JOINT SOURCE CHANNEL DECODING OF TURBO CODES

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

We present techniques for modifying a turbo decoder to incorporate a hidden Markov source model, a Markov channel model, or both. This allows the receiver to utilize the statistical characteristics of the source and/or channel during the decoding process, and leads to significantly improved performance relative to systems in which the Markov properties are not exploited. When applied for binary-input, binary-output Markov channels, the methods presented here outperform techniques based on conventional channel interleaving, as they can allow decoding at rates which are above the capacity of a memoryless channel with the same stationary bit error probability as the Markov channel.

I. Introduction

The potential to utilize source and channel memory in communications has been recognized by many authors. For example, Mushkin and Bar-David in [1] derive the capacity of Gilbert-Elliot channels and propose decision feedback decoders to improve the decoding. As Hagenauer has shown in [2], it is also possible to modify the soft-output Viterbi algorithm such that statistics of a source are used in the accumulation of state metrics.

The work presented here involves binary source and channel Markov models in combination with turbo codes, and differs from previous work in several key respects. Previous studies of binary channels with memory are based on Viterbi decoding, and involve techniques that are vulnerable to error propagation. By contrast, the techniques here use only the *a priori* structure of the channel and are effective for poor channels. Another difference lies in the role of channel interleaving. While channel interleavers are needed in traditional convolutional codes to handle fades, with turbo codes one can dispense with channel interleaving and work to directly exploit the structure of the noninterleaved channel in the decoding iterations. Similarly, the structure of a Markov source can also be utilized in decoding.

II. Background and notation

The MAP algorithm originally proposed in [3] laid the groundwork for the forward-backward algorithm that has been used in turbo codes. Following the original publication describing turbo codes, many authors have extended and refined techniques for iterative decoding. We cite in particular the recent paper by Benedetto *et al.* [4], which introduces notation to describe trellis transitions that we utilize here.

We consider the case of a parallel concatenated coder with L constituent convolutional encoders and $L-1$ interleavers [5]. The constituent decoders are enumerated by D_l, $0 \leq l \leq L-1$, 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 $L-1$ "interleaved decoders" D_l, $1 \leq l \leq L-1$ that operate on the observations corresponding to the interleaved inputs. $\pi_l(k)$ describes interleaving applied to input bit u_k at the lth constituent encoder; π_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 kth input bit (before interleaving) is denoted by u_k, $k = 1 \ldots K$, and can take on values i, $i = 0, 1$. 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. Holding the subscript constant and letting the superscript vary refers to all of the observations (in decoders D_0 through D_{L-1}) associated with the kth trellis transition. In discussing the decoding iterations it will often be 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 occuring and the observations $O_k^{\bar{p}} = [O_k^0 \ldots O_k^{p-1} O_k^{p+1} \ldots O_k^{L-1}]$ 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 $[O_1^l \ldots O_K^l]$. $\alpha_k(s)$ represents, for the forward 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. As in [4], we use e to symbolize the trellis edges, or branches, with the starting and ending state associated with a particular edge e given by $s^S(e)$ and $s^E(e)$ respectively. The input bit associated with a branch e is denoted by $u(e)$.

The key results in [3] are the recursions that express $\alpha_k(s)$ in terms of $\alpha_{k-1}(s)$, and $\beta_k(s)$ in terms of $\beta_{k+1}(s)$. In terms of the notation above, the α 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 $s^E(e) = s$:

$$\begin{aligned} \alpha_k(s) = \textstyle\sum_{e:s^E(e)=s} \alpha_{k-1}\left[s^S(e)\right] \times \\ P_k\left[e|O_1^{\bar{f}_p} \ldots O_K^{\bar{f}_p}, s^S(e)\right] P\left[O_k^p|e\right]. \end{aligned} \quad (1)$$

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

$$\begin{aligned} \beta_k(s) = \textstyle\sum_{e:s^S(e)=s} \beta_{k+1}\left[s^E(e)\right] \times \\ P_{k+1}\left[e|O_1^{\bar{f}_p} \ldots O_K^{\bar{f}_p}, s^S(e)\right] P[O_{k+1}^p|e] \end{aligned} \quad (2)$$

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 equations (4), (5), and (6), this allows incorporation of *a priori* source and channel probabilities in a straightforward way.

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.:

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

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

$$\beta_k\left[s^E(e)\right]. \quad (3)$$

In decoders D_l, $l \neq 0$ operating on interleaved inputs, u_k in equation (3) should be replaced by $u_{\pi_l^{-1}(k)}$. In the discussion below, 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 $P(O_1^p \ldots O_K^p|u_k = i) = \frac{P(O_1^p \ldots O_K^p)}{P(u_k=i)} P(u_k = i|O_1^p \ldots O_K^p)$ where $P(u_k = i)$ is the *a priori* probability of the source to generate the bit i. The decoding iterations obtain successively refined estimates of $P(u_k = i|O_1^p \ldots O_K^p)$ via equations (1,2,3).

III. Hidden Markov source models

In the non-interleaved decoder, the properties of a hidden Markov source can be exploited, and we replace the trellis corresponding to the non-interleaved convolutional encoder with an expanded trellis that jointly describes the source and the convolutional coder. The number of states in the joint trellis is the product of the number of states in the hidden Markov source times the number of states in the convolutional coder. Each one of the branches e in the joint trellis will have an associated *a priori* transition probability, a_e. In the interleaved decoders no expansion of the trellis is needed since interleaving masks the Markov properties of the source. Since the Markov properties of the source can only be fully exploited in the non-interleaved decoder, the benefits of the decoder modifications will be greatest when the number of constituent decoders is $L = 2$, and we consider only this case in Section III.

The goal during decoding is to pass the information available from one decoder to the other decoder in such a way that this information, jointly with the *a priori* information from the source available in the new decoder, enables the calculation 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). As described in [6], this can be obtained by using

$$P_k[e|O_1^0 \ldots O_K^0, s^S(e)] =$$

$$G_k P[u_{\pi^{-1}(k)} = u(e)|O_1^0 \ldots O_K^0] \quad (4)$$

for equations (1) and (2) in the interleaved decoder D_1, where G_k is a normalization factor. In the non-interleaved decoder D_0,

$$P_k[e|O_1^1 \ldots O_K^1, s^S(e)] =$$

$$H_k\, P[O_1^1 \ldots O_K^1|u_k = u(e)]a_e \quad (5)$$

is used, where H_k is a normalization factor.

To achieve good performance it is necessary that the non-interleaved and interleaved decoders perform independent estimations of $P(u_k = i|O_1^p \ldots O_K^p)$. 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 $P(u_k =$

$i|O_1^{\bar{f}_p}\ldots O_K^{\bar{f}_p})$ or $P(O_1^{\bar{f}_p}\ldots O_K^{\bar{f}_p}|u_k = i)$ produced by the other decoder.

To produce the input sequence u_k, a hidden Markov source with N states S_j, $0 \leq j \leq N-1$ is used. The probability of producing output bit i, $i \in \{0,1\}$, when the source is in state S_j is $P_o(i|S_j)$. The probability of transition from state S_l to S_k is $P_t(S_k|S_l)$. As an example, consider the case of $N = 2$ states, with transition probabilities $P_t(S_0|S_0) = .1$, $P_t(S_1|S_1) = .15$ and output probabilities $P_o(0|S_0) = .95$, $P_o(1|S_1) = .925$. Sequences produced by this model satisfy $P(u_k = 0) = P(u_k = 1) = .5$, and have an entropy rate of 0.729 bits/symbol. This hidden Markov source was used in simulation as input to a rate 1/3 encoder that included a systematic bit and two identical recursive 8-state, rate 1 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. Using this source, 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.

IV. Finite state Markov channels

In many communications systems of practical interest the channel error patterns seen at the output of the demodulator can be characterized using hidden Markov models. Although the approach we present below can be applied to general binary-input, binary-output Markov channels, for simplicity we consider the special case of Gilbert-Elliot channels. A Gilbert-Elliot channel contains a "good" state in which the probability of bit error is P_G and a "bad" state with error probability P_B. The probability of going from the good state to the bad state is b and the probability of going from the bad state to the good state is g [1]. 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. In constituent decoder D_l with rate $1/n_l$ each coder "superstate" contains N^{n_l} channel states as illustrated in Figure 1. Equations (1,2,3) can still be applied, although in contrast to case for the Markov source in Section III, the terms $P[O_k^p|e]$ and $P\left[O_{k+1}^p|e\right]$ also depend on $s^E(e)$. 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 $P(u_k = 0) = P(u_k = 1) = .5$ we obtain

$$P_k[e|O_1^{\bar{f}_p}\ldots O_K^{\bar{f}_p}, s^S(e)] = J_k a_e \prod_{i=0,i\neq p}^{L-1} P[u_{\pi_p^{-1}(k)} = u(e)|O_1^i \ldots O_K^i] \quad (6)$$

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. In contrast with the case of a source Markov model, the value of a_e depends only on the transition probabilities of the hidden Markov model representing the channel.

To demonstrate the performance of the modified decoder, consider the rate 1/3 encoder with i.i.d. input shown in Figure 2. Each constituent coder produces parity bits using a rate 1 recursive 8-state convolutional encoder with the same generator matrix as used for the simulation described in Section III. In contrast with customary turbo codes in which the parity and systematic bits are transmitted for the first consituent coder (E_0), the coder E_0 in Figure 2 is rate 1, obtained using a mix of systematic and parity bits (one systematic bit is sent after every 19 parity bits). In the simulations, the interleavers were of size 16384. The Gilbert-Elliot model parameters were $P_G = .01$, $P_B = .5$, $g = .0914$, and $b = .0486$. As described in [1], the "capacities" that can be associated with a Markov channel include the true capacity C^μ; and C^{NM}, ("no memory"), the capacity of a memoryless channel with the same stationary BER as the Markov channel. Clearly, $C^\mu \geq C^{NM}$. For the Gilbert-Elliot channel described above, the stationary bit error probability is .18. Note that this is higher than the bit error rate of $\rho = .174$ which corresponds to C^{NM} for a rate 1/3 code (i.e., $1 - H(.174) = 1/3$ where $H()$ is the binary entropy function). Thus, reliable communication over this channel using a rate 1/3 code is not possible in a traditional system that uses channel interleaving and does not exploit channel memory. However, as Figure 3 shows, when simulations are performed using approximately 20 million input bits, the rate

1/3 turbo code with the modified decoder trellises attains a residual BER of about 2×10^{-6} after 14 iterations. The decoder converges fully (though this may not be statistically significant) after 15 iterations. Also shown on Figure 3 are curves corresponding to the rate 1/3 encoder (with i.i.d. input) from Section III in which the decoder trellis is not modified to incorporate the channel model. As expected, convergence fails to occur regardless of whether a channel interleaver is used. In general we have found that as the number of constituent coders is increased, succesful decoding is possible for stationary BERs farther above C^{NM} and closer to C^{μ}. Though in the example above encoder E_0 was rate 1 and produced predominantly parity bits, for some channels and code rates it is best to use a rate 1/2 encoder (producing one systematic bit and one parity bit for each input bit) for E_0. If the systematic and then parity bits are transmitted as contiguous blocks, the decoder trellis must be modified to consider the combined channel state associated with both transmitted bits.

V. Conclusions

We have introduced a decoding method for parallel concatenated codes that considers source and channel memory. These methods provide a means to directly exploit the higher capacity of Markov channels relative to binary symmetric channels with the same stationary bit error probability. For Markov sources, these techniques allow convergence at higher noise levels than is possible when source statistics are not utilized.

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

[3] 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.

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

[5] D. Divsalar, F. Pollara, "Turbo Codes for PCS Applications," *Proc. of the International Conference in Communications*, pp. 54-59, 1995.

[6] J. Garcia-Frias and J. Villasenor, "Combining hidden Markov source models and parallel concatenated codes," *IEEE Comm. Letters*, July 1997.

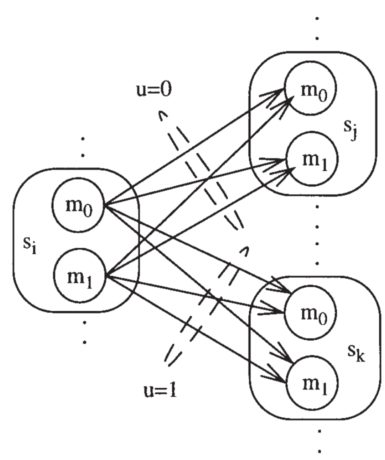

Fig. 1. Joint decoder trellis describing the encoder (states s) and the Markov channel (states m). u is the information bit.

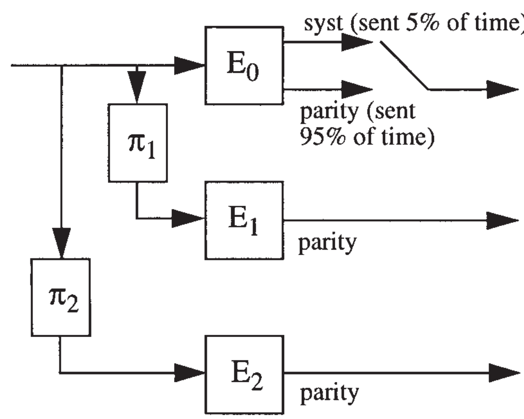

Fig. 2. Encoder structure used for the rate 1/3 example described in Section IV. All constituent encoders are rate 1. For E_0, one systematic bit is sent after each set of 19 parity bits.

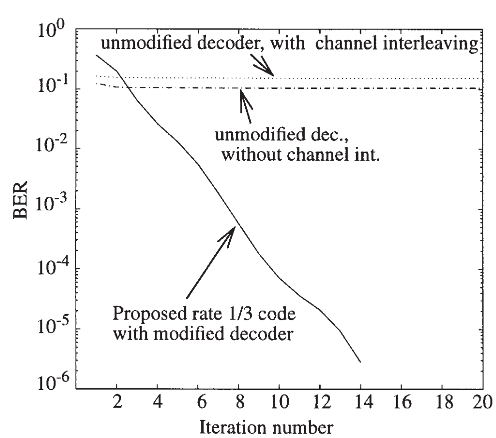

Fig. 3. Convergence behavior for the rate 1/3 code described in Section IV. The channel is a Gilbert-Elliot channel with stationary BER .18 (a BSC with this BER has capacity < 1/3). The performance of the rate 1/3 encoder from Section III, without modified decoder, is shown for comparison. The input is i.i.d.