Joint Source-Channel Decoding of Variable-Length Codes for Convolutional Codes and Turbo Codes

Marion Jeanne, Jean-Claude Carlach, *Member, IEEE*, and Pierre Siohan, *Senior Member, IEEE*

***Abstract*—Several recent publications have shown that joint source-channel decoding could be a powerful technique to take advantage of residual source redundancy for fixed- and variable-length source codes. This letter gives an in-depth analysis of a low-complexity method recently proposed by Guivarch *et al.*, where the redundancy left by a Huffman encoder is used at a bit level in the channel decoder to improve its performance. Several simulation results are presented, showing for two first-order Markov sources of different sizes that using *a priori* knowledge of the source statistics yields a significant improvement, either with a Viterbi channel decoder or with a turbo decoder.**

***Index Terms*—Convolutional codes (CCs), joint source-channel coding, turbo codes (TCs), variable-length codes (VLCs).**

I. Introduction

SOURCE and channel coding are essential functions in any communication system. The source-coding stage is designed to remove as much redundancy as possible from the source, while the channel-coding stage is bound to add controlled redundancy to the compressed source. For practically all systems, these blocks are optimized separately, which can be theoretically justified by Shannon's theorems related to source and channel coding [1]. However, for practical reasons related to complexity and delay, we are, most of the time, far from Shannon's theoretical scheme. Therefore, it can be more efficient to jointly optimize source and channel coders/decoders, rather than trying to improve source and channel coder design separately. In this letter, we focus on the case of joint source-channel decoding (JSCD) for variable-length codes (VLCs).

In order to limit the propagation-error phenomenon, the first attempts in JSCD considered the case where the source is encoded using fixed-length codes. References [2] and [3] are two illustrative examples of the innovative techniques proposed in this area. However, because of the high compression capacity of VLCs, attention is now shifted to these techniques that lead to a more difficult problem, and imply a packetization in order to limit error propagation. For the VLC-based JSCD methods, the distinction can be made between the ones that use the source redundancy at the source-decoding stage (Subbalakshmi and Vaisey [4], Bystrom *et al.* [5], or Perros-Meilhac and Lamy [6]), the ones that use it at the channel decoder, such as Murad and Fuja [7], Demir and Sayood [8], Park and Miller [9], or Lakovič and Villasenor [10], and finally, those that improve the overall decoding using the turbo principle between source and channel decoders (Bauer and Hagenauer [11], Guyader *et al.* [12]).

In this letter, we present an approach that belongs to the second category. A specificity of our method is that the *a priori* source information is introduced at a bit level, and allows a nearly optimal maximum *a posteriori* (MAP) decoding without introducing a noticeable modification of the usual channel decoders. Thus, we provide a new alternative to solve the complexity problem related to the design of optimal MAP JSCD. The JSCD algorithm proposed by Murad and Fuja [7] gives a precise idea of what the complexity is of getting an exact MAP estimation. Indeed, in the case of a Markov source which is encoded using a VLC and a convolutional code (CC), their JSCD technique corresponds to a generalized Viterbi algorithm which is applied on a "super-trellis." The resulting graph is the product of three trellises: the binary convolutional trellis, the VLC trellis, and the Markov trellis. Even if the states that are never reached are eliminated, this method still remains very complex. It may, therefore, be more advisable to look for suboptimal algorithms providing acceptable tradeoffs between complexity and performance. The high computational complexity involved with an optimum MAP sequence decoder is also noticed in [8] and [9], where a trellis state reduction is proposed, resulting in an approximate MAP sequence estimation. Our experimental setup is a conventional one, including a memoryless or a Markov source that is Huffman-encoded and protected by a channel code; either a CC or a turbo code (TC). We provide an in-depth analysis of the JSCD that we initially proposed and described in [13] and [14]. Our simulation results for these different setups correspond to a transmission over an additive white Gaussian noise (AWGN) channel.

Our paper is organized as follows. In Section II, we present the computation technique to obtain the *a priori* bit probabilities from the *a priori* symbol probabilities of the source. Our method to use the *a priori* bit probabilities at the channel decoder is then presented in Section III for CC, and in Section IV for TC. Simulation results are presented using two different Markov sources.

II. *A Priori* Bit Probabilities of VLCs

In the decoding of entropy-coded sources, it may be assumed that the symbol probabilities are known either directly from the source or from an estimation algorithm. This knowledge can naturally lead to improved performance in source decoding

Paper approved by C. Schlegel, the Editor for Coding Theory and Techniques of the IEEE Communications Society. Manuscript received July 8, 2003; revised May 14, 2004. This work was supported in part by the French Ministry of Research under Contract COSOCATI. This paper was presented in part at the IEEE Data Compression Conference, Snowbird, UT, March 2000, and in part at the IEEE International Conference on Communication, New York, NY, May 2002.

M. Jeanne was with the R&D Division of France Telecom, RESA/BWA, 35512 Cesson Sévigné Cedex, Cesson Sévigné, France. She is now with Project Temics, INRIA-Rennes, 35042 Rennes Cedex, France (e-mail: marion.jeanne@irisa.fr).

J.-C. Carlach is with the R&D Division of France Telecom, RESA/BWA, 35512 Cesson Sévigné Cedex, France (e-mail: jeanclaude.carlach@francetelecom.com).

P. Siohan was on sabbatical leave with INRIA-Rennes, 5042 Rennes Cedex, Rennes, France. He is now with the R&D Division of France Telecom, RESA/BWA, 35512 Cesson Sévigné Cedex, France (e-mail: Pierre.Siohan@francetelecom.com).

Digital Object Identifier 10.1109/TCOMM.2004.840664

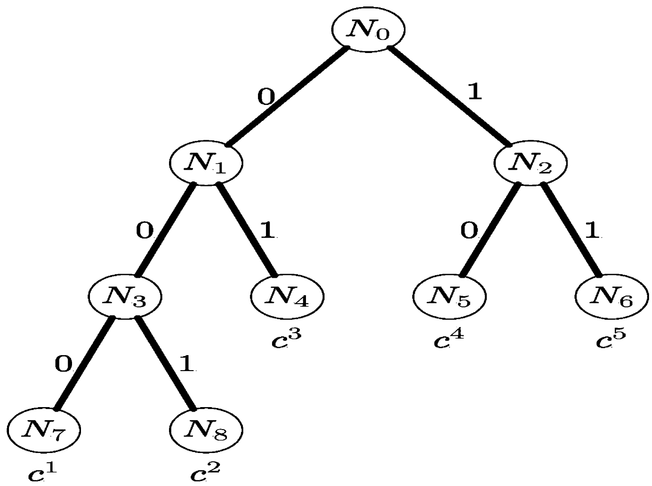

Fig. 1. VLC tree representation.

[4]–[6], but also in channel decoding [7]–[10] if a channel code is added to protect the data. Commonly used channel codes deal with a bit-based trellis at the decoding stage. It may then be convenient to derive the source probabilities at a bit level. Indeed, the source statistics can then directly be used in the metric of the channel-decoding algorithm. In this letter, we assume that the VLC can be represented by a finite-size tree: it is a Huffman code or a variant of a Huffman code. To obtain bit-source probabilities, one can compute the probabilities to have a zero and a one at the output of the VLC. Using these probabilities in a static mode may, of course, improve the channel-decoding performance. However, by keeping a synchronization between the VLC tree and the channel-decoder trellis, as much as possible, more accurate bit-source probabilities can be computed corresponding to any given location in the VLC tree.

We denote by $\mathcal{C} = \{c^1, c^2, \ldots, c^C\}$ the set of source symbols, and each symbol $c^j \in \mathcal{C}$ is written as $c^j = [c^j(1), \ldots, c^j(i), \ldots]$ with i being the bit index and $c^j(i)$ a bit equal to zero or one. Fig. 1 presents an example of a VLC tree borrowed from [15]. It is such that $C = \mathrm{card}(\mathcal{C}) = 5, c^1 = [0,0,0], c^2 = [0,0,1], c^3 = [0,1], c^4 = [1,0], c^5 = [1,1]$, and N_i denotes the node number i. Let $I_k^j = \{n \in \mathbf{N} | \forall i < k, c^n(i) = c^j(i)\}$ be the set of indexes of all codewords, with the first $k-1$ bits being equal to the first $k-1$ bits of the codeword c^j. By convention, we set $I_1^j = \{1, \ldots, C\}$. We denote by $P(c^j)$ and $P(c^j(i))$ the probabilities of symbol c^j and bit $c^j(i)$, respectively. The probability of each edge of the tree can be expressed, for $k > 1$, with respect to the symbol probabilities using the following relation, also illustrated in [13] with a small code:

$$P(c^j(k)|c^j(1), \ldots, c^j(k-1)) = \frac{\sum_{n \in I_k^j | c^n(k) = c^j(k)} P(c^n)}{\sum_{n \in I_k^j} P(c^n)}. \tag{1}$$

For $k = 1$, i.e., for the two branches connected to the root, the left-hand side in (1) is reduced to $P(c^j(1))$. This latter equation is relevant for a source of independent symbols. If we have a first-order Markov relation between two successive symbols instead, the calculations must be done considering each symbol as a possible previous symbol, with for $k > 1$

$$\forall c^m \in \mathcal{C}, P(c^j(k)|c^j(1), \ldots, c^j(k-1), c^m) = \frac{\sum_{n \in I_k^j | c^n(k) = c^j(k)} P(c^n|c^m)}{\sum_{n \in I_k^j} P(c^n|c^m)}. \tag{2}$$

The probabilities given in (1) and (2) are the *a priori* bit probabilities, they will be used afterwards in a soft decoding scheme at the channel-decoder level.

Huffman codes, as, for instance, the one depicted in Fig. 1, are built using the probabilities $P(c^n)$. However, the resulting code does not, in general, reach the source entropy that depends on $P(c^n)$ for a memoryless source, and also on $P(c^n|c^m)$ for a first-order Markov source. A residual redundancy is then left that can be exploited through a JSCD technique.

For a stationary source, the residual redundancy R is given by the expression $R = \bar{L} - H_i$, with H_i being the entropy of order i ($i = 0$ for a memoryless source and one for a first-order Markov source) and $\bar{L}$ the average codeword length. Note also that, when using VLCs, a single bit error can produce a loss of synchronization that, in the worst case, will last until the end of the corresponding packet of symbols. To measure the performance of a given transmission scheme, bit-error rate (BER) is no longer the correct measure, and a symbol-error rate (SER) has to be computed. In our case, the SER is simply the ratio between the number of different symbols and the number of transmitted symbols. It is an appropriate measure for source sequential decoding. If one wants a more precise idea of the synchronization capabilities of a VLC, an alternative measure is given by the Levenshtein metric [16].

III. JSCD of Entropy-Coded Sources With CCs

A. *Our Low-Complexity JSCD Technique*

Our method is presented in the case of memoryless and first-order Markov sources. In both cases, channel coding is performed by a CC of rate $1/2$. To limit error propagation at some point, we have used some resynchronization. As shown in Fig. 2, which presents the transmission scheme, data is packetized into blocks of P symbols. We also use the following notations. d_l is an entropy-encoded bit at bit-time l. There is a variable number of bits per packet, denoted L. Let us denote the two channel-coded bits by u_l and v_l, and their noisy version by x_l and y_l, respectively. The input of the decoding process is denoted by r_l, representing the pair (x_l, y_l). The sequence of L decoder inputs is denoted by R_1^L. Finally, S_l is a state of the channel trellis at time l. There are M states at each trellis stage. S_0^L is a sequence of $L+1$ states. $\mathcal{S}$ represents the set of all possible sequences of states.

Our JSCD method is based on a variant of the Viterbi algorithm that makes use of the *a priori* source probabilities. Indeed, Viterbi's algorithm [17] is the most common way to decode CCs. It is optimum with respect to the sequence's maximum-likelihood criterion, i.e., it maximizes the probability $P(R_1^L|S_0^L)$. If the source is equally distributed, it is known that the Viterbi algorithm is also optimum with respect to the MAP sequence criterion, i.e., it maximizes the probability

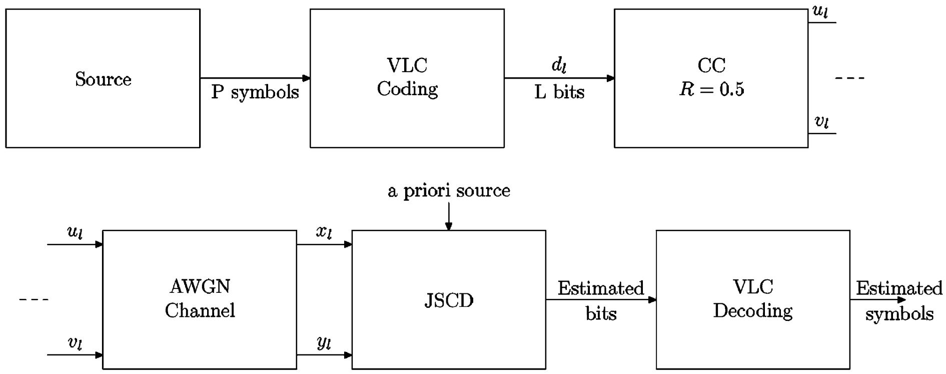

Fig. 2. Transmission process for a VLC-encoded source with a CC.

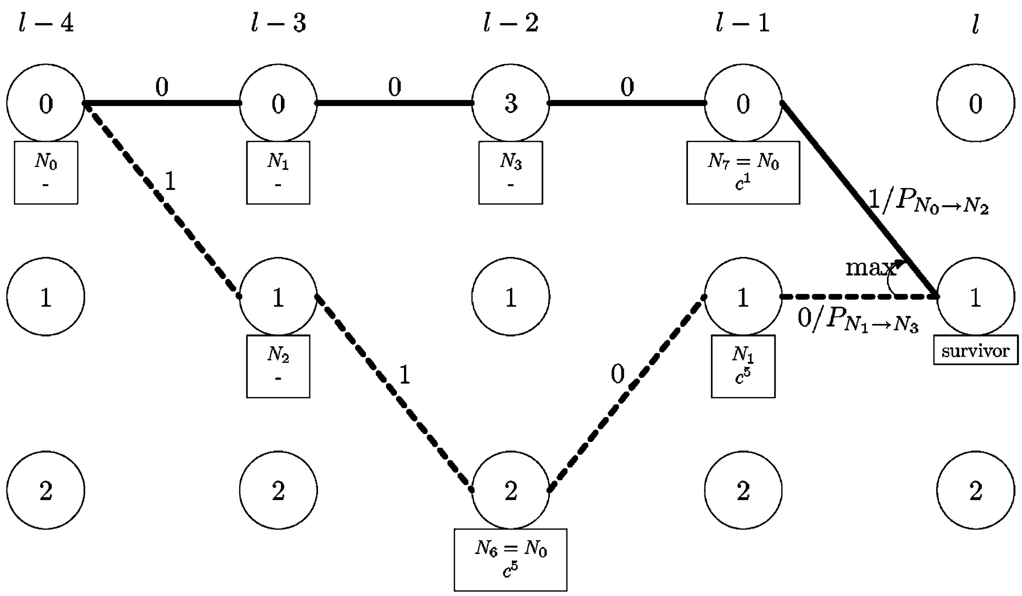

Fig. 3. Illustration, for a memoryless source, of the selection mechanism between concurrent paths at state number 1 in stage l.

$P(S_0^L|R_1^L)$. In our letter, the criterion used is the MAP sequence, but the source is assumed to be unequally distributed. Therefore, to approach the optimum, we use the generalized Viterbi algorithm [18]. Using Bayes' rule, we have

$$P\left(S_0^L\middle|R_1^L\right) = \frac{P\left(R_1^L\middle|S_0^L\right)P\left(S_0^L\right)}{P\left(R_1^L\right)}. \tag{3}$$

$P(R_1^L)$ being constant with respect to $\mathcal{S}$, the sequence MAP of states is such that

$$\begin{aligned}
&\max_{\mathcal{S}} P\left(S_0^L\middle|R_1^L\right)\\
&\Leftrightarrow \max_{\mathcal{S}} P\left(R_1^L\middle|S_0^L\right)P\left(S_0^L\right)\\
&\Leftrightarrow \max_{\mathcal{S}} \left(\ln\prod_{l=1}^{L} P(r_l|S_l,S_{l-1}) + \ln\prod_{l=1}^{L} P(S_l|S_{l-1})\right)\\
&\Leftrightarrow \max_{\mathcal{S}} \sum_{l=1}^{L} \left(\ln(P(r_l|S_l,S_{l-1})) + \ln(P(S_l|S_{l-1}))\right). \quad (4)
\end{aligned}$$

The noise samples being independent, we have the conditional independence of x_l and y_l, given u_l and v_l, respectively. Hence, $P(r_l|S_l,S_{l-1}) = P(x_l|u_l)P(y_l|v_l)$. The two last probabilities depend upon the channel and upon the modulation used. In the case of an AWGN channel using binary phase-shift keying (BPSK) modulation and a CC of rate $1/2$, we get

$$P(x_l|u_l) = \frac{1}{\sqrt{\pi N_0}} e^{-\frac{(x_l-(2u_l-1)\sqrt{E_b/2})^2}{N_0}} \tag{5}$$

and a similar equation for $P(y_l|v_l)$, with N_0 being the single-sided noise density, and E_b the energy per transmitted information bit. The specificity of our problem is that the *a priori* source probability $P(S_l|S_{l-1})$ in (4) is not equally distributed.

To get the optimum decoded output with the generalized Viterbi algorithm, only one path for each state of each stage is stored: the survivor. In this context, it is not too complex to keep track of the channel trellis survivors in the VLC tree. $P(S_l|S_{l-1})$ is then equal to the right VLC edge probability, and has to be explicitly determined using (1) or (2), depending on the source model. This mechanism is shown, for a memoryless source, in Fig. 3. $P_{N_i->N_j}$ is the transition probability between node N_i and N_j in the VLC tree. This scheme illustrates, for $S_l = 1$ in a channel trellis code, an example of the selection for the VLC given in Section II, between the sequence of bits $\{0,0,0,1\}$ corresponding to the sequence of symbols $\{c^1,\ldots\}$, and the sequence of bits $\{1,1,0,0\}$ corresponding to the sequence of symbols $\{c^5,\ldots\}$.

If we use the same source statistics as in [15, p. 93], i.e., $P(c^1) = 0.15, P(c^2) = 0.15, P(c^3) = 0.25, P(c^4) = 0.25$, and $P(c^5) = 0.2$, then the *a priori* source information for the first sequence, i.e., the probability of being in node N_0 and of receiving a bit equal to one, is such that

$$\begin{aligned} P(S_l = 1|S_{l-1} = 0) &= P_{N_0->N_2} = P(c^4(1)) \\ &= \frac{P(c^4) + P(c^5)}{P(c^1) + P(c^2) + P(c^3) + P(c^4) + P(c^5)} \\ &= 0.45 \end{aligned} \tag{6}$$

and the *a priori* source information for the second sequence, i.e., the probability of being at node N_1 and of receiving a bit equal to zero, is given by

$$\begin{aligned} P(S_l = 1|S_{l-1} = 1) &= P_{N_1->N_3} \\ &= P(c^1(2)|c^1(1)) \\ &= \frac{P(c^1) + P(c^2)}{P(c^1) + P(c^2) + P(c^3)} \approx 0.545. \end{aligned} \tag{7}$$

Due to unavoidable decoding errors, the one-to-one correspondence with the VLC tree may be badly used, making our algorithm slightly suboptimal. However, the insertion of $P(S_l|S_{l-1})$ in (4) will, in general, significantly improve the decoding step.

If we now consider a Markov model for the source, the *a priori* probability of the source also depends upon the previous transmitted symbol, so its knowledge is necessary at each state of each stage of the decoding process. As for the memoryless source, we also have to keep track of the survivor paths in the VLC tree. But now, in (4), the *a priori* source information also depends on the previous symbol, i.e., $P(S_l|S_{l-1})$ is computed using (2) instead of (1). Therefore, the extra complexity to be added is limited to the storage of the previous symbol.

B. Simulation Results

In our simulations, as in [7], $P = 256$ symbols. As we use a decoding trellis, different from [8] and [9], which use P and L, our decoder, initialized at the beginning of each packet, only requires the knowledge of L. The transmission ends when either the maximum bit-error number (fixed to 1000) or the maximum transmitted bits (equal to 10^7) is reached. The generator polynomials of the CC are $G_1 = 1 + D + D^4$ and $G_2 = 1+D^2+D^3+D^4$. Without any *a priori* knowledge of the source (tandem scheme), an equal distribution is assumed, and the CC decoder operates at $P(S_l|S_{l-1}) = 1/2$. Otherwise, $P(S_l|S_{l-1})$ is computed as explained in Section III-A (JSCD scheme). To allow a comparison with the optimum MAP, we have used the three-symbol VLC source given in [7] and [14]. This Markov source, denoted M-F, is such that $R = 0.67$ b/symbol. We also carried out simulations with a first-order Gauss–Markov source with unit variance and a correlation factor equal to 0.9. After a uniform quantization over 16 levels and Huffman encoding, we get our second Markov source, denoted G-M. It is such that $R = 1.13$ b/symbol. The results obtained for the transmission of these sources over an AWGN channel and using BPSK modulation are reported in Fig. 4.

For the M-F source, it can be seen that, at least in this particular example, our JSCD method is nearly optimal. Indeed, the resulting curve is practically superimposed onto the one resulting from the "super-trellis" algorithm. Due to the high complexity of the "super-trellis," comparisons with a bigger Huffman code are not feasible. On the contrary, our JSCD remains applicable for larger VLCs and provides substantial improvement, compared with the tandem scheme. When SER $= 10^{-2}$ the gain in $(E_b)/(N_0)$ is around 1.3 dB for the M-F source, and around 0.7 dB for the G-M source. As the gain is decreasing with R, this suggests that R is probably not the appropriate parameter to estimate what JSCD gain can be expected with our method. If we use the "relative residual redundancy" defined by $R_r = 1-\bar{L}/H_1$ instead, we get $R_r = 0.54$ for the three-symbol M-F source, and $R_r = 0.33$ for the 16-symbol G-M one. These last figures are more in accordance with our results.

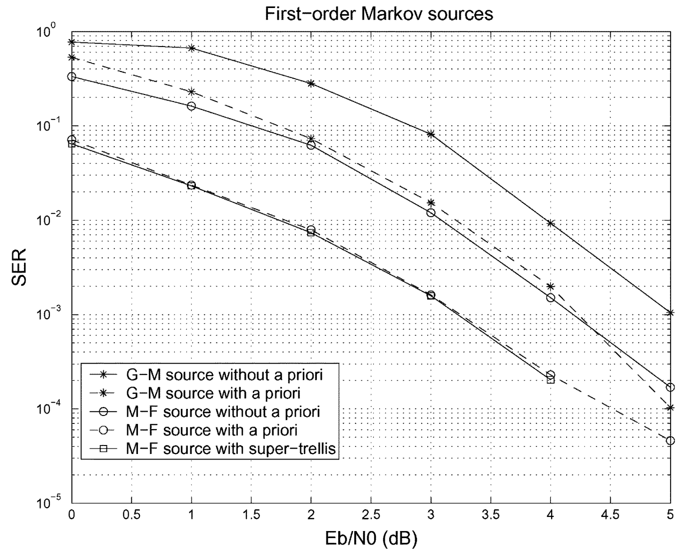

Fig. 4. SER with and without use of the *a priori* source information for the two Markov sources with a CC.

IV. JSCD of Entropy-Coded Sources With TCs

A. Our Source-Controlled Turbo Decoding

The transmission scheme is the same as in the previous section, but we replace the convolutional encoder/decoder by a turbo encoder/decoder. The TC used is a parallel concatenation [19] of two recursive systematic convolutional (RSC) coders separated by a line-column interleaver, followed by a $1/2$ puncturing. Keeping the notations given in Section III, we can note that for these systematic encoders, $u_l = d_l$, and that v_l comes from one of the two coders, depending on the puncturing. The turbo decoding requires the computation of extrinsic information, denoted $\mathrm{Ext}(d_l)$, for each transmitted bit. Turbo decoding is an iterative process, the extrinsic information is processed through the two constituent decoders and through a number of iterations. A schematic description of this decoder can be found in [13] and [14]. Several algorithms can be used to compute the extrinsic information. In this letter, the Max-Log-MAP [20] is chosen, because it offers a good tradeoff between complexity and efficiency.

Let us first examine how to use the *a priori* source information in a turbo decoder based on the bit-by-bit MAP algorithm.

The SUBMAP will be presented afterwards as an approximation of this optimal algorithm. Our goal is thus to compute the *a posteriori* probability (APP) of the transmitted bits

$$\mathrm{APP}(d_l) = P\left(d_l \left| R_1^L \right.\right) = \sum_{S_l=1}^{M} P\left(d_l, S_l \left| R_1^L \right.\right). \quad (8)$$

The forward and backward probabilities at state S_l are $\alpha(S_l) = P(S_l, R_1^l)$ and $\beta(S_l) = P(R_{l+1}^L | S_l)$, respectively. Then, the joint probability $P(d_l, S_l, R_1^L)$, that is proportional to $P(d_l, S_l | R_1^L)$, is given by

$$P\left(d_l, S_l, R_1^L\right) = \sum_{S_{l-1}=1}^{M} \beta(S_l) P(d_l, S_l, r_l | S_{l-1}) \alpha(S_{l-1}) \quad (9)$$

where each of the three terms are

$$\begin{aligned} \beta(S_l) &= \sum_{S_{l+1}=1}^{M} \sum_{d_{l+1}=0}^{1} \beta(S_{l+1}) \\ &\quad \times P(d_{l+1}, S_{l+1}, r_{l+1} | S_l) \quad (10) \\ \alpha(S_l) &= \sum_{S_{l-1}=1}^{M} \sum_{d_l=0}^{1} \alpha(S_{l-1}) \\ &\quad \times P(d_l, S_l, r_l | S_{l-1}) \quad (11) \\ P(d_l, S_l, r_l | S_{l-1}) &= P(r_l | d_l, S_l, S_{l-1}) \\ &\quad \times P(d_l | S_l, S_{l-1}) P(S_l | S_{l-1}). \quad (12) \end{aligned}$$

At this point, we use a result from Berrou and Glavieux [19], with $P(d_l | S_l, S_{l-1})$ equal to 0 or 1, depending on whether the branch exists or not, with $P(r_l | d_l, S_l, S_{l-1})$ being the error probability of the memoryless (AWGN) channel and $P(S_l | S_{l-1})$ the *a priori* source information. As the encoders are systematic, and due to the noise samples' independence, we have

$$P(r_l | d_l, S_l, S_{l-1}) = P(x_l | d_l) P(y_l | v_l). \quad (13)$$

But as $P(x_l | d_l)$ does not depend on the trellis state, the APPs are proportional to the following quantities:

$$\begin{aligned} \mathrm{APP}(d_l) &\sim P(x_l | d_l) \sum_{S_l=1}^{M} \sum_{S_{l-1}=1}^{M} \beta(S_l) P(y_l | v_l) \\ &\quad \times P(d_l | S_l, S_{l-1}) P(S_l | S_{l-1}) \alpha(S_{l-1}) \\ &\sim P(x_l | d_l) \mathrm{Ext}(d_l). \quad (14) \end{aligned}$$

Equation (14) is the basis of turbo decoding.

As in the previous section, the originality of our approach is to propose a channel-decoding algorithm that uses accurate values of $P(S_l | S_{l-1})$. In this case, APPs are computed with (14) using *a priori* source information, computed with (1) or (2), depending on the source model. This can be carried out, as explained in Section III, by keeping a one-to-one relation between the channel trellis and the VLC tree.

However, in turbo decoding, the addition of the *a priori* source information is only possible for the first decoder, because the interleaver breaks the one-to-one correspondence between the VLC tree and the decoding trellis in the second decoder. Thus, maintaining a correspondence between the probability edge computed with (1) or (2) and $P(S_l | S_{l-1})$ in the second decoder would involve great complexity. So, in this last decoder, the probability $P(S_l | S_{l-1})$ is set to $1/2$, which is equivalent to considering the source as binary symmetric.

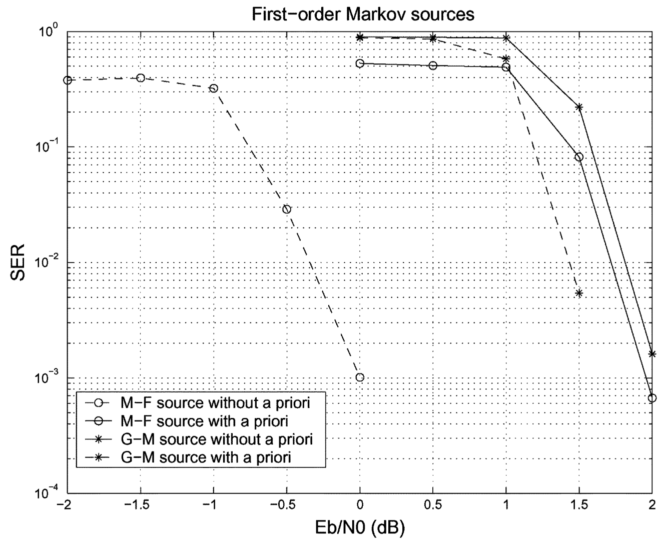

Fig. 5. SER with and without use of the *a priori* source information for the two Markov sources with a TC (third iteration).

In order to limit the computational complexity, our JSCD is, in fact, based on a Max-Log-MAP algorithm [20]. Let $\bar{\alpha}(S_l)$, $\bar{\beta}(S_l)$, and $\bar{P}(d_l, S_l, r_l | S_{l-1})$ denote the approximation of $\ln \alpha(S_l)$, $\ln \beta(S_l)$, and $\ln P(d_l, S_l, r_l | S_{l-1})$, respectively. Then, using logarithms, we finally obtain the equations that we actually used in our simulations

$$\begin{aligned} \bar{\alpha}(S_l) &= \max_{S_{l-1}, d_l} \left(\bar{P}(d_l, S_l, r_l | S_{l-1}) + \bar{\alpha}(S_{l-1})\right) \\ \bar{\beta}(S_l) &= \max_{S_{l+1}, d_{l+1}} \left(\bar{P}(d_{l+1}, S_{l+1}, r_{l+1} | S_l) + \bar{\beta}(S_{l+1})\right) \\ \ln P\left(d_l, S_l, R_1^L\right) &= \max_{S_{l-1}} (\bar{\beta}(S_l) + \bar{P}(d_l, S_l, r_l | S_{l-1}) \\ &\quad + \bar{\alpha}(S_{l-1})). \end{aligned}$$

B. Simulation Results

The TC used is a parallel concatenation of two encoders. Its RSC constituent encoders have an identical generator equal to $(1+D+D^2+D^4)/(1+D^3+D^4)$. The code rate is $1/2$, and the interleaver size is 64×64, which is also the packet size, equal to 4096 and now given in bits. As we work with VLCs, this means a packet does not obviously correspond to an integer number of symbols, so some extra bits are added. We again initialized the decoding trellises at the beginning of each packet. In our simulations, we compute the SER as a function of the signal-to-noise ratio for the first three iterations of turbo decoding. Results are given in Fig. 5 for M-F and G-M sources.

The conclusions are similar to those given for the CC. With the *a priori* source information, the decoding, here turbo decoding, is improved. For instance, at SER$=10^{-2}$ using the Markov model, a gain of 2.1 dB is obtained for the three-symbol source of Murad and Fuja $(R_r = 0.54)$, while the 16-symbol Gauss–Markov source $(R_r = 0.33)$ leads to a gain of 0.4 dB.

V. Conclusion

In this letter, we described a simple and new method to use the residual redundancy left by Huffman encoders at a bit level. For a first-order Markov source transmitted over an AWGN channel, we showed that, compared with a tandem scheme, our JSCD method may lead to 1.3 dB improvement with a CC and 2.1 dB for a TC. As reported in [14], by using a different measure based on the Levenshtein metric, similar gains can be obtained.

Acknowledgment

The authors would like to thank Dr. A. Murad and Prof. E. Fuja for providing further information on [7]. The authors would also like to thank the anonymous reviewers for their helpful comments.

References

[1] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, 1948.

[2] J. Hagenauer, "Source-controlled channel decoding," *IEEE Trans. Commun.*, vol. 43, pp. 2449–2457, Sep. 1995.

[3] N. Görtz, "Analysis and performance of iterative source-channel decoding," in *Proc. Int. Symp. Turbo Codes, Related Topics*, Brest, France, Sep. 2000, pp. 251–254.

[4] K. P. Subbalakshmi and J. Vaisey, "On the joint source-channel decoding of variable length encoded sources: The BSC case," *IEEE Trans. Commun.*, vol. 49, pp. 2052–2055, Dec. 2001.

[5] M. Bystrom, S. Kaiser, and A. Kopansky, "Soft source decoding with applications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, pp. 1108–1120, Oct. 2001.

[6] L. Perros-Meilhac and C. Lamy, "Huffman tree based metric derivation for a low-complexity sequential soft VLC decoding," in *Proc. Int. Conf. Commun.*, New York, NY, Apr. 2002, pp. 783–787.

[7] A. H. Murad and T. E. Fuja, "Joint source-channel decoding of variable length encoded sources," in *Proc. ITW*, Killarney, Ireland, Jun. 1998, pp. 94–95.

[8] N. Demir and K. Sayood, "Joint source/channel coding for variable length codes," in *Proc. IEEE Data Compression Conf.*, Snowbird, UT, Mar. 1998, pp. 139–148.

[9] M. Park and D. J. Miller, "Joint source-channel decoding for variable-length encoded data by exact and approximate MAP sequence estimation," *IEEE Trans. Commun.*, vol. 48, pp. 2449–2457, Jan. 2000.

[10] K. Laković and J. Villasenor, "Combining variable length codes and Turbo codes," in *Proc. IEEE Veh. Technol. Conf.*, vol. 4, May 2002, pp. 1719–1723.

[11] R. Bauer and J. Hagenauer, "Iterative source/channel decoding using reversible variable length codes," in *Proc. IEEE Data Compression Conf.*, Snowbird, UT, Mar. 2000, pp. 93–102.

[12] A. Guyader, E. Fabre, C. Guillemot, and M. Robert, "Joint source-channel turbo decoding of entropy-coded sources," *IEEE J. Sel. Areas Commun.*, vol. 19, pp. 1680–1696, Sep. 2001.

[13] L. Guivarch, J. C. Carlach, and P. Siohan, "Joint source-channel soft decoding of Huffman sources with turbo-codes," in *Proc. IEEE Data Compression Conf.*, Snowbird, Utah, Mar. 2000, pp. 83–92.

[14] M. Jeanne, J. C. Carlach, P. Siohan, and L. Guivarch, "Source and joint source-channel decoding of variable length codes," in *Proc. Int. Conf. Commun.*, vol. 2, New York, NY, Apr. 2002, pp. 768–772.

[15] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York: Wiley, 1990.

[16] T. Okuda, E. Tanaka, and T. Kasai, "A method for the correction of garbled words based on the Levenshtein metric," *IEEE Trans. Comput.*, vol. C-25, pp. 1064–1070, Feb. 1976.

[17] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimal decoding algorithm," *IEEE Trans. Inf. Theory*, vol. IT-13, pp. 260–269, Feb. 1967.

[18] G. D. Forney, "The Viterbi algorithm," *Proc. IEEE*, vol. 61, pp. 268–278, Mar. 1973.

[19] C. Berrou and A. Glavieux, "Near-optimum error-correcting coding and decoding: Turbo-codes," *IEEE Trans. Commun.*, vol. 44, pp. 1261–1271, Oct. 1996.

[20] P. Robertson, E. Villebrun, and P. Hoeher, "A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log-domain," in *Proc. Int. Conf. Commun.*, Seattle, WA, Jun. 1995, pp. 1009–1013.