do have the estimations of the coded bits $\{x_k\}$ given by equation (4). These estimations can be used in the formulas below via the terms $P_k\left[e, D^p|E, s^S(e)\right]$ and $P_k^{\prime}\left[e, D^p|E, s^S(e)\right]$. The resulting equations are:

$$\alpha_k(s) = \sum_{e:s^E(e)=s} \alpha_{k-1}\left[s^S(e)\right] P_k\left[e, D^p|E, s^S(e)\right] \times P_k\left[e, D^p|D^f, s^S(e)\right], \quad 1 \le k \le K \qquad (5)$$

$$\beta_k(s) = \sum_{e:s^S(e)=s} \beta_{k+1}\left[s^E(e)\right] P_{k+1}\left[e, D^p|E, s^S(e)\right]$$

$$\times P_{k+1}[e, D^p|D^f, s^S(e)], \;\; K-1 \geq k \geq 1 \qquad (6)$$

$$P(u_k = i|D^p) = \sum_{e:u(e)=i} \alpha_{k-1}\left[s^S(e)\right] P_k\left[e, D^p|E, s^S(e)\right]$$

$$\times \beta_k\left[s^E(e)\right], \;\; 1 \leq k \leq M \qquad (7)$$

$$P(c_k^{r,p} = i|D^p) = \sum_{e:c^{r,p}(e)=i} \alpha_{k-1}\left[s^S(e)\right]$$

$$\times P'_k\left[e, D^p|E, s^S(e)\right] P_k\left[e, D^p|D^f, s^S(e)\right] \beta_k\left[s^E(e)\right], \qquad (8)$$

with $1 \leq k \leq K$ and $r \in [1, n_p]$, where $1/n_p$ is the rate of the present constituent encoder. $P_k\left[e, D^p|E, s^S(e)\right]$, $P'_k\left[e, D^p|E, s^S(e)\right]$ and $P_k\left[e, D^p|D^f, s^S(e)\right]$ are calculated as indicated in section II.C. In the interleaved decoder, u_k in equation (7) should be replaced by $u_{\pi^{-1}(k)}$, where π is the turbo code interleaving function.

C. *Calculation of the estimated transition probabilities (extrinsic information)*

As indicated before, the estimation of the transition probabilities in a given constituent decoder has to be done in such a way that all possible information coming from the other decoders, including the equalizer block, should be used, but at the same time positive feedback between decoders should be eliminated. This is the reason for calculating and passing extrinsic information in standard turbo decoding. In the approach presented here, there are four different classes of extrinsic information that are passed between the decoders.

The extrinsic information passed from both constituent convolutional encoders to the equalizer block is denoted by $P_k\left[e, E|D^1, D^2, s^S(e)\right]$ and its value can be calculated as:

$$P_k\left[e, E|D^1, D^2, s^S(e)\right] = P(c_{k'}^{j,i} = x(e)|D^i), \qquad (9)$$

where $P(c_{k'}^{j,i} = x(e)|D^i)$ is the probability, as obtained from the constituent convolutional decoders by equation (8), that the input bit x_k corresponding to trellis transition k in the equalizer is equal to the input bit x(e) associated with the branch e in the equalizer trellis. Due to the channel interleaver and grouping process, bit x_k will correspond to one of the coded bits (for example the one in position j) associated with trellis transition k' of one of the constituent decoders (for example, decoder i) and therefore it will be denoted by $c_{k'}^{j,i}$.

There are two different classes of extrinsic information that are passed from the equalizer block to the constituent

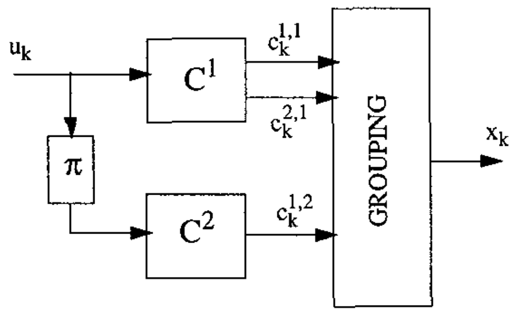

Fig. 3. Encoder structure used for the method 2 proposed in section III. π represents the turbo encoder interleaver (length $M = 16384$). In order to construct the supertrellises no channel interleaver is used and the coded bits are transmitted in blocks, i.e., first all the bits $\{c_k^{1,1}\}$ followed by $\{c_k^{2,1}\}$ and $\{c_k^{1,2}\}$.

decoders. $P_k\left[e, D^p|E, s^S(e)\right]$, to be used in equations (5,6,7) and $P'_k\left[e, D^p|E, s^S(e)\right]$, to be used in equation (8). Both are calculated in the same way, but in order to avoid positive feedback, only the coded bits different from $c_k^{r,p}$ are used to obtain the value of $P'_k\left[e, D^p|E, s^S(e)\right]$. The resulting equations are given by:

$$P_k\left[e, D^p|E, s^S(e)\right] = \prod_{j=1}^{n_p} P(x_{k^{(j)}} = c(e)^{j,p}|E) \qquad (10)$$

$$P'_k\left[e, D^p|E, s^S(e)\right] = \prod_{j=1, j\neq r}^{n_p} P(x_{k^{(j)}} = c(e)^{j,p}|E), \quad (11)$$

where $P(x_{k^{(j)}} = c(e)^{j,p}|E)$ is the probability, as given by equation (4), that the output bit $c_k^{j,p}$ corresponding to trellis transition k in the present decoder (which due to the channel interleaver will be in position $k^{(j)}$ in the trellis corresponding to the equalizer and will be denoted by $x_{k^{(j)}}$) is equal to the output bit, $c(e)^{j,p}$, associated with the branch e in the present decoder. $1/n_p$ is the rate of the present encoder.

Finally, the extrinsic information passed between constituent decoders is calculated as in commonly used implementations of turbo codes and given by:

$$P_k\left[e, D^p|D^f, s^S(e)\right] = P(u_{k'} = u(e)|D^f), \qquad (12)$$

where $P(u_{k'} = i|, D^f)$ is calculated from equation (7), with k' accounting for the turbo coder interleaver.

III. Method 2: Combined trellises for equalization and decoding

In this method, the Markov model representing the ISI in the channel is combined with each one of the constituent convolutional encoders to form a supertrellis. Therefore, in order to build the supertrellis, a channel interleaver is not used. As in section II, the received bits are denoted by $\{v_k\}$, but, in contrast with the first method, the coded bits are sent through the channel grouped in blocks corresponding to the same class of coded bits. This is shown in Fig. 3, where transmission of the bits $\{c_k^{j,p}\}$ is performed as a single block for each j. If the rate of a constituent encoder is less than 1, the supertrellis corresponding to that decoder will consist of the combination of the convolutional coder with several independent Markov models (each of them representing the ISI channel). In this section we slightly change the notation used before. We maintain the notation D^p to refer to the "present" and D^f to denote the "former" decoder (interleaved or non-interleaved), but we also use these superscripts to denote the received bits corresponding to the different packets sent through the channel. For example, if the present constituent encoder has rate 1/2 and the other encoder has rate 1, $v_k^p = [v_k^{1,p}, v_k^{2,p}]$ and $v_k^f = [v_k^{1,f}]$. When the present constituent coder has rate $1/n_p$, each branch in the supertrellis will have associated n_p output bits (after the ISI channel processing) denoted by $v(e)^p = [v(e)^{1,p} \dots v(e)^{n_p,p}]$, n_p coded bits denoted by $O(e)^p = [c(e)^{1,p} \dots c(e)^{n_p,p}]$ and one input bit denoted by u(e). The decoding structure is shown in Fig. 4 and the corresponding equations are given by:

$$\alpha_k(s) = \sum_{e:s^E(e)=s} \alpha_{k-1}\left[s^S(e)\right] P_k\left[e, D^p|D^f, s^S(e)\right] \times P\left[v_k^p|e\right], \; 1 \le k \le K+L \tag{13}$$

$$\beta_k(s) = \sum_{e:s^S(e)=s} \beta_{k+1}\left[s^E(e)\right] P_{k+1}\left[e, D^p|D^f, s^S(e)\right] \times P[v_{k+1}^p|e], \; K+L-1 \ge k \ge 1 \tag{14}$$

$$P(u_k = i|D^p) = \frac{1}{P(v_1^p \dots v_K^p)} \sum_{e:u(e)=i} \alpha_{k-1}\left[s^S(e)\right] \times P\left[v_k^p|e\right] \beta_k\left[s^E(e)\right], \; 1 \le k \le M, \tag{15}$$

where $P\left[v_k^p|e\right] = \prod_{j=1}^{n_p} \frac{1}{\sigma\sqrt{2\pi}} exp(-\frac{(v_k^{j,p} - v(e)^{j,p})^2}{2\sigma^2})$, with $1/n_p$ the rate of the present decoder. In the interleaved decoder, u_k in equation (15) should be replaced by $u_{\pi^{-1}(k)}$, where π is the turbo code interleaving function.

The extrinsic information $P_k\left[e, D^p|D^f, s^S(e)\right]$ is calculated as:

$$P_k\left[e, D^p|D^f, s^S(e)\right] = P(u_{k'} = u(e)|D^f), \tag{16}$$

where $P(u_{k'} = u(e)|D^f)$ is calculated from equation (15) and k' accounts for the effect of the turbo coder interleaver.

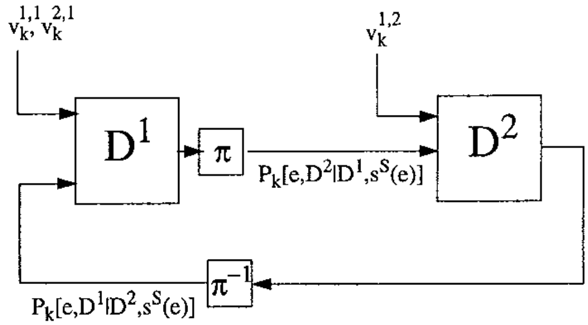

Fig. 4. Decoder structure for the method 2. Constituent decoders D^1 and D^2 make use of the supertrellises defined by the trellises of the constituent encoders and the ISI. In this case both decoders make use of the corresponding received sequence.

IV. Blind equalization

In the previous sections both methods assumed that the filter taps $\{h_n\}$ of the equivalent discrete ISI channel (1) were known. However, in most cases this information is not available. Since in order to apply the previous equations the parameters of the channel are needed, the simplest approach is to first estimate them and then apply the equations corresponding to either method 1 (described in section II) or method 2 (described in section III). One method for doing this estimation, proposed in [10] for the case where only equalization is performed, is to consider the trellis for the ISI channel and apply the Baum-Welch algorithm [11], which iteratively estimates the parameters of the model that are needed for decoding. Note that this ISI trellis is already used in the first method, however we have to build it for the second method. The equations resulting from the Baum-Welch algorithm can be found in [10]. Assuming that the values $v(e)^{(i-1)}$ and $\sigma_{(i-1)}$ are available from iteration $i-1$, the transition probability of each branch in the trellis is calculated as:

$$P_k(e = e'|E) = \frac{1}{P(v_1 \dots v_J)} \alpha_{k-1}\left[s^S(e')\right] \times P\left[v_k|e'\right] \beta_k\left[s^E(e')\right], \tag{17}$$

with $P\left[v_k|e\right] = \frac{1}{\sigma_{(i-1)}\sqrt{2\pi}} exp(-\frac{(v_k - v(e)^{(i-1)})^2}{2\sigma^2_{(i-1)}})$. The new parameters $v(e)^{(i)}$ and $\sigma_{(i)}$, resulting from iteration i, are then calculated as:

$$v(e)^{(i)} = \frac{\sum_{k=1}^{J} P_k(e|E) v_k}{\sum_{k=1}^{J} P_k(e|E)} \tag{18}$$

$$\sigma^2_{(i)} = \frac{1}{J} \sum_{k=1}^{J} \sum_{e} P_k(e|E)(v(e)^{(i)} - v_k)^2. \tag{19}$$

Another possibility is to estimate the parameters of the channel directly during the equalization/turbo decoding process. For the first method proposed in section II, the trellis representing the ISI is already a part of the decoder, so instead of initially iterating over this block until convergence to the channel parameters is reached, it is also possible to proceed with the decoding as indicated in section II and refine the estimation of the channel parameters after each one of the iterations over this block. For the second method proposed in section III, the trellis corresponding to the ISI channel is included in each one of the supertrellises representing the constituent decoders and the ISI, so it is also possible to obtain a new estimation of the channel parameters after decoding in each one of the supertrellises. We are not giving the final equations in this case, but a similar development can be found in [12] for the case of Markov channels.

The advantage of incorporating the estimation in the decoding loop as opposed to estimating the parameters first and then proceeding with the decoding is a reduction in complexity. This occurs because most of the parameters needed for the channel estimation are already calculated in the decoding equations and we eliminate the need of a first step of iterations of the Baum-Welch algorithm. Another advantage is that the estimates of the parameters are refined after each turbo decoding iteration. However, considering only the trellis for the ISI and iterating over this trellis is faster than iterating over the whole decoding block. Therefore, there is a trade-off involved in minimizing the complexity of the decoding process.

V. Simulation results

In order to assess the performance of the proposed method, we consider an ISI channel with taps $h_0 = .5$ and $h_1 = -.5$. We use a turbo code that includes a systematic bit and two identical recursive 8-state convolutional encoders (properly punctured depending on the turbo code rate) with generator matrix $G(D) = \frac{1+D+D^2+D^3}{1+D^2+D^3}$ and an interleaver with length $M = 16384$. Each simulation consisted of at least 20 million bits. For the implementation of the algorithm a logarithmic version of the equations was used to avoid numerical overflows.

Fig. 5 shows the decoded bit error rate as a function of the iteration number for several different values of E_b/N_0 using the previously defined 1/3 turbo code. The values of the filter taps are not known *a priori* and their initial values are initialized to $h_0^{in} = .1$, $h_1^{in} = -.3$. The initial value of E_b/N_0 is 0 dB. Notice that the performance of method 2 is slightly better than method 1 (about .1 dB) and it is about .75 dB from the theoretical limit for E_b/N_0 (2.95 dB) using this code rate and channel. Fig. 6 shows the decoding performance for a 1/2 turbo code obtained by puncturing the coded bits of the constituent encoders. As before, the values of the filter taps are not known *a priori* and their initial values are initialized to $h_0^{in} = .1$, $h_1^{in} = -.3$. The initial value of E_b/N_0 is 0 dB. Again the performance of method 2 is .1 dB better than method 1 and it is about .8 dB from the theoretical limit for E_b/N_0 (3.7 dB) using this rate and channel. The results in both figures have been obtained by first using the Baum-Welch algorithm and then refining the estimation in each decoding iteration. It is important to remark that in our simulations we assume that the lack of information holds for all the blocks, which implies that for each input block the estimation of the channel (initial Baum-Welch algorithm) has to be performed again with no information available. Similar performance (in E_b/N_0) is obtained when the parameters are estimated only in the decoding iterations (i.e. the Baum-Welch algorithm is not used to provide initial estimations). The only difference in this case is that more iterations are necessary to achieve convergence. Although not shown in the graphs, we have confirmed that the lack of initial knowledge about the filter taps and the noise in the channel does not increase the E_b/N_0 required for convergence.

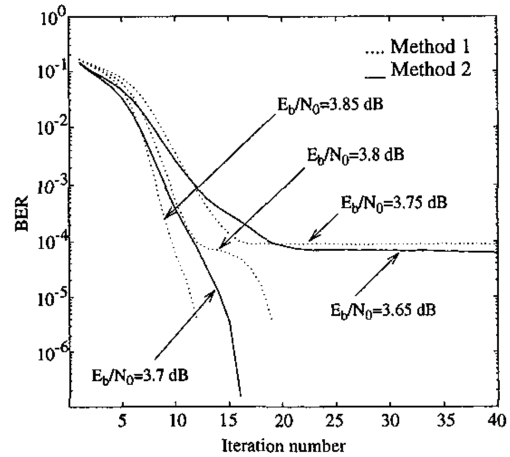

Fig. 5. Residual BER as a function of the decoding iteration number for the rate 1/3 turbo code and ISI channel defined in the paper. The dotted lines illustrate the performance for the method 1 proposed in section II. The solid lines represent the performance for the method 2 proposed in section III. The theoretical limit for this channel and rate is $E_b/N_0 = 2.95$ dB.

The simulation results presented in this paper assume some *a priori* information about the ISI channel ($h_0^{in} = .1$, $h_1^{in} = -.3$). However, it is also possible to perform the combined blind equalization/turbo decoding when no information about the ISI taps is assumed (i.e., $h_0^{in} = h_1^{in} = .0$). The performance (in terms of E_b/N_0 necessary to achieve convergence) is not degraded, but in this case more iterations of the initial Baum-Welch algorithm are needed and the method becomes more complicated, since the decoding process has to address several problems: First, sometimes the initial estimation of the ISI channel (obtained by the Baum-Welch algorithm) is the inverse of the channel, and the turbo decoding process is negatively affected. It is also necessary to avoid convergence into local maxima.

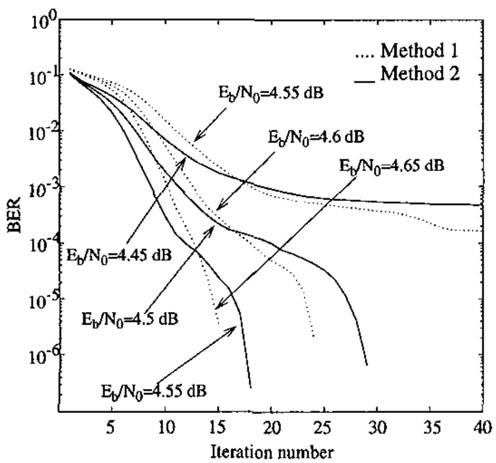

Fig. 6. Residual BER as a function of the decoding iteration number for the rate 1/2 turbo code and ISI channel defined in the paper. The dotted lines illustrate the performance for the method 1 proposed in section II. The solid lines represent the performance for the method 2 proposed in section III. The theoretical limit for this channel and rate is $E_b/N_0 = 3.7$ dB. The last coded bits leading the convolutional code to state 0 are not punctured.

VI. Conclusions

We have introduced two methods for combined equalization and turbo decoding. The performance of both methods is very similar, allowing decoding at values of E_b/N_o about .75 dB above the capacity corresponding to rate 1/3 codes and .8 dB from capacity for rate 1/2 codes, even when the taps of the channel are not known *a priori*. In both cases, the performance degradation in comparison with the case in which the ISI filter is known *a priori* is always very small. The first method (separate trellis for equalization and decoding) seems to be preferable to the second (joint trellis for equalization/decoding), since its complexity is much less and its performance is only .1 dB inferior with respect to the second method.

References

[1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon Limit Error-Correcting Coding and Decoding: Turbo Codes," *Proc. of ICC '93*, pp. 1064-1070, 1993.

[2] G. Bauch, H. Khorram and J. Hagenauer, "Iterative Equalization and Decoding in Mobile Communications Systems," *Second EPMCC'97 together with Third ITG-Fachtagung "Mobile Kommunikation"*, Bonn, Germany, October 1997.

[3] A. Glavieux, C. Laot and J. Labat, "Turbo Equalization over a Frequency Selective Channel," *Proceedings of the International Symposium on Turbo Codes and Related Topics*, pp. 96-102, Brest, France, September 1997.

[4] D. Raphaeli and Y. Zarai, "Combined Turbo Equalization and Turbo Decoding," *Proceedings of the International Symposium on Turbo Codes and Related Topics*, pp. 180-183, Brest, France, September 1997.

[5] D. Raphaeli and Y. Zarai, "Combined Turbo Equalization and Turbo Decoding," *Proc. of ICC '97*, pp. 639-643, Phoenix, Arizona November 1997.

[6] C. Heegard, "Turbo Coding for Magnetic Recording," *Proc. of the Winter 1998 Information Theory Workshop, pp. 18-19*, San Diego, California, February 1998.

[7] J.K. Omura, " Optimal Receiver Design for Convolutional Codes and Channels with Memory via Control Theoretical Concepts," *Information Sciences*, pp. 243-266, July 1971.

[8] S. Benedetto, D. Divsalar, G. Montorsi, G. and F. Pollara, "A Soft-Input Soft-Output APP Module for Iterative Decoding of Concatenated Codes," *IEEE Comm. Letters*, vol. 1, pp. 22-24, January 1997.

[9] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate," *IEEE Trans. on Inf. Theory*, pp. 284-287, March 1974.

[10] C. Anton-Haro, J. A. R. Fonollosa, "Blind Channel Estimation and Data Detection Using Hidden Markov Models," *IEEE Trans. on Signal Processing*, Vol.45, No. 2, pp. 241-246, Jan. 1997.

[11] L. R. Rabiner, "A Tutorial on hidden Markov Models and Selected Applications on Speech Recognition," *Proc. of the IEEE*, Vol.77, No. 2, pp. 257-285, Febr. 1989.

[12] J. Garcia-Frias and J. Villasenor, "Exploiting Binary Markov Channels with Unknown Parameters in Turbo Decoding," *Proc. of Globecom'98*, pp. 3244-3249, Sidney, Australia, November 1998.