Turbo Decoding of Hidden Markov Sources with Unknown Parameters [1]

Javier Garcia-Frias and John D. Villasenor
Electrical Engineering Department
University of California, Los Angeles
jgarcia,villa@icsl.ucla.edu

Abstract: *We describe techniques for joint source-channel coding of hidden Markov sources using a modified turbo decoding algorithm. This avoids the need to perform any explicit source coding prior to transmission, and instead allows the decoder to utilize the a priori structure due to the hidden Markov source. In addition, we present methods that allow the decoder to estimate the parameters of the Markov model. In combination, these techniques allow the decoder to identify, estimate, and exploit the source structure. The estimation does not degrade the performance of the system, i.e. the joint estimation/decoding allows convergence at the same noise levels as a system in which the decoder has perfect a priori knowledge of the source parameters.*

I. Introduction

We consider communications of a binary source that follows a hidden Markov model. In a traditional communications approach in which source and channel coding are performed separately, one would apply entropy coding to the Markov source and channel coding to the output of the source coder. However, the residual bit errors at the output of the channel decoder could cause a catastrophic loss of synchronization for the source decoder. By contrast, we develop a joint source-channel coding approach that exploits the statistical characteristics of the source in the decoder so that any bit errors arising during transmission are not propagated. Furthermore, joint source-channel decoding allows a channel code of a single rate to be used in combination with sources having arbitrary entropy rates, with the modifications to maintain efficient coding involving only processing in the decoder.

The principal contributions of the work presented here are as follows: First, we describe joint source-channel coding of hidden Markov sources. While previous authors have considered Markov source structure in combination with transmission over a noisy channel [1, 2], the previous work has emphasized (non-hidden) Markov models. By contrast, we focus on the more general case of hidden Markov models, in which the state of the model is related to the output probabilistically as opposed to deterministically. Second, we describe joint source-channel coding using parallel concatenated codes, or "turbo" codes [3, 4, 5, 6]. Turbo codes are the most widely practiced example of a new approach to channel coding in which iterative processing in multiple "constituent" decoders is used to enable decoding of long blocks of data. Although there has been a rapidly growing body of publications on turbo codes since their introduction in [3], much of the research in this area has focused on transmission of i.i.d. sources over additive white Gaussian noise (AWGN) channels. To exploit a source with Markov structure in turbo codes requires a formulation that utilizes the structure of the source in both the calculation of the *a posteriori* probabilities of the input bits and in the passage of information between constituent decoders in iterative decoding. This is related to techniques used in coding for intersymbol interference

[1] Proceedings of the 1998 IEEE Data Compression Conference, pp. 159-168, Snowbird, UT, March 1998

(ISI) channels, though in contrast with ISI channels in which the path through the expanded trellis is a deterministic function of the input sequence, in turbo decoders modified for hidden Markov sources the path is probabilistic. Third, we present a method for estimating the parameters of a hidden Markov source based on observations after channel coding and corruption by noise. This parameter estimation is refined in each turbo decoding iteration and allows successful turbo decoding of hidden Markov sources in the absence of initial knowledge of the parameters of the source. Hidden Markov parameter estimation is well known in speech coding in the context of the Baum-Welch algorithm [7]. However, iterative estimation of hidden Markov sources in a turbo decoder has not been addressed before.

II. Background and notation

As described in the references on turbo codes cited above, a parallel concatenated code is formed by several constituent convolutional coders, each one acting over permutated (interleaved) versions of the same input sequence. Though the methods we describe can be applied to parallel concatenated codes with any number of constituent convolutional encoders, they are most effective for the case of a single interleaver and two constituent convolutional encoders, so we consider this case here. 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 $k = 1 \dots L$, where L is the interleaver length) and can take on values i, $i = 0, 1$. We use $O_1 \dots O_K$ to represent the observation sequence, with each O_k a vector whose elements are the observations at the channel output associated with the kth trellis transition. Since the convolutional coder is forced to end in state 0, $K > L$.

Some of the elements of each O_k are produced using the interleaved input; the rest are produced from the non-interleaved input. To facilitate distinguishing between these two observations, we will denote by O_k^0 the subset of elements of O_k associated with the non-interleaved decoder; O_k^1 will refer to observations used in the interleaved decoder. During the decoding iterations, processing is performed in the non-interleaved and interleaved decoders in alternation. Therefore, it is also helpful to be able to refer to the observations in the decoders in a relative sense. Specifically, we will denote by O_k^p the subset of elements of O_k associated with the "present" decoder (non-interleaved or interleaved); i.e. the one in which processing is occurring, while O_k^f is used to denote the elements of O_k associated with the other, or "former" decoder (interleaved or non-interleaved). As in [6], 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 associated with a branch e is denoted by $u(e)$.

The MAP algorithm originally proposed in [8] laid the groundwork for the forward-backward algorithm that is commonly used in turbo codes. While this algorithm is now well known, we present the recursion equations below using notation that differs from that typically used in the turbo code literature, and which much more intuitively admits modifications that allow consideration of Markov source structure. In the discussion that follows, $\alpha_k(s)$ represents, for the forward recursion, $P(O_1^p \dots O_k^p, q_k = s)$, the probability of the observation and that the trellis state q_k after the kth transition is s. $\beta_k(s)$ represents the probability $P(O_{k+1}^p \dots O_K^p | q_k = s)$ as calculated using the backward recursion. The key results in [8] are the recursions that express $\alpha_k(s)$ in terms of $\alpha_{k-1}(s)$, and $\beta_k(s)$ in terms of $\beta_{k+1}(s)$. For parallel concatenated codes the forward recursion can be written in a form that utilizes a summation over all edges e that terminate in state s; i.e. that satisfy $s^E(e) = s$:

$$\alpha_k(s) = \sum_{e:s^E(e)=s} \alpha_{k-1}\left[s^S(e)\right] P_k\left[e|O_1^f \ldots O_K^f, s^S(e)\right] P\left[O_k^p|e\right], \quad 1 \le k \le K. \tag{1}$$

Since the convolutional code starts in state 0 the recursion is initialized using $\alpha_0(s = 0) = 1$ and $\alpha_0(s) = 0$ for $s \neq 0$. By analogy with the case for the forward recursion, the backward recursion used to calculate $\beta_k(s)$ can be expressed:

$$\beta_k(s) = \sum_{e:s^S(e)=s} \beta_{k+1}\left[s^E(e)\right] P_{k+1}\left[e|O_1^f \ldots O_K^f, s^S(e)\right] P[O_{k+1}^p|e], \quad K-1 \ge k \ge 1, \tag{2}$$

where the summation is performed over all edges e that begin in state s. The convolutional code ends in state 0, giving initialization $\beta_K(s = 0) = 1$ and $\beta_K(s) = 0$ for $s \neq 0$. Equations (1) and (2) explicitly show the flow of information between the two decoders via the center term in the summation. As will be discussed below in connection with equations (5) and (6), this allows direct incorporation of the *a priori* source probabilities.

The goal of the decoding iterations is to determine the probability that the kth input bit u_k was equal to i, i=0 or 1, given the observations. In the non-interleaved decoder, this probability can be expressed as a sum of probabilities over all edges in section k of the trellis for which the input bit $u(e)$ has value i, i.e.:

$$P(u_k = i|O_1^p \ldots O_K^p) = \frac{1}{P\left(O_1^p \ldots O_K^p\right)} \sum_{e:u(e)=i} \alpha_{k-1}\left[s^S(e)\right] P\left[e|s^S(e)\right] P\left[O_k^p|e\right] \beta_k\left[s^E(e)\right], \tag{3}$$

with $1 \le k \le L$.

In the interleaved decoder, u_k in equation (3) should be replaced by $u_{\tau^{-1}(k)}$, where τ is the interleaving function. We will also use the probability $P(O_1^p \ldots O_K^p|u_k = i)$, which can be related to $P(u_k = i|O_1^p \ldots O_K^p)$ through:

$$P(O_1^p \ldots O_K^p|u_k = i) = \frac{P(O_1^p \ldots O_K^p)}{P(u_k = i)} P(u_k = i|O_1^p \ldots O_K^p), \tag{4}$$

where $P(u_k = i)$ is the *a priori* probability of the source to generate the bit i. In terms of the notation introduced above, decoding in turbo codes involves forming estimates of $P(u_k = i|O_1^p \ldots O_K^p)$, and iteratively passing these estimates via the interleaver to the other constituent decoder, where they are further refined. After any number of iterations, the estimates $P(u_k = i|O_1^j \ldots O_K^j)$, $j = 0, 1$ from both decoders can be combined to obtain the hard decision over the input bit.

III. Decoder modifications to include source models

We now consider the case where the input sequence u_k is provided by a binary hidden Markov source with N states $\{S_0 \ldots S_{N-1}\}$. We denote the state at time k as q_k. Complete specification of a binary hidden Markov model requires specification of three probability measures, A, B, and π, defined as:

- A=(a_{ij}) is the state transition probability matrix, with a_{ij} the probability of transition from state S_i to state S_j, i.e. $a_{ij} = P_t(S_j|S_i) = P(q_{k+1} = S_j|q_k = S_i), 0 \le i, j \le N-1$.

- B=$(b_j(v))$ is the observation symbol probability matrix, with $b_j(v)$ the probability of getting the output v in state S_j, i.e. $b_j(v) = P_o(v|S_j) = P(\text{bit v at k}|q_k = S_j), 0 \leq j \leq N-1,\ 0 \leq v \leq 1$.

- $\pi = (\pi_j)$ is the initial state distribution vector, with π_j the probability for the initial state to be S_j, i.e. $\pi_j = P(q_0 = S_j), 0 \leq j \leq N-1$.

We use $\lambda = \{A, B, \pi\}$ to denote the parameters of a hidden Markov source.

In the non-interleaved decoder, the properties of the hidden Markov source can be exploited by replacing the trellis corresponding to the non-interleaved convolutional encoder with an expanded trellis (supertrellis) that jointly describes the source and the convolutional coder [9]. As shown in Figure 1, the number of states in the supertrellis is the product of the number of states in the hidden Markov source times the number of states in the convolutional coder. A given state s of the supertrellis is represented by a vector $s = (s^{(c)}, s^{(m)})$, where the component $s^{(c)}$ indicates the state of the convolutional code and $s^{(m)}$ expresses the state of the hidden Markov source. Obviously, the complexity of using a joint trellis in decoding is higher for hidden Markov sources with more states. Each one of the branches e in the supertrellis will have an associated *a priori* transition probability, a_e, which depends on the parameters of the hidden Markov source according to $a_e = P(e|s^S(e)) = P_o\left(u(e)\,|[\,s^S(e)]^{(m)}\right) \times P_t\left([s^E(e)]^{(m)}\,|[\,s^S(e)]^{(m)}\right)$. Since the initial and final state of the hidden Markov source are not known, equations (1) and (2) for the non-interleaver decoder are initialized using $\alpha_0(s) = 1/N$ if $s^{(c)} = 0$ and $\beta_K(s) = 1/N$ if $s^{(c)} = 0$. For all other states $\alpha_0(s)$ and $\beta_K(s)$ are equal to 0.

In the interleaved decoder no expansion of the trellis is needed since interleaving masks the Markov properties of the source. However, in the interleaved decoder one can still exploit any asymmetry between the fraction of 0's and 1's produced by the source. This is handled by assigning a transition probability a_e to each branch, though in contrast with the a_e in the non-interleaved decoder, the a_e in the interleaved decoder depends only on the associated $u(e)$, i.e., $a_e = P(u_k = u(e))$.

As in the case where the input is i.i.d., decoding when the source has Markov structure can be performed by iteratively applying equations (1,2,3) in the non-interleaved and interleaved decoders to obtain successively more reliable estimates of $P(u_k = i|O_1^p \ldots O_K^p)$. The key difference is that the calculation of $P_k[e|O_1^f \ldots O_K^f, s^S(e)]$ for use in equations (1) and (2) should utilize both the information available from the "former" decoder, $P(u_k = i|O_1^f \ldots O_K^f)$ or $P(O_1^f \ldots O_K^f|u_k = i)$, as well as the *a priori* information (a_e) in each of the branches due to the structure of the source. In the non-interleaved decoder $P(u_k = i|O_1^p \ldots O_K^p)$ is calculated and passed to the input of the interleaved decoder (where it gets renamed $P(u_k = i|O_1^f \ldots O_K^f)$ to identify it as being from the former decoder). This gives

$$P_k[e|O_1^f \ldots O_K^f, s^S(e)] = G_k P[u_{\tau^{-1}(k)} = u(e)|O_1^f \ldots O_K^f] \tag{5}$$

for use in equations (1) and (2) in the interleaved decoder, where G_k is a normalization factor. In passing information from the interleaved decoder to the non-interleaved decoder, we use $P(O_1^f \ldots O_K^f|u_k = i)$. To calculate the transition probability to use in equations (1,2) in the non-interleaved decoder, we apply $P_k[e|O_1^f \ldots O_K^f, s^S(e)] \times P_k[O_1^f \ldots O_K^f|s^S(e)] = P_k[O_1^f \ldots O_K^f|e, s^S(e)]P[e|s^S(e)]$. Due to the randomization introduced by the interleaver, given the initial state ($s^S(e)$) and transition branch (e) in the non-interleaved decoder, the only information that can be used to calculate the probability of the observation sequence ($O_1^f \dots O_K^f$) received by the interleaved decoder is the input bit $u(e)$ associated with the branch e, which is common to both decoders. That is, $P_k[O_1^f \dots O_K^f|e, s^S(e)] = P[O_1^f \dots O_K^f|u_k = u(e)]$. Similarly, given the initial state ($s^S(e)$) in the non-interleaved decoder, there is no information available to calculate the probability of the observation sequence received by the interleaved decoder, which implies $P_k[O_1^f \dots O_K^f|s^S(e)] = P_k[O_1^f \dots O_K^f]$ and leads to the equation:

$$P_k[e|O_1^f \dots O_K^f, s^S(e)] = H_k\, P[O_1^f \dots O_K^f|u_k = u(e)]a_e \tag{6}$$

where H_k is a normalization factor. Equation (5) can also be derived by passing the information in terms of $P(O_1^f \dots O_K^f|u_k = i)$ instead of $P(u_k = i|O_1^f \dots O_K^f)$, following a procedure similar to that used for expressing equation (6). By contrast, to take full advantage of the structure of the joint (i.e. including the source and encoder) hidden Markov model, the information in equation (6) must be passed using $P(O_1^f \dots O_K^f|u_k = i)$.

To achieve good performance it is necessary that the non-interleaved and interleaved decoders perform independent estimations of $P(u_k = i|O_1^p \dots O_K^p)$. Therefore, to calculate the value of $P[e|s^S(e)]$ in equation (3) we consider only its *a priori* value (i.e., $P[e|s^S(e)] = a_e$, with a_e the transition probability of branch e) and do not consider the value of $P(u_k = i|O_1^f \dots O_K^f)$ or $P(O_1^f \dots O_K^f|u_k = i)$ produced by the former decoder. For the case of i.i.d. sources the equations above are equivalent to the usual turbo decoding method in which "extrinsic" information is passed between the constituent decoders. As discussed in Section V, the modifications to incorporate the source model produce a strong improvement in decoding performance relative to the case in which source structure is not considered.

IV. Parameter estimation at the receiver

The methods above allow consideration of a hidden Markov source whose parameters are known. A more powerful approach would allow the receiver to use the noise-corrupted observations available in the decoder to estimate the parameters characterizing the hidden Markov source, and to apply these probabilities in decoding. Parameter estimation for hidden Markov models has been described before, most notably in conjunction with the Baum-Welch algorithm [7, 10, 11] which has been widely used in speech processing, but also for characterizing mobile channels [12] and in blind equalization. While the basic problem faced here and in the Baum-Welch algorithm is the same – to estimate the parameters of a hidden Markov model – the critical difference is that in the Baum-Welch algorithm the "observations" used in the algorithm are provided by the output of the hidden Markov model, while in our case the model output is subject to channel coding, corrupted by noise, and then observed.

One way to perform estimation is to make use of a standard turbo decoder and utilize the hard decisions obtained after the first (and subsequent if desired) iteration as the observation sequence provided to the Baum-Welch algorithm. The parameters obtained from the Baum-Welch algorithm can then be utilized using the methods of Section III. However, since a very large percentage of the hard decisions after one iteration of turbo decoding are wrong, the parameter estimation based on this observation is not very accurate, and this mismatch will limit the effectiveness of the joint source-channel decoding approach. Another drawback of this approach is that it does not utilize the soft information available in the turbo decoder. In order to have an initial estimation of the hidden Markov source parameters, we will use this approach in the first iteration.

A much better approach for the subsequent iterations is to estimate the hidden Markov source parameters in the supertrellis of the non-interleaved decoder that jointly describes the Markov source and the first constituent convolutional coder. In fact, it is possible to apply a reestimation algorithm over the supertrellis of the non-interleaved decoder so that each iteration in turbo decoding also results in an iteration on the parameter estimation. In this way, both the turbo decoding and the parameter estimation make use of soft information. This allows reduced complexity and better performance relative to an approach in which the Baum-Welch algorithm is applied after each turbo decoding iteration.

To derive the estimation algorithm we will assume that the source is characterized by a hidden Markov model $\lambda = \{A, B, \pi\}$ and therefore that we can calculate the value of a_e for each branch in the supertrellis. The probability of going through branch e in the trellis section k is given by:

$$P_k(e|O_1^p \ldots O_K^p, \lambda) = \frac{1}{P\left(O_1^p \ldots O_K^p\right)} \alpha_{k-1}\left[s^S\left(e\right)\right] a_e P\left[O_k^p|e\right] \beta_k \left[s^E\left(e\right)\right]. \tag{7}$$

Since branch e is determined by the initial state $s^S(e)$, final state $s^E(e)$ and its associated input bit $u(e)$, we can express the probability of the branch e given the observation as the joint probability of $s^S(e), u(e)$ and $s^E(e)$, i.e. $P_k(e|O_1^p \ldots O_K^p, \lambda) = P_k(s^S(e), u(e), s^E(e)|O_1^p \ldots O_K^p, \lambda)$. Therefore, the probability $P(S_i, v, S_j|O_1^p \ldots O_K^p, \lambda)$ can be expressed as:

$$P(S_i, v, S_j|O_1^p \ldots O_K^p, \lambda) = \frac{1}{K} \sum_{k=1}^{K} \sum_{e \in C} P_k(e|O_1^p \ldots O_K^p, \lambda), \quad 0 \le i, j \le N-1, \quad 0 \le v \le 1, \tag{8}$$

where $C = \{e : u(e) = v, [s^S(e)]^{(m)} = S_i, [s^E(e)]^{(m)} = S_j\}$.

The product of this probability times the number of trellis transitions represents the expected number of transitions from state S_i to state S_j generating the input bit v. From this joint probability it is easy to calculate the expected number of transitions from state S_i to state S_j ($K \times P(S_i, S_j|O_1^p \ldots O_K^p, \lambda)$), the expected number of times symbol v is generated in state S_i ($K \times P(S_i, v|O_1^p \ldots O_K^p, \lambda)$), and the expected number of times in state S_i ($K \times P(S_i|O_1^p \ldots O_K^p, \lambda)$):

$$P(S_i, S_j|O_1^p \ldots O_K^p, \lambda) = \sum_{v=0}^{1} P(S_i, v, S_j|O_1^p \ldots O_K^p, \lambda), \quad 0 \le i, j \le N-1 \tag{9}$$

$$P(S_i, v|O_1^p \ldots O_K^p, \lambda) = \sum_{j=0}^{N-1} P(S_i, v, S_j|O_1^p \ldots O_K^p, \lambda), \quad 0 \le i \le N-1, \ 0 \le v \le 1 \tag{10}$$

$$P(S_i|O_1^p \ldots O_K^p, \lambda) = \sum_{j=0}^{N-1} \sum_{v=0}^{1} P(S_i, v, S_j|O_1^p \ldots O_K^p, \lambda), \quad 0 \le i \le N-1 \tag{11}$$

The resulting equations to calculate the reestimated model $\overline{\lambda} = \{\overline{A}, \overline{B}, \overline{\pi}\}$, which will be used for the non-interleaver decoder in the next turbo decoding iteration are:

$$\overline{a_{ij}} = P_t(S_j|S_i) = \frac{P(S_i, S_j|O_1^p \dots O_K^p, \lambda)}{P(S_i|O_1^p \dots O_K^p, \lambda)} = \frac{\sum_{k=1}^{K} \sum_{e \in C_1} \alpha_{k-1}[s^S(e)] a_e P[O_k^p|e]\, \beta_k[s^E(e)]}{\sum_{k=1}^{K} \sum_{e \in C_2} \alpha_{k-1}[s^S(e)] a_e P[O_k^p|e]\, \beta_k[s^E(e)]} \tag{12}$$

$$\overline{b_i(v)} = P_o(v|S_i) = \frac{P(S_i, v|O_1^p \dots O_K^p, \lambda)}{P(S_i|O_1^p \dots O_K^p, \lambda)} = \frac{\sum_{k=1}^{K} \sum_{e \in C_3} \alpha_{k-1}[s^S(e)] a_e P[O_k^p|e]\, \beta_k[s^E(e)]}{\sum_{k=1}^{K} \sum_{e \in C_2} \alpha_{k-1}[s^S(e)] a_e P[O_k^p|e]\, \beta_k[s^E(e)]}, \tag{13}$$

with $C_1 = \{e : [s^S(e)]^{(s)} = S_i, [s^E(e)]^{(s)} = S_j\}$, $C_2 = \{e : [s^S(e)]^{(s)} = S_i\}$, $C_3 = \{e : u(e) = v, [s^S(e)]^{(s)} = S_i\}$, $0 \le i, j \le N-1$ and $0 \le v \le 1$.

Although it is also possible to reestimate the initial state, we have skipped this step in the previous equations. The reason is that since the block length is large enough the degradation resulting from this omission is practically negligible. In practice, instead of calculating the values of $\overline{a_{ij}}$ and $\overline{b_i(v)}$ using the right hand side of equations (12)(13), it is faster to calculate the joint probability defined in (8) and then use equations (9)(10)(11) directly to obtain the reestimated values.

V. Simulation results

In order to study the behavior of the proposed method, two hidden Markov sources with entropy rates H on the order of .7 and .9 bits/symbol are considered. Both sources consist of $N = 2$ states. In order to study only the improvement due to the use of the hidden Markov structure, the sequences produced by these sources satisfy $P(u_k = 0) = P(u_k = 1) = .5$. These sequences were input to a rate 1/3 encoder that included a systematic bit and two identical recursive 8-state convolutional encoders with generator matrix $G(D) = \frac{1+D+D^2+D^3}{1+D^2+D^3}$ and an interleaver with length 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.

Figure 2 shows the decoded bit error rate as a function of iteration number for several different values of E_b/N_0 (where the noise is Gaussian and white) for a decoder incorporating the changes described in the paper when a source with transition probabilities $P_t(S_0|S_0) = .1$, $P_t(S_1|S_1) = .15$ and output probabilities $P_o(0|S_0) = .95$, $P_o(1|S_1) = .925$ is used. Both the performance of the decoding method when the parameters of the source and the noise in the channel are perfectly known as well as the results when this information is not available at the beginning of the iterative decoding (the value of E_b/N_0 is estimated after each turbo decoding iteration) are shown. For this source (H=.73 bits/symbol), the theoretical unconstrained limit for E_b/N_0 using rate 1/3 codes is approximately -2.2 dB. The modified decoder performs within about 1.2 dB of this limit, showing convergence at $E_b/N_0 = -1.0$ dB. A decoder without the modifications described here fails to converge even at an E_b/N_0 of .1 dB. The lack of initial knowledge about the source parameters and the noise in the channel does not seem to increase the E_b/N_0 required for convergence. The only negative effect is that more iterations are necessary to achieve convergence. It is important to remark that when the hidden Markov model is not available to the decoder, we assume in our simulations that this lack of information holds for all the blocks. In other words, we allow the source to be different for different input blocks, which implies that for each input block the estimation of the source has to be performed again with no information available. This is an extremely conservative method of doing the decoding. In practice, if the source is changing slowly enough the Markov parameters are likely to be correlated across adjacent blocks, and this information could be used to supply the initial estimate for all blocks except the first.

Similar results can be seen in Figure 3, which corresponds to a hidden Markov source with entropy rate H=.9 bits/symbol, transition probabilities $P_t(S_0|S_0) = .99$, $P_t(S_1|S_1) = .989$ and output probabilities $P_o(0|S_0) = .71$, $P_o(1|S_1) = .73$. In contrast with the previous source which was characterized by fast changes between states, this model stays in each state for longer periods. For this source, the theoretical (unconstrained) limit for E_b/N_0 using rate 1/3 codes is approximately -1.1 dB. The modified decoder performs within about 1 dB of this limit, showing convergence at $E_b/N_0 = -.1$ dB.

Although we have only presented plots for two sources here, we have performed simulations for many other sources and have found in all cases that there is no penalty in E_b/N_0 incurred by the need to perform source estimation. We have also observed that the gap between the theoretical limit on E_b/N_0 and the lowest E_b/N_0 at which convergence is achieved is dependent on the entropy of the input sequence generated by the hidden Markov source, and it increases as the entropy decreases. This behavior is expected, since we are exploiting the Markov structure in the noninterleaved decoder but not in the interleaved decoder. Therefore, when the entropy is small and E_b/N_o is decreased, the performance of the second decoder tends to degrade, worsening the convergence of the whole system.

VI. Conclusions

We have introduced a generalization of decoding of parallel concatenated codes that allows joint source-channel coding of hidden Markov sources. We have also provided a method for estimating the parameters of the hidden Markov model during the decoding. This enables the joint source-channel coding methods to be applied with no apparent loss of performance even when the source parameters are initially unknown. As expected, simulation shows a strong improvement with respect to E_b/N_0 over the case in which source information is not considered. Provided that at least one constituent decoder operates on the non-interleaved input, the techniques we presented here can be applied to parallel concatenated codes with any number of constituent coders, although the performance will be better when the number of constituent coders is lower. Though we have focused here on combining source Markov structure with turbo coding, it is possible to use a similar approach to design coders for hidden Markov channels [13].

References

[1] J. Hagenauer, "Source-Controlled Channel Decoding," *IEEE Trans. on Communications*, pp. 2449-2457, September 1995.

[2] F. Alajaji, N. Phamdo, T. Fuja, "Channel Codes that Exploit the Residual Redundancy in CELP-Encoded Speech," *IEEE Trans. on Speech and Audio Processing*, pp. 325-336, September 1996.

[3] 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.

[4] C. Berrou, A. Glavieux, "Near Optimum Error Correcting Coding and Decoding: Turbo-Codes," *IEEE Trans. on Communications*, pp. 1261-1271, October 1996.

[5] J. Hagenauer, E. Offer, and L. Papke, "Iterative Decoding of Binary Block and Convolutional Codes," *IEEE Trans. on Information Theory*, pp. 429-445, March 1996.

[6] 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.

[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] 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.

[9] J. Garcia-Frias and J. Villasenor, "Combining hidden Markov Source Models and Parallel Concatenated Codes," *IEEE Comm. Letters*, pp. 111-113, July 1997.

[10] L. E. Baum and T. Petrie, "Statistical Inference for Probabilistic Functions on Finite State Markov Chains," *Ann. Math. Stat.*, Vol.37, pp. 1554-1563, 1966.

[11] L. E. Baum and G. R. Sell, "Growth Functions for Transformations on Manifolds,"*Pac. J. Math.*, Vol.32, No. 2, pp. 211-227, 1968.

[12] J. Garcia-Frias and P. M. Crespo, "Hidden Markov Models for Burst Error Characterization in Indoor Radio Channels," *IEEE Trans. on Vehicular Technology*, pp. 1006-1020, November 1997.

[13] J. Garcia-Frias and J.D. Villasenor, "Joint Source-Channel Decoding of Turbo Codes," *Proceedings of the International Symposium on Turbo Codes and Related Topics*, pp. 259-262, Brest, France, September 1997.

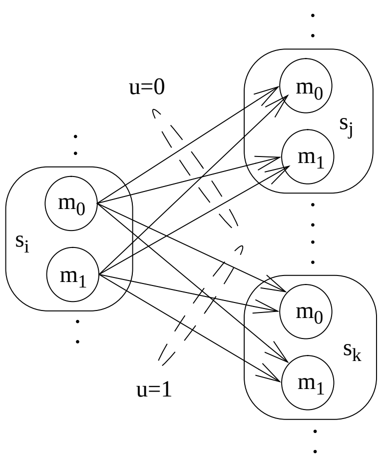

Figure 1: Supertrellis describing the non-interleaved encoder (states s) and the hidden Markov source (states m). u is the information bit.

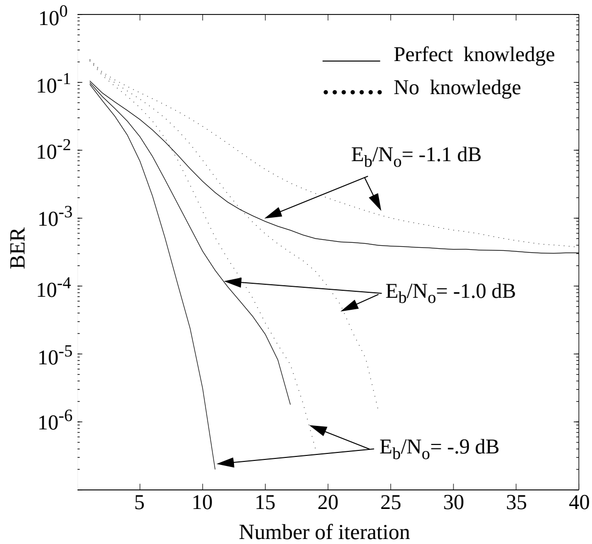

Figure 2: BER as a function of the decoding iteration number for the hidden Markov source with entropy rate $H = .73$ and channel decoder described in Section V. The solid lines represent the performance when perfect knowledge of the parameters of the source is available at the decoder. The dotted lines illustrate the performance when no knowledge of the parameters of the source is initially available to the decoder. The convergence is slower when the source parameters need to be estimated, but the final BER that is obtained is the same.

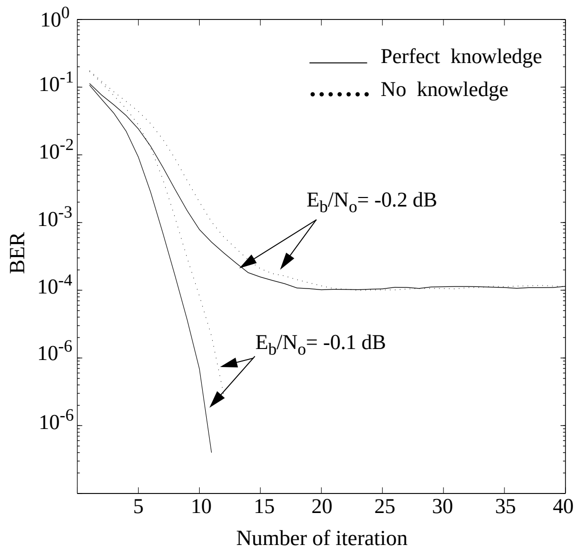

Figure 3: BER as a function of the decoding iteration number for the hidden Markov source with entropy rate $H = .9$ and channel decoder described in Section V. The solid lines represent the performance when perfect knowledge of the parameters of the source is available in the decoder. The dotted lines illustrate the performance when no knowledge of the parameters of the source is initially available to the decoder.