

## TURBO CODES FOR BINARY MARKOV CHANNELS

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. We present encoder design techniques and decoder processing modifications that utilize the *a priori* statistics of the channel and show that the resulting codes allow reliable communication at rates which are above the capacity of a memoryless channel with the same stationary bit error probability as the Markov channel. These codes outperform systems based on the traditional approach of using a channel interleaver to create a channel which is assumed to be memoryless. In addition, we introduce a joint estimation/decoding method that allows the estimation of the parameters of the Markov model when they are not known *a priori*.

### 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, and the list giving the bit error rate (BER) to associate with each state. 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  $C^\mu$  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.

The main contributions of the present paper lie in encoder design and decoding methods for parallel concatenated codes, or “turbo” codes [3], [4], [5], [6], [7] that exploit the structure of Markov channels and allow communication at rates  $> C^{NM}$ . Our work contrasts with most previous approaches to Markov channels in that no channel interleaver is used and in that the decoder exploits only the *a priori* structure of the channel as opposed to performing recursive state estimation. As will be shown below, this allows good performance for poor channels. Our work also contrasts with much

of the work on turbo codes which until very recently has emphasized additive white Gaussian noise (AWGN) channels, though other types of channels are now getting more attention. In [8] we presented trellis modifications which allow a turbo decoder to exploit Markov structure in the source bits. In [9] we gave a single example of how turbo decoding could be modified to handle Markov channels. Here we focus in particular on two techniques that lead to effective communication over Markov channels: modification of the decoder structure to exploit the *a priori* channel statistics, and encoder design approaches that involve varying the number of constituent encoders and the puncturing scheme.

### II. DECODER MODIFICATIONS FOR MARKOV CHANNELS

#### A. Groundwork

We begin by formulating the decoding equations in a way that supports modification to incorporate channel statistics. For AWGN channels, this framework is identical to that usually used for turbo codes [10]. We consider the case of a parallel concatenated coder with  $N + 1$  constituent convolutional encoders and  $N$  interleavers [11]. 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  $k$ th input bit (before interleaving) is denoted by  $u_k, k = 1 \dots K$ , and can take on values  $i, i = 0, 1$ .  $\pi_l(k)$  describes interleaving applied to input bit  $u_k$  at the  $l$ th constituent encoder;  $\pi_0$  denotes the identity operation (no interleaving). Processing at the receiver is performed in one constituent decoder at a time and uses all available results from the other constituent decoders. The observations are denoted by the matrix  $\mathbf{O}$  in which  $O_k^l$  represents the observations at the channel output associated with the  $l$ th constituent coder and input bit  $u_{\pi_l^{-1}(k)}$  (i.e. the bit mapped to position  $k$  by interleaver  $\pi_l$ ). Note that each  $O_k^l$  contains  $n_l$  elements and therefore may itself be a vector. Holding the subscript constant and letting the superscript vary refers to all of the observations (in decoders  $D_0$  through  $D_N$ ) associated with the  $k$ th trellis transition. 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  $O_k^{\bar{p}} = [O_k^0 \dots O_k^{p-1} O_k^{p+1} \dots O_k^N]$  in the other constituent decoders. Holding the superscript constant and letting the subscript vary specifies observations associated with

a single decoder and multiple input bits; for example, the set of all observations for decoder  $D_l$  is  $[O_1^l \dots O_K^l]$ .  $\alpha_k(s)$  represents, for the forward trellis recursion,  $P(O_1^p \dots O_k^p | s_k = s)$ , 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  $k$ th transition.  $\beta_k(s)$  represents the probability  $P(O_{k+1}^p \dots O_K^p | s_k = s)$  as calculated using the backward recursion. As in [10], 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 forward backward algorithm described in [12] provides recursions that express  $\alpha_k(s)$  in terms of  $\alpha_{k-1}(s)$ , and  $\beta_k(s)$  in terms of  $\beta_{k+1}(s)$ . In terms of the notation above, the  $\alpha$  recursion as used in decoding of parallel concatenated codes can be written in a form that utilizes a summation over all edges  $e$  that terminate in state  $s$ ; i.e. that satisfy  $s^E(e) = s$ :

$$\alpha_k(s) = \sum_{e: s^E(e)=s} \alpha_{k-1} [s^S(e)] P_k [e | O_1^p \dots O_K^p, s^S(e)] P [O_k^p | e]. \quad (1)$$

By analogy with the case for the forward recursion, the backward recursion used to calculate  $\beta_k(s)$  can be expressed:

$$\beta_k(s) = \sum_{e: s^S(e)=s} \beta_{k+1} [s^E(e)] P_{k+1} [e | O_1^p \dots O_K^p, s^S(e)] P [O_{k+1}^p | e] \quad (2)$$

where the summation is performed over all edges  $e$  that begin in state  $s$ . Writing the equations in this way explicitly shows the flow of information (via the center term in the summation) between constituent decoders. As will be discussed below in connection with equation (4), this allows direct incorporation of *a priori* channel probabilities.

In the non-interleaved decoder  $D_0$  the probability that the  $k$ th 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.:

$$P(u_k = i | O_1^p \dots O_K^p) = \frac{1}{P(O_1^p \dots O_K^p)} \sum_{e: u(e)=i} \alpha_{k-1} [s^S(e)] \times P [e | s^S(e)] P [O_k^p | e] \beta_k [s^E(e)]. \quad (3)$$

In decoders  $D_l$ ,  $l \neq 0$  operating on interleaved inputs,  $u_k$  in equation (3) should be replaced by  $u_{\pi_l^{-1}(k)}$ . To achieve good performance it is necessary that each decoder perform an independent estimation of  $P(u_k = i | O_1^p \dots O_K^p)$ . For this reason, the probabilities  $P_k$  and  $P_{k+1}$  in equations (1) and (2) are functions of the observations in the other decoders, while the term  $P[e | s^S(e)]$  in equation (3) is not. This ensures that the “extrinsic” information is properly considered. The decoding iterations obtain successively refined estimates of  $P(u_k = i | O_1^p \dots O_K^p)$  via equations (1,2,3). As we showed in [8], it is possible to use the above equations as the basis for a modified turbo decoder that can exploit hidden Markov structure in a source.

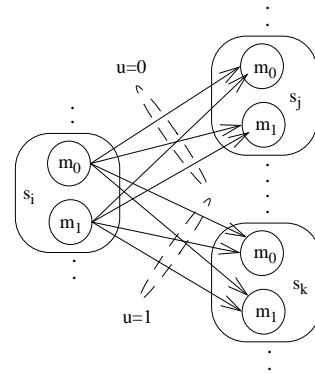


Fig. 1. Joint decoder trellis describing the encoder (states  $s$ ) and the Markov channel (states  $m$ ).  $u$  is the information bit.

### B. Incorporating Channel Statistics

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-Elliott channels ( $S = 2$ ). A Gilbert-Elliott 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-Elliott 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. When  $\mu$  approaches  $1$  the channel will tend to remain in one state for longer periods of time, creating long bursts of errors. (The less realistic case of  $\mu \rightarrow -1$  corresponds to a channel which tends to rapidly alternate states).

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, as contrasted with the interleaved transmission that is typically used. 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 state combinations. In constituent decoder  $D_l$  with rate  $1/n_l$  (where  $n_l$  is an integer), each superstate contains  $S^{n_l}$  channel states. For an  $S$ -state channel, a given starting coder superstate, and a given value of the input bit  $u_k = i$ ,  $i = 0$  or  $1$ , there are at most  $S^{2n_l}$  trellis edges, corresponding to all possible combinations of starting and ending channel states. 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. This is illustrated in Figure 1 for a constituent decoder for a rate 1 code and a 2 state channel. If the rate of the constituent encoder is equal to 1 or greater (i.e. by using puncturing) the value of  $O_k^p$  is only defined for the non-punctured

trellis transitions and in this case it consists of only 1 bit. Equations (1,2,3) can be applied over the supertrellis. Since the initial and final state of the Markov model are not known, equations (1) and (2) for the non-interleaver decoder are initialized using  $\alpha_0(s) = 1/S^{n_i}$  if  $s^{(c)} = 0$  and  $\beta_K(s) = 1/S^{n_i}$  if  $s^{(c)} = 0$ . For all other states  $\alpha_0(s)$  and  $\beta_K(s)$  are equal to 0. The value of  $P_k[e|O_1^{\bar{f}_p} \dots O_K^{\bar{f}_p}, s^S(e)]$  for use in equations (1) and (2) (when applied to the constituent decoder  $D_p$ ) can be calculated from the equation  $P_k[e|O_1^{\bar{f}_p} \dots O_K^{\bar{f}_p}, s^S(e)]P_k[O_1^{\bar{f}_p} \dots O_K^{\bar{f}_p}|s^S(e)] = P_k[O_1^{\bar{f}_p} \dots O_K^{\bar{f}_p}|e, s^S(e)]P[e|s^S(e)]$ . 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

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

where  $J_k$  is a normalization factor and  $a_e = P[e|s^S(e)]$  is the *a priori* transition probability of branch  $e$  in the joint trellis. As we assume that the source bits are i.i.d., the value of  $a_e$  depends only on the transition probabilities of the hidden Markov model representing the channel according to  $a_e = P[e|s^S(e)] = P_t([s^E(e)]^{(m)}|[s^S(e)]^{(m)})$ . It is also important to note that the value of  $P[O_k^p|e]$  to be used in equations (1,2,3) depends on the branch ( $e$ ), received coded bit ( $O_k^p$ ) and coded bit associated with the corresponding branch ( $O(e)$ ). For the case of constituent encoders of rate greater than or equal to 1, it can be expressed as  $P[O_k^p|e] = P_o(O_k^p \oplus O(e)|[s^S(e)]^{(m)})$ , where  $\oplus$  is the modulo-2 operation.

### III. ENCODER DESIGN AND EXPERIMENTAL RESULTS

In designing the encoder, we use puncturing to decouple the rate from the number of constituent coders. Figure 2 illustrates the framework for the rate 1/2 encoders used in this paper, though, as shown below, the approach extends to other rates as well. The top (noninterleaved) constituent coder  $E_0$  in the figure is rate 1. For the systems used here encoder  $E_0$  simply produced a systematic bit, although in a more general implementation it could output both systematic and coded bits which are punctured to rate 1. The interleaved constituent coders  $E_1$  through  $E_N$  each produce an output at rate 1 (i.e.  $n_i = 1$ ). These bits are then punctured (thereby requiring appropriate modifications at the decoder to equations (1,2,3)), such that the combined output from all of the interleaved coders has rate 1. The puncturing is performed with respect to the ordering of the bits at the input to coder  $E_1$ , as indicated by the "puncturing function" boxes shown in Figure 2. This contrasts with the puncturing in which the action of the interleavers is not considered. Referring the puncturing to the input of a single constituent coder is advantageous when the number of constituent coders is larger. The structure in Figure 2 has the advantage of enabling a rate 1/2 output using an arbitrarily high number of constituent coders. This allows customization to the

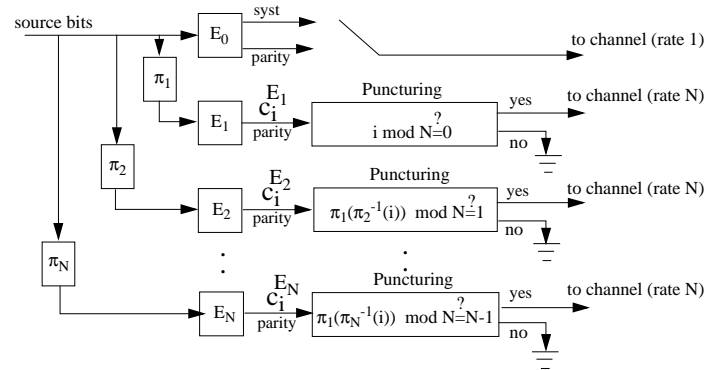


Fig. 2. Encoder structure used for rate 1/2 turbo coders.  $E_0$  is a rate 1 encoder which produces only systematic bits in the implementation used in this paper, but which in the general case can output any rate 1 combination of systematic and parity bits. Interleavers are denoted by  $\pi_i$ . Encoders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

expected channel error burst length, which depends on the channel parameter  $\mu$  introduced above. For strongly correlated channels that occur for  $\mu$  approaching 1, the bursts will tend to be long. Increasing the number of constituent coders gives an increased number of simultaneous interleaver realizations, thereby distributing the transmission times of source bits in a more diverse manner. The disadvantage of larger numbers of encoders is that the puncturing creates gaps in the trellises in the constituent decoders. Therefore, for weakly correlated channels it is better to use a reduced number of constituent encoders. The complexity of the processing at the decoder grows sublinearly with the number of constituent coders at the encoder.

Coding results are presented in Tables 1 and 2. Each row of the tables corresponds to a rate 1/2 encoder with a different number of interleaved constituent coders. For example, in the first row of Table 1, the encoder has a total of 3 constituent coders, 2 of which operate on interleaved inputs. The columns of the tables give the number of blocks and bits simulated (each block contained 16384 bits), the number of blocks containing 1 or more errors after decoding, and the corresponding residual bit error rate. The tables also give the average number of bit errors per block conditioned on blocks that contained one or more bit errors. Table 1 considers a Markov channel with 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). Thus, reliable communication over this channel using a rate 1/2 code is not possible in a traditional system that uses channel interleaving and does not exploit channel memory. However, as Table 1 shows, when 4 constituent coders are used, the residual error rate is below the resolution of the simulation, which used over  $10^7$  bits. By contrast using coders with 3 and 5 constituent coders gives residual BERs of  $1.5 \times 10^{-2}$  and  $3.6 \times 10^{-3}$  respectively. Note

Number of constituent coders (N+1)	Total number of blocks	Number of blocks in error	Residual BER	Average number of errors for the blocks in error
3	106	106	$1.5 \times 10^{-2}$	239
4	617	0	0	---
5	304	13	$3.6 \times 10^{-3}$	1396

Tab. I: Simulation results for three different rate 1/2 turbo codes (with 3, 4 and 5 constituent coders) over a binary Markov (Gilbert-Elliot) channel with parameters  $P_G = .0128$ ,  $P_B = .5$ ,  $g = .1092$ ,  $b = .0308$ . For this channel  $\mu = .86$  and the stationary bit error probability is  $\rho = .12$ . For a memoryless channel, capacity  $C^{NM}$  occurs for  $\rho = .11$ . Thirty decoding iterations were performed for all simulations. The block size is 16384 bits.

Number of constituent coders (N+1)	Total number of blocks	Number of blocks in error	Residual BER	Average number of errors for the blocks in error
4	152	57	$6.2 \times 10^{-4}$	27
5	632	1	$1.5 \times 10^{-6}$	16
6	617	7	$1.5 \times 10^{-3}$	2211

Tab. II: Simulation results for three different rate 1/2 turbo codes (with 4, 5 and 6 constituent coders) over a binary Markov (Gilbert-Elliot) channel with parameters  $P_G = .01925$ ,  $P_B = .5$ ,  $g = .0156$ ,  $b = .0044$ . For this channel  $\mu = .98$  and  $\rho = .125$ . Thirty decoding iterations were performed for all simulations. The block size is 16384 bits.

that “residual BER” is used to describe the errors remaining in the information bits after coding, transmission, and decoding. This contrasts with the use of “BER” to describe the error rate applied by the channel. Table 2 considers a Markov channel with  $P_G = .01925$  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 bit error rate is  $\rho = .125$ . This channel has stronger dependencies and therefore longer burst lengths than the channel of Table 1. Table 2 considers rate 1/2 encoders with 4, 5, and 6 constituent coders respectively, and clearly illustrates that only the encoder with 5 constituent coders can provide reliable communication. The results in these table show that for channels with higher  $\mu$ , the “optimum” number of constituent coders is higher.

Note that while the residual BERs for 4 and 6 constituent coders in Table 2 are of the same order of magnitude, the system with 6 coders has only a few blocks containing errors, but the number of bit errors within these blocks is high. The 4-encoder system has many more blocks in error, but with far fewer errors per block. A similar effect can be seen in Table 1 for the cases of 3 and 5 constituent coders. More generally, we have observed that if too few constituent coders are used, there will be more block errors, but a lower bit error rate within errored blocks. When too many constituent coders are used the opposite effect occurs. Figures 3 and 4 illustrate the convergence behavior when the channel degrades. All curves in Figure 3 correspond to  $\mu = .86$ . By making the error rate  $P_G$  in the “good” state worse, one increases the stationary bit error rate,  $\rho$ , of the Markov channel while holding  $\mu$  constant. As noted above, the capacity for a memoryless channel with  $\rho = .11$  is  $C^{NM} = 1/2$ . For the Gilbert-Elliot

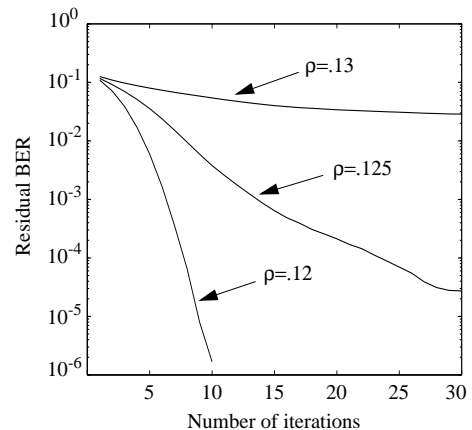


Fig. 3. Convergence behavior for a rate 1/2 turbo code with 4 constituent coders ( $N=3$ ) over a binary Markov (Gilbert-Elliot) channel with parameters  $P_B = .5$ ,  $g = .1092$ ,  $b = .0308$  ( $\mu = .86$ ) and a variable  $P_G$  resulting in stationary bit error probability  $\rho$ . For this channel the  $\rho$  giving a capacity  $C^\mu = 1/2$  is  $\rho = .137$ . For a memoryless channel, capacity  $C^{NM}$  occurs for  $\rho = .11$ . The vertical axis gives the residual BER after decoding.

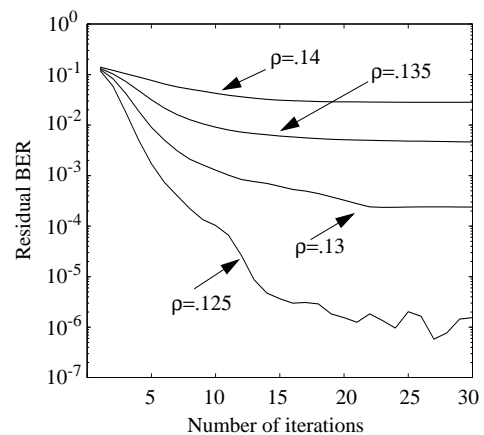


Fig. 4. Convergence behavior for a rate 1/2 turbo code with 5 constituent coders ( $N=4$ ) over a binary Markov (Gilbert-Elliot) channel with parameters  $P_B = .5$ ,  $g = .0156$ ,  $b = .0044$  ( $\mu = .98$ ) and a variable  $P_G$  resulting in stationary bit error probability  $\rho$ . For this channel the  $\rho$  giving a capacity  $C^\mu = 1/2$  is  $\rho = .154$ . For a memoryless channel, capacity  $C^{NM}$  occurs for  $\rho = .11$ . The vertical axis gives the residual BER after decoding.

channel discussed above with  $P_B = .5$ ,  $g = .1092$ ,  $b = .0308$  and  $\mu = 1 - g - b = .86$ , the value of the stationary bit error probability  $\rho$  corresponding to  $C^\mu = 1/2$  is  $.137$ . In other words, if we increase  $P_G$  until the stationary bit error rate reaches  $.137$ , then even a communications system that exploits channel dependencies will be unable to achieve reliable communication at rate 1/2. The curves in Figure 3 show that the approach we are using is successful for Gilbert-Elliot channels with  $\rho = .12$  (which is above the memoryless capacity  $C^{NM}$ ), but fails rather abruptly when  $\rho$  is raised to  $.13$ . Figure 4 gives a corresponding set of curves for the Markov channel discussed previously with  $\mu = .98$ . In this case the value of  $\rho$  corresponding to  $C^\mu = 1/2$  is  $\rho = .154$ . In contrast with Figure 3, in Figure 4 the degradation in performance as  $\rho$  increases is much smoother. This is partly due to the higher variance in the bit error probability for

different realizations of the channel with  $\mu = .98$ .

The above results show that channels with greater  $\mu$  lead to inferior performance relative to the capacity  $C^\mu$ . In other words, when  $\mu$  is higher, there is a greater gap between the stationary bit error probability corresponding to  $C^\mu$  and the stationary bit error probability at which convergence is achieved. This gap can be reduced by decreasing the rate of the code, since that causes the variance of the bit error probability for different realizations of the channel to decrease. For rate  $1/k$  codes the turbo encoder consists of a rate 1 encoder ( $E_0$  as described in Figure 2) and  $k-1$  blocks. Block  $i$ , ( $1 \leq i \leq k-1$ ) comprises  $N_i$  constituent encoders and has the same structure as the block consisting of constituent encoders  $E_1, \dots, E_N$  in Figure 2. As a rule of thumb, the optimum value for  $N_i$  decreases with the rate of the code, and for fixed rate it decreases with the value of  $\mu$ . For rate  $1/3$  codes, for which the bit error rate corresponding to  $C^{NM}$  is  $\rho = .174$ , we have considered two Markov channels. The first channel has parameters  $P_G = .0252$ ,  $P_B = .5$ ,  $g = .0914$ , and  $b = .0486$ , which produces  $\mu = 1 - g - b = .86$  and a stationary BER of  $\rho = .19$ . The bit error rate (when  $P_G$  is increased) corresponding to  $C^\mu$  is  $.2083$ . By using a number of constituent decoders  $N_1 = 2$  and  $N_2 = 2$  we can decode with no errors. The second channel has parameters  $P_G = .0405$ ,  $P_B = .5$ ,  $g = .013057$ , and  $b = .006943$ , which results in  $\mu = .98$  and a stationary bit error of  $\rho = .20$ . The bit error rate corresponding to  $C^\mu$  is  $.2307$ . We have obtained no errors in the decoding by choosing a number of constituent decoders  $N_1 = 2$  and  $N_2 = 3$ . For rate  $1/6$  codes (which verify that the bit error rate corresponding to  $C^{NM}$  is  $\rho = .264$ ) we have considered two Markov channels. The parameters of the first channel are  $P_G = .05$ ,  $P_B = .5$ ,  $g = .0672$ , and  $b = .0728$ , which produces  $\mu = .86$  and a stationary bit error of  $\rho = .284$ . For this channel, the bit error rate corresponding to  $C^\mu$  is  $.301$ . Perfect decoding can be obtained by choosing the number of constituent coders in each block as  $N_i = 1, 1 \leq i \leq 5$ . The second channel has parameters  $P_G = .08333$ ,  $P_B = .5$ ,  $g = .0096$ , and  $b = .0104$ , which produce  $\mu = .98$  and a stationary bit error of  $\rho = .3$ . The bit error rate corresponding to  $C^\mu$  is  $.3255$ . By using a number of constituent decoders  $N_i = 1, 1 \leq i \leq 2$  and  $N_i = 2, 3 \leq i \leq 5$  we can decode with no errors. When the rate of the code decreases, the gap between the stationary bit error probability corresponding to  $C^\mu$  and the stationary bit error probability at which convergence is achieved decreases for  $\mu = .98$  and it becomes close to the gap obtained for channels with  $\mu = .86$  (which is approximately the same for all rates). Also, in both cases ( $\mu = .86$  and  $\mu = .98$ ) the gap between the stationary bit error probability corresponding to  $C^{NM}$  and the stationary bit error probability at which convergence is achieved increases when the code rate decreases, which implies that the improvement obtained by considering the Markov structures is more important for lower rates.

#### IV. JOINT CHANNEL ESTIMATION AND DECODING

Although in the above simulation we have assumed that the parameters of the binary Markov channel are known *a priori* by the decoder, it is also possible to successfully per-

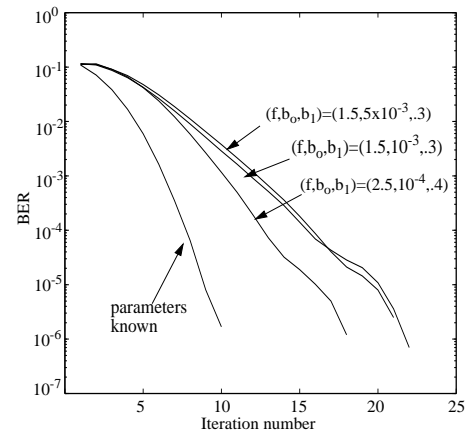


Fig. 5. Convergence behavior for the rate  $1/2$  turbo code and Markov channel defined in Figure 3 (with  $P_G$  chosen such that  $\rho = .12$ ) when *a priori* information about the parameters of the Markov channel is not known. The initial Markov channel is defined by the transition matrix  $b^{in} = .0005$ ,  $g^{in} = .0005 \times f$  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).

form decoding when the parameters describing the Markov channel are initially unknown. The method for doing that is to apply a re-estimation procedure (that is related to the Baum-Welch algorithm [13]) over each one of the supertrellises describing jointly the hidden Markov model and the corresponding constituent decoder. In this way each iteration in turbo decoding also results in an iteration on the parameter estimation and no training sequence is needed. We described a closely related joint estimation/decoding algorithm in some detail for the case of hidden Markov sources in [14]. Figure 5 shows that joint estimation/decoding can also be successfully applied for Markov channels. Figure 5 shows the residual BER as a function of decoding iteration for rate  $1/2$  code and channel associated with Figure 3 ( $\mu = .86$ ,  $\rho = .12$  and code rate equal to  $.5$ ). For the case where the channel parameters are known *a priori*, convergence is achieved in approximately 10 iterations. When the channel parameters are not known *a priori*, convergence to the same residual BER is obtained, though the number of iterations required is larger. The loss in convergence speed is a function of the initial estimate of the parameters. Capacity considerations can be used to provide guidance in this initialization. For example, we used an initial channel estimate specified by the transition matrix  $b^{in} = .0005$ ,  $g^{in} = .0005 \times f$  and the probabilities of error in the good and in the bad state  $P_G^{in}$  and  $P_B^{in}$  respectively. Parameters  $(f, P_G^{in}, P_B^{in})$  should be chosen such that the stationary BER of the initial Markov channel is close to or greater than the one corresponding to the  $C^{NM}$ . When the parameters of the Markov channel are 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 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 blocks except the first, leading to much faster convergence.

## V. CONCLUSIONS

We have introduced parallel concatenated coders for binary Markov channels. We have presented decoder modifications that leverage the *a priori* channel statistics and encoder design approaches that allow the number of constituent encoders to be chosen as a function of the channel statistics and the code rate. In combination, these techniques allow reliable communication at rates above the capacity  $C^{NM}$  of the corresponding interleaved and assumed memoryless channel and near  $C^\mu$ , the true capacity of the channel. In addition, we have discussed joint estimation/decoding that allows the estimation during decoding of Markov model parameters that are initially unknown.

## 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] 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, G. Montorsi, "Unveiling Turbo Codes: Some Results on Parallel Concatenated Coding Schemes," *IEEE Trans. on Information Theory*, pp. 409-428, March 1996.
- [7] 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.
- [8] J. Garcia-Frias and J. Villasenor, "Combining hidden Markov source models and parallel concatenated codes," *IEEE Comm. Letters*, pp. 111-113, July 1997.
- [9] 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*, Brest, France, Sept. 1997.
- [10] 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.
- [11] D. Divsalar, F. Pollara, "Turbo Codes for PCS Applications," *Proc. of the International Conference in Communications*, pp. 54-59, 1995.
- [12] 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.
- [13] 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.
- [14] J. Garcia-Frias and J. Villasenor, "Turbo Decoding of Hidden Markov Sources with Unknown Parameters," to appear in *Proc. of the 1998 IEEE Data Compression Conference*, Snowbird, UT, April 1998.