BLIND TURBO DECODING AND EQUALIZATION

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

Abstract— **We present two turbo decoding structures which combine channel equalization and decoding, allowing communication in the presence of intersymbol interference (ISI). 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, with no *a priori* information available to the decoder. Compared with the case where the channel taps are known by the receiver, no loss in performance is observed. Application of the first method to the case of joint source-channel coding for hidden Markov sources is also presented.**

I. Introduction

Many practical channels in communications, as well as in magnetic recording, present intersymbol interference (ISI). We will assume that the equivalent channel is modeled as a discrete time filter with coefficients $\{h_n\}$ and the output sequence $\{v_k\}$ can be represented as:

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

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.

In this paper we propose two different methods for combined equalization and turbo decoding. We also show how to apply these methods for the case in which the decoder has no *a priori* information of the channel. Previous approaches to the problem of combined equalization (but without considering blind equalization) and turbo coding can be found in [1], [2] and in [3] in the context of magnetic recording.

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. 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 in blocks of J bits (forming the sequence $\{x_k\}$) and sent through

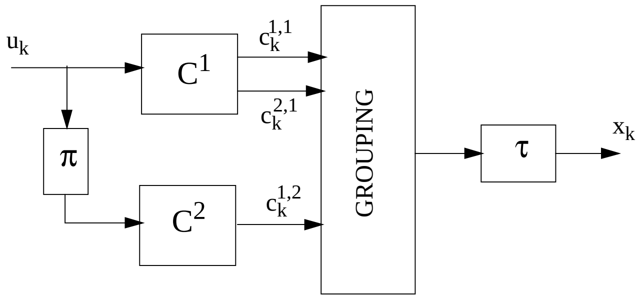

Fig. 1. Encoder structure used for the method 1 proposed in section II. π represents the turbo encoder interleaver (length $M = 16384$). τ represents the channel interleaver, with length, J, depending on M and the code rate. Both interleavers are generated pseudorandomly.

the equivalent channel (1), as shown in Fig. 1. The received bits are denoted by $\{v_k\}$.

We will denote by $O_k^p = [c_k^{1,p} \ldots c_k^{n_p,p}]$ the subset of elements of the coded sequence at the encoder (not directly observable in the receiver) associated with the $1/n_p$ rate "present" decoder (non-interleaved or interleaved); i.e. the one in which processing is occurring. $O_k^f = [c_k^{1,f} \ldots c_k^{n_f,f}]$ is used to denote the elements of O_k associated with the other, or "former" decoder (interleaved or non-interleaved) of rate $1/n_f$. As in [4], 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 $c(e) = [c(e)^{1,p} \ldots c(e)^{n_p,p}]$.

The basic idea of this method is to treat the trellis describing the ISI in the 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 [5]) 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 the next subsections, 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.

A. Equalizer block

The output and input bits associated with a branch e are denoted by $v(e)$ and $x(e)$ respectively. The resulting equations are:

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

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

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

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

B. Constituent decoder blocks

The estimations of the coded bits $\{x_k\}$ given by equation (4) can be used in the formulas below via the terms $P_k\left[e, D^p|E, s^S(e)\right]$ and $P'_k\left[e, D^p|E, s^S(e)\right]$. The resulting equations are:

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

$$\beta_k(s) = \sum_{e:s^S(e)=s} \beta_{k+1}\left[s^E(e)\right] P_{k+1}\left[e, D^p|E, s^S(e)\right] \times P_{k+1}[e, D^p|D^f, s^S(e)], \ K-1 \geq k \geq 1 \quad (6)$$

$$P(u_k = i|D^p) = \sum_{e:u(e)=i} \alpha_{k-1}\left[s^S(e)\right] \times P_k\left[e, D^p|E, s^S(e)\right] \beta_k\left[s^E(e)\right], \ 1 \leq k \leq M \quad (7)$$

$$P(c_k^{r,p} = i|D^p) = \sum_{e:c^{r,p}(e)=i} \alpha_{k-1}\left[s^S(e)\right] \times P'_k\left[e, D^p|E, s^S(e)\right] P_k\left[e, D^p|D^f, s^S(e)\right] \beta_k\left[s^E(e)\right], \quad (8)$$

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)

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:

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

where 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 decoders: $P_k\left[e, D^p|E, s^S(e)\right]$ and $P'_k\left[e, D^p|E, s^S(e)\right]$. 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:

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

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

where, due to the channel interleaver, the output bit $c_k^{j,p}$ will be in position $k^{(j)}$ in the trellis corresponding to the equalizer and will be denoted by $x_{k^{(j)}}$).

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

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

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. The turbo encoder structure is the same as in Fig. 1, with the difference that (in order to build the supertrellises) channel interleaver is not used. 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 (i.e. transmission of the bits $\{c_k^{j,p}\}$ is performed as a single block for each j). As in section II, the received bits are denoted by $\{v_k\}$. 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). 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 corresponding equations are given by:

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

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

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

where $P\left[v_k^p|e\right] = \prod_{j=1}^{n_p} \frac{1}{\sigma\sqrt{2\pi}} exp(-\frac{(v_k^{j,p} - v(e)^{j,p})^2}{2\sigma^2})$. 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 $P_k\left[e, D^p|D^f, s^S(e)\right] = P(u_{k'} = u(e)|D^f)$, with k' accounting for the effect of the turbo coder interleaver.

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 [6] for the case where only equalization is performed, is to consider the trellis for the ISI channel and apply the Baum-Welch algorithm [7], 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. 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:

$$P_k(e = e'|E) = \frac{1}{P(v_1 \dots v_J)} \alpha_{k-1}\left[s^S(e')\right] \times P\left[v_k|e'\right] \beta_k\left[s^E(e')\right], \tag{16}$$

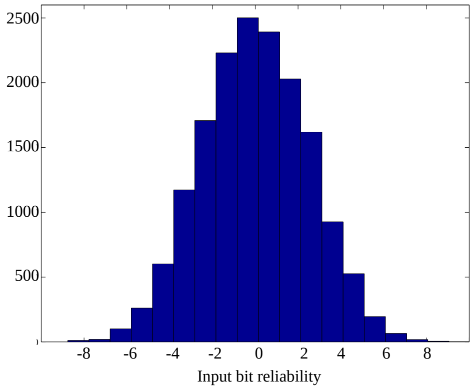

Fig. 2. Histogram of the input bit reliabilities when the estimation obtained by the Baum-Welch algorithm is the negative of the real channel. The histogram remains a unimodal curve independently of the number of turbo decoding iterations.

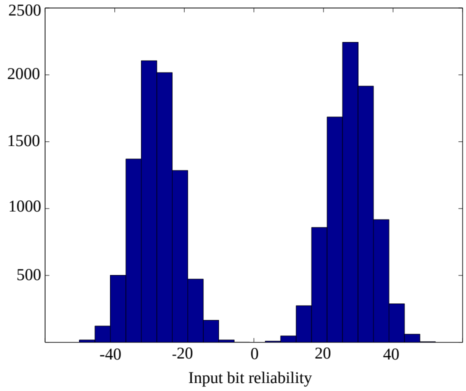

Fig. 3. Histogram of the input bit reliabilities when the estimation obtained by the Baum-Welch algorithm is the correct channel and the joint turbo decoding/equalization method has converged. The histogram becomes more "bi-modal" with the iteration number until a distribution such the one shown in the figure is reached.

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

$$v(e)^{(i)} = \frac{\sum_{k=1}^{J} P_k(e|E) v_k}{\sum_{k=1}^{J} P_k(e|E)} \tag{17}$$

$$\sigma^2_{(i)} = \frac{1}{J} \sum_{k=1}^{J} \sum_{e} P_k(e|E)(v(e)^{(i)} - v_k)^2. \tag{18}$$

The simulation results presented in this paper assume no *a priori* information about the ISI channel. Several problems need to be addressed in this case: 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. However, this problem can be solved by looking at distribution of the input bits reliabilities $log[P(u_k = 1)/P(u_k = 0)]$ obtained after turbo decoding. Essentially, turbo decoding implicitly acts as an error detection code. This is so because, as indicated in Fig. 2, for the blocks that are not converging (due to the use of the inverse of the channel resulting from the Baum-Welch algorithm) the distribution of the input bits reliabilities is unimodal, and centered around 0. However, it can be seen in Fig. 3 that the distribution of the input bits reliabilities for the blocks that achieve convergence is two sided, with a clear separation between the positive and negative reliabilities. It is therefore easy to perform blind equalization by using the following algorithm:

- Apply the Baum-Welch algorithm to estimate the filter taps and the noise in the channel.
- Perform the joint equalization/turbo decoding algorithm (method 1 or method 2) using the filter taps obtained from the previous step (the channel parameter can be refined in the decoding process).
- Decide if the block is in error by studying the histograms of the input bits reliabilities. If the histogram is of the type shown in Fig. 2, repeat step 2 changing the sign of the estimated tabs.

It is also necessary to avoid convergence into local maxima, which occurs when the parameters $v(e)^{(i)}$ do not change between successive iterations. However, this problem can be solved by adding a random noise to the value of $v(e)^{(i)}$.

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 10 million bits. For the implementation of the algorithm a logarithmic version of the equations was used to avoid numerical overflows.

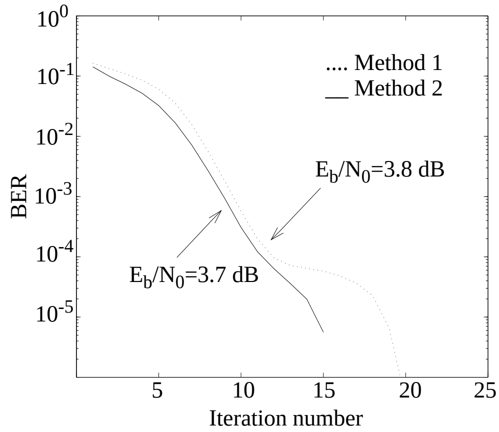

Fig. 4. Residual BER as a function of the decoding iteration number for the rate 1/3 turbo code and ISI channel defined in the paper. The dotted lines illustrate the performance for the method 1 proposed in section II. The solid lines represent the performance for the method 2 proposed in section III. The theoretical limit for this channel and rate is $E_b/N_0 = 2.95$ dB.

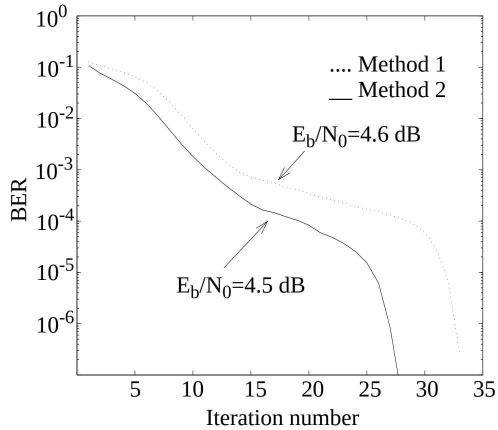

Fig. 5. Residual BER as a function of the decoding iteration number for the rate 1/2 turbo code and ISI channel defined in the paper. The dotted lines illustrate the performance for the method 1 proposed in section II. The solid lines represent the performance for the method 2 proposed in section III. 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.

Fig. 4 shows the decoded bit error rate as a function of the iteration number for the worst value of E_b/N_0 which allows convergence using the previously defined 1/3 turbo code. The values of the filter taps are not known *a priori* and therefore their initial values are initialized to $h_0^{in} = 0$, $h_1^{in} = 0$. 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. 5 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 $h_0^{in} = 0$, $h_1^{in} = 0$. 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. 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.

VI. Application to Hidden Markov Sources

An approach similar to method 1 can be applied for the case of transmission of hidden Markov sources over AWGN channels. The idea is to treat the trellis describing the hidden Markov source as another constituent decoder which produces and uses extrinsic information in each one of the turbo decoding iterations. More specifically, the source block uses as extrinsic information the probability of the input bits that is provided by the constituent decoder blocks. On the other hand, it produces a new estimation of such probability which will be used as extrinsic information by the constituent turbo decoders. In order to achieve good performance, it is necessary to use a channel interleaver to separate the turbo coder from the hidden Markov source. One difference with respect to the case of equalization is that now the trellis representing the hidden Markov source has two parallel branches between states (one representing the input bit 0 and the other representing the input bit 1). An *a priori* transition probability (that is obtained from the hidden Markov model) is associated with each one of the branches in the trellis.

Although we are not giving the detailed equations, we have simulated this method for different types of hidden Markov sources. No *a priori* information about the parameters of the model is needed, since those parameters can be estimated at the decoder with no loss in performance. Simulation results are very similar (perhaps a little bit better) to the ones obtained in [8]. However, the method proposed in [8] was quite complex, since it involved building a supertrellis which jointly described the non-interleaved constituent encoder and the hidden Markov source. The method proposed here achieves the same performance with a slightly more complexity than a standard turbo decoder.

VII. Conclusions

We have introduced two methods for combining blind 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. Application of ideas based on method 1 to the case of joint source-channel coding for hidden Markov sources results in a performance similar to the one obtained by much more complex approaches.

References

[1] 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.

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

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

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] J. Garcia-Frias and J. Villasenor, "Turbo Decoding of Hidden Markov Sources with Unknown Parameters," *Proc. of DCC'98*, pp. 159-168, Snowbird, Utah, March 1998.