EXPLOITING BINARY MARKOV CHANNELS WITH UNKNOWN PARAMETERS IN 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 over binary-input, binary-output hidden Markov channels when the parameters of the Markov channel are unknown *a priori*. Specifically, we develop a joint estimation/decoding method that allows the estimation of the parameters of the model without the need for training sequences. This method involves little or no sacrifice in performance relative to the case where the Markov channel parameters are provided to the receiver as *a priori* information. Furthermore, we show communication at rates which are above the capacity of a memoryless channel with the same stationary bit error probability as the Markov channel, thereby outperforming systems based on the traditional approach of using a channel interleaver to create a channel which is assumed to be memoryless.**

I. Introduction

Many practical communications channels can be modeled using discrete Markov channels. Such channels are characterized by a set of states $S_j, 0 \leq j \leq S-1$, the matrix of transition probabilities among states (A=(a_{ij}), with a_{ij} the probability of transition from state S_i to state S_j, i.e. $a_{ij} = P_t(S_j|S_i), 0 \leq i,j \leq S-1$), and the list giving the bit error probability (BER) to associate with each state (B=$(b_j(v))$, with $b_j(v)$ the probability of getting the output v in state S_j, i.e. $b_j(v) = P_o(v|S_j), 0 \leq j \leq S-1,\ v \in \{0,1\}$). It is intuitive that the presence of memory in Markov channels leads to increased capacity relative to memoryless channels with the same stationary BER. Markov channel capacity was treated for the special case of Gilbert-Elliot channels by Mushkin and Bar-David in [1] and more generally by Goldsmith and Varaiya in [2]. In terms of the notation introduced in [1], the "capacities" that can be associated with a Markov channel include the true capacity C^{μ} and the capacity C^{NM} ("no memory") of a memoryless channel with the same stationary BER as the Markov channel. Clearly, $C^{NM} \leq C^{\mu}$.

In practice, many communications systems make use of a channel interleaver to distribute the errors so that codes designed for a memoryless channel can be used. While the application of interleaving does not change the capacity of the channel, the achievable performance of a decoder which assumes that the channel is memoryless is limited by C^{NM}. Exploiting the higher capacity of Markov channels in practice has proven to be challenging. Both [1] and [2] utilize decision feedback decoders which perform recursive state estimations that are used in the decoding process. However, the recursions are vulnerable to error propagation, and the decision feedback decoder can not be reliably used when the quality of the channel degrades. In [3] and [4] we proposed modifications of the decoding of parallel concatenated codes, or "turbo" codes [5], [6], [7], [8], [9] that exploit the structure of Markov channels and allow communication at rates $> C^{NM}$. No channel interleaver is used and the decoder exploits only the *a priori* structure of the channel as opposed to performing recursive state estimation. In this paper we extend that work by considering turbo decoding when the parameters of the Markov channel are not known *a priori*. A joint estimation/decoding method that allows to estimate the Markov channel parameters jointly with the turbo decoding is introduced and its performance is compared with the case in which perfect knowledge about the Markov channel parameters is available.

II. Decoder Modifications for Markov Channels

We consider the case of a parallel concatenated coder with $N+1$ constituent convolutional encoders and N interleavers [10]. The constituent decoders are enumerated by D_l, $0 \leq l \leq N$, and each is associated with an encoder of rate $1/n_l$. At the receiver, we refer to the constituent decoder operating on the observations associated with the non-interleaved input sequence as the "non-interleaved decoder", D_0; this contrasts with the N "interleaved decoders" D_l, $1 \leq l \leq N$, that operate on the observations corresponding to the interleaved inputs. The kth input bit (before interleaving) is denoted by u_k, $k = 1 \ldots L$, and can take on values i, $i = 0,1$. $\pi_l(k)$ describes interleaving applied to input bit u_k at the lth constituent encoder; π_0 denotes the identity operation (no interleaving). The observations are denoted by the matrix $\mathbf{O}$ in which O_k^l represents the observations at the channel output associated with the lth constituent coder and input bit $u_{\pi_l^{-1}(k)}$ (i.e. the bit mapped to position k by interleaver π_l). Note that each O_k^l contains n_l elements and therefore may itself be a vector. In discussing the decoding iterations it is necessary to distinguish between the observations O_k^p associated with the decoder D_p (and input bit $u_{\pi_p^{-1}(k)}$) in which processing is occurring and the observations $O_k^{\bar{f}_p} = [O_k^0 \ldots O_k^{p-1} O_k^{p+1} \ldots O_k^N]$ in the other constituent decoders. $\alpha_k(s)$ represents, for the forward trellis recursion, $P(O_1^p \ldots O_k^p, s_k = s)$, the probability of the observations in decoder D_p due to all inputs up to time k and that the trellis is in state s after the kth transition. $\beta_k(s)$ represents the probability $P(O_{k+1}^p \ldots O_K^p | s_k = s)$ as calculated using the backward recursion. 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 and coded bits associated with a branch e are denoted by $u(e)$ and $O(e)$, respectively.

The forward and backward recursions to use in turbo codes are based on [11] and can be expressed as:

$$\alpha_k(s) = \sum_{e:s^E(e)=s} \alpha_{k-1}\left[s^S(e)\right] \times$$

$$\times P_k\left[e|O_1^{\bar{f}_p} \ldots O_K^{\bar{f}_p}, s^S(e)\right] P\left[O_k^p|e\right]. \quad (1)$$

$$\beta_k(s) = \sum_{e:s^S(e)=s} \beta_{k+1}\left[s^E(e)\right] \times$$

$$\times P_{k+1}\left[e|O_1^{\bar{f}_p} \ldots O_K^{\bar{f}_p}, s^S(e)\right] P[O_{k+1}^p|e]. \quad (2)$$

Writing the equations in this way explicitly shows the flow of information (via the center term in the summation) between constituent decoders.

In the non-interleaved decoder D_0 the probability that the kth input bit u_k is equal to i, i=0 or 1 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.:

$$P(u_k = i|O_1^p \ldots O_K^p) = \frac{1}{P\left(O_1^p \ldots O_K^p\right)} \times$$

$$\times \sum_{e:u(e)=i} \alpha_{k-1}\left[s^S(e)\right] P\left[e|s^S(e)\right] P\left[O_k^p|e\right] \beta_k\left[s^E(e)\right]. \quad (3)$$

In decoders D_l, $l \neq 0$ operating on interleaved inputs, u_k in equation (3) should be replaced by $u_{\pi_l^{-1}(k)}$.

Although the decoder modifications to incorporate channel statistics can be applied to general finite state Markov channels with S states, for simplicity we consider the special case of Gilbert-Elliot channels ($S = 2$). A Gilbert-Elliot channel contains a "good" state (state S_0) in which the probability of bit error is $P_o(1|S_0) = P_G$ and a "bad" state (state S_1) with error probability $P_o(1|S_1) = P_B$. The transition probability of going from the good state to the bad state is $P_t(S_1|S_0) = b$ and the probability of going from the bad state to the good state is $P_t(S_0|S_1) = g$. The capacity of Gilbert-Elliot channels is described in [1]. It is also helpful to introduce, as in [1], a parameter $\mu = 1 - g - b$, which can range from -1 to 1, and serves as a measure of the "strength" of the memory of the Markov channel.

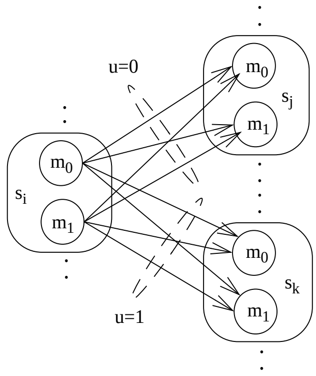

Fig. 1. Joint decoder trellis describing the encoder (states s) and the Markov channel (states m). u is the information bit.

In the following discussion, we assume that each of the constituent encoders comes from the puncturing of a rate 1 coder; i.e. that O_k^p is only defined for the non-punctured trellis transitions and in this case it consists only of 1 bit. As shown in [4], this is generally the case for the turbo encoders that result in the best performance for a wide range of channels and overall rates. If the rate of a constituent encoder is less than 1, the equations below should be modified accordingly, since 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 channel). In contrast with the AWGN channel, in a Markov channel the order of transmission of the bits becomes important. In order to exploit the hidden Markov structure of the errors introduced by the channel, we perform transmission of bits from each constituent encoder as a single contiguous block. In each constituent decoder, this allows construction of a supertrellis which jointly describes the channel and encoder, with each state in the coder becoming a superstate containing all possible channel states combinations. As shown in Figure 1, for rate ≥ 1 constituent encoders, the number of states in the supertrellis is the product of the number of states in the binary Markov channel 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 Markov channel.

Each one of the branches e in the supertrellis will have an associated *a priori* transition probability, a_e, which depends on the transition probabilities of the hidden Markov channel according to $a_e = P(e|s^S(e)) = P_t\left([s^E(e)]^{(m)} \| [s^S(e)]^{(m)}\right)$. Equations (1,2,3) can still be applied over each one of the supertrellises. Since the initial and final states

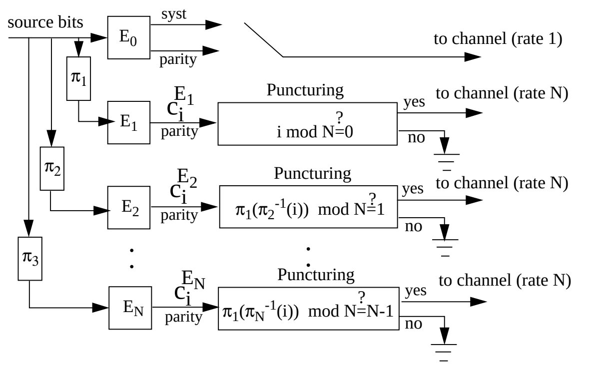

Fig. 2. Encoder structure used for rate 1/2 turbo coders. E_0 is a rate 1 encoder which produces only systematic bits in the implementation used in this paper, but which in the general case can output any rate 1 combination of systematic and parity bits. Interleavers are denoted by π_i. Encoders E_1, E_2 .. E_N each have rate 1 before puncturing, and rate N after puncturing using the puncturing function given in the figure. Taken together, encoders E_1, E_2 .. E_N produce a rate 1 code. $c_i^{E_j}$ represents the parity bit associated with trellis transition i in the code E_j.

of the Markov model are not known, equations (1) and (2) for all constituent decoders are initialized using $\alpha_0(s) = 1/S$ if $s^{(c)} = 0$ and $\beta_K(s) = 1/S$ if $s^{(c)} = 0$. For all other states $\alpha_0(s)$ and $\beta_K(s)$ are equal to 0. It is also important to note that the value of $P\left[O_k^p|e\right]$ to be used in equations (1,2,3) depends on the branch (e), received coded bit (O_k^p) and coded bit associated to the corresponding branch $(O(e))$ through the equation $P\left[O_k^p|e\right] = P_o(O_k^p \oplus O(e)|[s^S(e)]^{(m)})$, where $\oplus$ is the modulo-2 operation. The value of $P_k[e|O_1^{\bar{f}_p} \ldots O_K^{\bar{f}_p}, s^S(e)]$ for use in equations (1) and (2) (when applied to the constituent decoder D_p) can be calculated from the equation $P_k[e|O_1^{\bar{f}_p} \ldots O_K^{\bar{f}_p}, s^S(e)] P_k[O_1^{\bar{f}_p} \ldots O_K^{\bar{f}_p}|s^S(e)] = P_k[O_1^{\bar{f}_p} \ldots O_K^{\bar{f}_p}|e, s^S(e)] P[e|s^S(e)]$. After some manipulation, and assuming that the source sequence u_k is i.i.d. with probabilities $P(u_k = 0) = P(u_k = 1) = .5$, this gives

$$P_k[e|O_1^{\bar{f}_p} \ldots O_K^{\bar{f}_p}, s^S(e)] =$$

$$= J_k a_e \prod_{i=0, i\neq p}^{N} P[u_{\pi_p^{-1}(k)} = u(e)|O_1^i \ldots O_K^i], \quad (4)$$

where J_k is a normalization factor.

There are a number of important issues regarding the optimum number of constituent coders to associate with a particular rate and channel that were described in [4]. In this paper we will assume that this number is chosen adequately, thereby fixing the structure of the encoder, and we focus on the description of the method to perform joint estimation/decoding for Markov channels with unknown parameters, comparing the performance of this scheme with the method proposed in [4] when the parameters of the Markov channel are known *a priori*. The structure of the turbo encoders used in this paper is shown in Figure 2.

III. Joint Channel Estimation and Decoding

Parameter estimation for hidden Markov models has been described before, most notably in conjunction with the Baum-Welch algorithm [12] which has been widely used in speech processing, but also for characterizing mobile channels 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 noise-free output of the hidden Markov model, while in our case the output of the hidden Markov model that we want to estimate is added to the coded bits, which come from another independent Markov model (convolutional encoder).

Since the hidden Markov channel model is included in the supertrellis of each one of the constituent decoders, its parameters can be estimated simultaneously with decoding using the same underlying ideas as in the Baum-Welch algorithm. In fact, it is possible to apply an estimation algorithm over the supertrellis of each one of the constituent decoders so that each iteration in every constituent turbo decoder also results in an iteration on the parameter estimation.

In deriving the estimation algorithm we use a channel characterized by a hidden Markov model $\lambda = \{A, B, \pi'\}$, where A, B and π' are respectively the transition matrix probability, the output matrix probability and the initial distribution probability. Therefore, we can calculate the value of a_e for each branch in the supertrellises as indicated in the previous section.

The probability of going through branch e in the trellis section k has been calculated before and is given by:

$$P_k(e|O_1^p \ldots O_K^p, \lambda) =$$

$$= \frac{1}{P\left(O_1^p \ldots O_K^p\right)} \alpha_{k-1}\left[s^S(e)\right] a_e P\left[O_k^p|e\right] \beta_k \left[s^E(e)\right]. \quad (5)$$

Since branch e can be determined by the initial state $s^S(e)$, final state $s^E(e)$ and its associated error pattern $O_k^p \oplus O(e)$, we can express the probability of the branch e given the observation as the joint probability of $s^S(e), O_k^p \oplus O(e)$ and $s^E(e)$, i.e. $P_k(e|O_1^p \ldots O_K^p, \lambda) = P_k(s^S(e), O_k^p \oplus O(e), s^E(e)|O_1^p \ldots O_K^p, \lambda)$. Therefore, for the Markov channel the joint probability of the initial state S_i, final state S_j and error pattern $v \in \{0, 1\}$, where $v = 1$ corresponds to a channel error, can be expressed as:

$$P(S_i, v, S_j|O_1^p \ldots O_K^p, \lambda) =$$

$$= \frac{1}{K}\sum_{k=1}^{K}\sum_{e\in C} P_k(e|O_1^p \dots O_K^p, \lambda), \qquad (6)$$

where $0 \leq i,j \leq S-1$, $v \in \{0,1\}$ and $C = \{e : O_k \oplus O(e) = v, [s^S(e)]^{(m)} = S_i, [s^E(e)]^{(m)} = S_j\}$.

The product of this probability times the number of trellis transitions represents the expected number of transitions (for the Markov channel) from state S_i to state S_j generating the error pattern v. From this joint probability it is easy to calculate the expected number of transitions from state S_i to state S_j ($K \times P(S_i, S_j|O_1^p \dots O_K^p, \lambda)$), the expected number of times in state S_i generating error pattern v ($K \times P(S_i, v|O_1^p \dots O_K^p, \lambda)$), and the expected number of times in state S_i ($K \times P(S_i|O_1^p \dots O_K^p, \lambda)$):

$$P(S_i, S_j|O_1^p \dots O_K^p, \lambda) =$$

$$= \sum_{v=0}^{1} P(S_i, v, S_j|O_1^p \dots O_K^p, \lambda), \;\; 0 \leq i,j \leq S-1 \qquad (7)$$

$$P(S_i|O_1^p \dots O_K^p, \lambda) =$$

$$= \sum_{j=0}^{S-1}\sum_{v=0}^{1} P(S_i, v, S_j|O_1^p \dots O_K^p, \lambda), \;\; 0 \leq i \leq S-1 \qquad (8)$$

$$P(S_i, v|O_1^p \dots O_K^p, \lambda) = \sum_{j=0}^{S-1} P(S_i, v, S_j|O_1^p \dots O_K^p, \lambda), \qquad (9)$$

with $0 \leq i \leq S-1$, $v \in \{0,1\}$.

Therefore, the process of estimating the parameters of the hidden Markov channel can be performed as part of the decoding processing on the supertrellises describing the hidden Markov channel and each of the constituent decoders. The resulting equations to calculate the estimated model $\overline{\lambda} = \{\overline{A}, \overline{B}, \overline{\pi'}\}$, in the corresponding constituent decoder (which will be used as initial model for the next constituent decoder) are:

$$\overline{a_{ij}} = P_t(S_j|S_i) = \frac{P(S_i, S_j|O_1^p \dots O_K^p, \lambda)}{P(S_i|O_1^p \dots O_K^p, \lambda)} =$$

$$= \frac{\sum_{k=1}^{K}\sum_{e\in C_1} \alpha_{k-1}[s^S(e)] a_e P[O_k^p|e]\, \beta_k[s^E(e)]}{\sum_{k=1}^{K}\sum_{e\in C_2} \alpha_{k-1}[s^S(e)] a_e P[O_k^p|e]\, \beta_k[s^E(e)]} \qquad (10)$$

$$\overline{b_i(v)} = P_o(v|S_i) = \frac{P(S_i, v|O_1^p \dots O_K^p, \lambda)}{P(S_i|O_1^p \dots O_K^p, \lambda)} =$$

$$= \frac{\sum_{k=1}^{K}\sum_{e\in C_3} \alpha_{k-1}[s^S(e)] a_e P[O_k^p|e]\, \beta_k[s^E(e)]}{\sum_{k=1}^{K}\sum_{e\in C_2} \alpha_{k-1}[s^S(e)] a_e P[O_k^p|e]\, \beta_k[s^E(e)]}, \qquad (11)$$

with $C_1 = \{e : [s^S(e)]^{(m)} = S_i, [s^E(e)]^{(m)} = S_j\}$, $C_2 = \{e : [s^S(e)]^{(m)} = S_i\}$, $C_3 = \{e : O_k^p \oplus O(e) = v, [s^S(e)]^{(m)} = S_i\}$, $0 \leq i,j \leq S-1$ and $v \in \{0,1\}$.

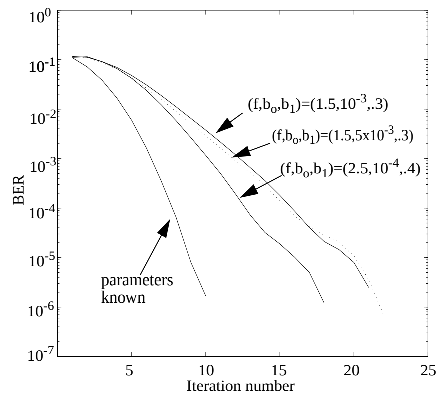

Fig. 3. Convergence behavior for the rate 1/2 turbo code and Markov channel with $\mu = .86$ when the *a priori* information about the parameters of the Markov channel is not known. The convergence for the case where the parameters are known is shown for comparison. The initial Markov channel is defined by the transition matrix $b^{in} = .0005$, $g^{in} = .0005f$ and the probabilities of error in the good and in the bad state ($P_G^{in} = b_0$ and $P_B^{in} = b_1$, respectively). The interleaver length in all figures is 16384.

In practice, instead of calculating the values of $\overline{a_{ij}}$ and $\overline{b_i(v)}$ using the right hand side of equations (10,11), it is faster to calculate the joint probability defined in (6) and then use equations (7,8,9) directly to obtain the estimated values. It is important to remark that the previous equations are valid for rate greater than or equal to 1 constituent encoders. For constituent encoders with rate greater than 1, all the sums in the index k should be carried out only for the trellis transitions which are associated with a coded bit O_k^p (i.e. the "holes" due to the puncturing should not be considered for the estimation process).

As it will be shown in next section, the convergence of this algorithm will depend on the Markov model used for its initialization. The initial Markov channel is defined by the transition matrix given by $P_t^{in}(S_1|S_0) = b^{in}$, $P_t^{in}(S_0|S_1) = g^{in}$ (where *in* denotes initial) and the probabilities of error in the good and in the bad state ($P_o^{in}(1|S_0) = P_G^{in}$ and $P_o^{in}(1|S_1) = P_B^{in}$, respectively).

IV. Simulation Results

In order to asses the performance of the joint estimation/decoding method, we consider two Markov channels. The first has parameters $P_G = .0128$, $P_B = .5$, $g = .1092$, and $b = .0308$ and $\mu = 1-g-b = .86$. The stationary bit error rate of this channel is $\rho = .12$. Note that this is higher than the bit error rate of $\rho = .11$ at which $C^{NM} = 1/2$ (i.e., $1 - H(.11) = 1/2$ where $H()$ is the binary entropy function). The parameters of the second Markov channel are $P_G = .01925$ and $P_B = .5$, with b and g chosen so that the ratio b/g is the same as in the channel above but with $\mu = 1-g-b = .98$. The stationary

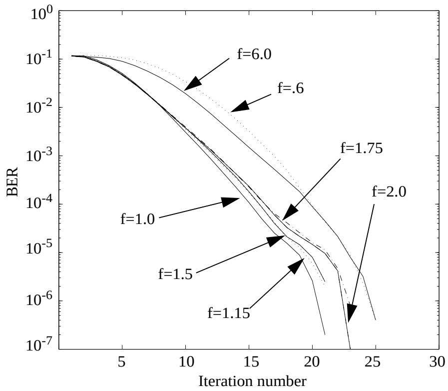

Fig. 4. Convergence behavior for the rate 1/2 turbo code and Markov channel with $\mu = .86$ when the *a priori* information about the parameters of the Markov channel is not known. The initial error probabilities are the same for all curves and correspond to the worst case in Figure 3 ($P_G^{in} = 10^{-3}$ and $P_B^{in} = .3$). The factor f is changed as indicated in the graph. The figure shows the similar performance obtained for a wide range of values of f (and consequently for a large range of the initial stationary error probability).

bit error rate is $\rho = .125$. This channel has stronger dependencies and therefore longer burst lengths than the previous one. As shown in [4], when the parameters of the hidden Markov model are known *a priori*, it is possible for both channels to build turbo encoders of rate 1/2 capable of performing decoding with a residual error probability less than the resolution of the simulation, which consisted of 10^7 bits. These encoders are of the type shown in Figure 2 (the total number of constituent coders is 4 for the $\mu = .86$ channel and 5 for the $\mu = .98$ channel). In all cases considered here, the interleaver length is 16384 and the generator polynomial of the all the constituent convolutional encoders is $G(D) = \frac{1+D+D^2+D^3}{1+D^2+D^3}$.

In our simulations the transition matrix of the initial Markov channel is always selected as $P_t^{in}(S_1|S_0) = b^{in} = .0005$, $P_t^{in}(S_0|S_1) = g^{in} = .0005 \times f$. The reason for choosing these small values is that we have observed that in the joint estimation decoding method it is better (in order to achieve convergence) to initialize with transition probabilities smaller than the real ones. Therefore, the selected values guarantee the ability to deal with very bursty channels. As a rule of thumb, parameters (f, P_G^{in}, P_B^{in}) must be chosen such that the stationary probability of error of the initial Markov channel is close to or higher than the one corresponding to C^{NM}. Figure 3 illustrates the performance of the proposed algorithm for different values of the parameters (f, P_G^{in}, P_B^{in}) for the channel with $\mu = .86$. We note [4] that increasing the value of P_G (and therefore ρ) for this channel causes the non-convergence of turbo decoding (even when the parameters of the Markov channel are known). Therefore, for this particular channel the lack of information of the Markov channel does not degrade the decoding performance (except in the number of iterations needed to convergence). Figure 4 shows the performance of the proposed algorithm for the worst case in Figure 3 when the factor f is changed (and correspondingly the value of ρ^{in}) but P_G^{in} and P_B^{in} are maintained ($P_G^{in} = 10^{-3}$, $P_B^{in} = .3$). We can see that the range of f in which convergence is achieved is very large, corresponding to initial stationary error probabilities in the range $\rho^{in} \in [.044, .19]$ (at least). This implies that for this channel the initialization of the Markov channel used in the estimation procedure is not critical at all and we do not need to chose the initial parameters very carefully. However, as we can see in Figures 5, 6 and 7, this initialization is quite more important for the channel with $\mu = .98$. As Figure 5 shows, in order to achieve convergence in the joint estimation/decoding method for a value of $\rho = .125$ (which as shown in [4] is the maximum value of ρ for which turbo decoding with known parameters can be achieved with residual probability of error less than 10^{-6}) it is necessary for the initial value P_G^{in} to be quite close to the real one. Although this seems to be a very stringent requirement, it can however be a reasonable assumption depending on the channel under consideration (for example if we know that errors in the channel are the result of packet losses, the initialization of $P_B^{in} = .5$ seems to be a good assumption). As shown in Figure 6, the range of parameters at which convergence can be achieved improves if P_G decreases producing an stationary error probability of $\rho = .12$. Figure 7 illustrates the performance of the proposed algorithm for the worst case in Figure 6 when the factor f is changed, but not P_G^{in} or P_B^{in} ($P_G^{in} = 2.5 \times 10^{-3}$, $P_B^{in} = .3$). We can see that in this case the range of f over which convergence is achieved is narrower than for the previous channel (initial stationary error probabilities in the range $\rho^{in} \in [.153, .175]$) and that for values of ρ^{in} inside this range the resulting performance is very similar. The worse performance for the channel with $\mu = .98$ seems to be due to the higher variance in the bit error probability for different realizations of the channel. Also, we should note that the block length needed to have good parameter estimates decreases with the value of μ.

In all our simulations, we have assumed that the lack of information holds for all the blocks. In other words, we allow the channel to be different for different input blocks, which implies that for each input block the estimation of the channel has to be performed again with no *a priori* information available. In practice, if the channel 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

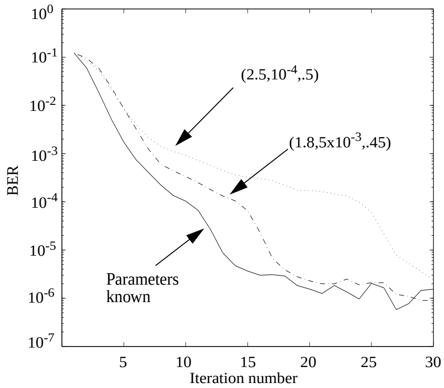

Fig. 5. Convergence behavior for the rate 1/2 turbo code and Markov channel with $\mu = .98$ and $\rho = .125$ when the *a priori* information about the parameters of the Markov channel is not known. The initial Markov channel is defined by the transition matrix $b^{in} = .0005$, $g^{in} = .0005f$ and the probabilities of error in the good and in the bad state ($P_G^{in} = b_0$ and $P_B^{in} = b_1$, respectively). This is represented in the figure by the 3-tuple (f, b_0, b_1).

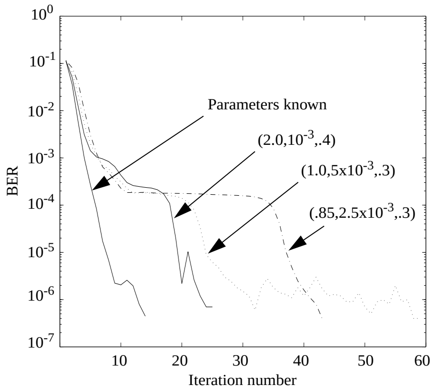

Fig. 6. Convergence behavior for the rate 1/2 turbo code and Markov channel with $\mu = .98$ and $\rho = .12$ when the *a priori* information about the parameters of the Markov channel is not known. As in the previous figures, the initial Markov channel is represented by the 3-tuple (f, b_0, b_1). The range of initial parameters which achieve convergence is much wider than in Figure 5.

blocks except the first, leading to much faster convergence.

V. Conclusions

We have introduced a joint estimation/decoding method for parallel concatenated codes over binary Markov channels without the need of transmission of training sequences. This method allows reliable communication at rates above the capacity C^{NM} of the corresponding interleaved and assumed memoryless channel and near C^{μ}, the true capacity of the channel, even when the parameters of the Markov channel are not known *a priori*. The performance degradation in comparison with the case in which the statistics of the Markov channel are known *a priori* is always very small, though it increases with the value of μ. Also,

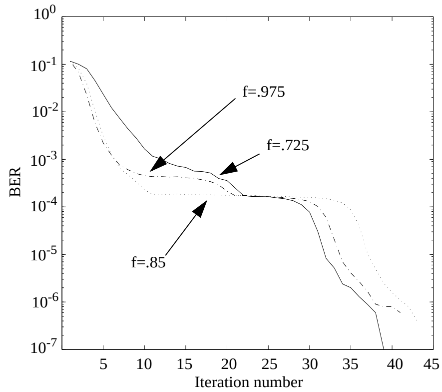

Fig. 7. Convergence behavior for the rate 1/2 turbo code and Markov channel with $\mu = .98$ and $\rho = .12$ when the *a priori* information about the parameters of the Markov channel is not known. The initial error probabilities are the same for all curves and correspond to the worst case in Figure 6 ($P_G^{in} = 2.5 \times 10^{-3}$ and $P_B^{in} = .3$). The parameter f is changed as indicated in the figure. Notice the similar convergence behavior for different values of f. However, the range for f in which convergence is achieved is much smaller than for the previous channel with $\mu = .86$.

the selection of the initial Markov model parameters becomes more critical as the value of μ increases.

References

[1] M. Mushkin and I. Bar-David, "Capacity and Coding for the Gilbert-Elliott Channels," *IEEE Trans. on Inf. Theory*, vol. 35, pp. 1277-1290, Nov. 1989.

[2] A.J. Goldsmith and P.P. Varaiya, "Capacity, mutual information, and coding for finite-state Markov channels," *IEEE Trans. on Inf. Theory*, vol. 42, pp. 868-886, May 1996.

[3] 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.

[4] J. Garcia-Frias and J. Villasenor, "Turbo Codes for Binary Markov Channels," *Proc. of the ICC*, pp. 110-115, June 1998.

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

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

[7] 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.

[8] S. Benedetto, G. Montorsi, "Unveiling Turbo Codes: Some Results on Parallel Concatenated Coding Schemes," *IEEE Trans. on Information Theory*, pp. 409-428, March 1996.

[9] L. C. Perez, J. Seghers, and D. Costello, "A Distance Spectrum Interpretation of Turbo Codes," *IEEE Trans. on Information Theory*, pp. 1698-1709, November 1996.

[10] D. Divsalar, F. Pollara, "Turbo Codes for PCS Applications," *Proc. of the ICC*, pp. 54-59, 1995.

[11] 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.

[12] 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.