REDUCED COMPLEXITY JOINT SOURCE-CHANNEL TURBO DECODING OF ENTROPY CODED SOURCES

Karim Ali Fabrice Labeau

Center for Advanced Systems and Technologies in Communications (SYTACom)
Department of Electrical & Computer Engineering
McGill University Montreal, Quebec, 3480 University Street, H3A 2A7
Tel: 514.398.2002, 514.398.7140, Fax: 514.398.4470, Email:{karim, flabeau}@tsp.ece.mcgill.ca

ABSTRACT

In [6], we presented a new joint source channel turbo decoding algorithm for entropy coded sources. The algorithm was based on a Bayesian network representation of the coding chain, and incorporates three types of redundancies, namely the source memory, the residual redundancy of the source coder and the artificial redundancy introduced by the channel coder. The algorithm was shown to yield several decibels in gain in the symbol error rate along with a drastic reduction in computational complexity, when compared with the existing one [5]. Here we present a novel technique that allows for considerable computational savings with virtually no performance losses. The technique essentially entails declaring certain bits known, early on in the iterative process. Experimental results are presented and discussed.

1 INTRODUCTION

Practical systems that utilize tandem encoding must necessarily use finite block lengths for the source coder and channel coder. This in turn implies that the received data stream posses additional redundancies, namely the residual redundancy of the source encoder and the remaining source memory, that are ignored by a tandem *decoding* scheme. A natural consideration, therefore, is the design of a joint decoding scheme, specifically for such systems, that would take these additional sources of natural redundancies into account; a possibility mentioned as early as Shannon's [1] seminal paper. Such a design strategy is motivated further by the fact that optimal source coders of the variable length code (VLC) variety have corresponding source decoders that are extremely sensitive to noise: the lack of set symbol boundaries resulting in a vulnerability to synchronization errors. Joint decoders have been shown to reduce the effects of de-synchronization and generally improve the overall decoding performance [2]-[6].

The authors in [3] developed a generic solution to the joint decoding problem by deriving the product finite state machine (FSM) model of the source, the source coder and the channel coder. Various algorithms such as Hard Viterbi, Soft Viterbi and BCJR (Kalman smoothing) are then readily applicable yielding the *optimal* solution with respect to the algorithms' criteria. Unfortunately, this solution has intractable complexity. This unaffordable phenomenon, leads to the need for less complex and therefore sub-optimal joint decoders. In this context, the authors in [4] provided a sub-optimal joint decoding solution under the additional assumption of a memoryless source. Specifically, their proposed algorithm uses the principle of turbo-decoding and alternates the use of a soft VLC decoder with a soft channel decoder. This approach was recently extended in [5] to include sources with memory. The algorithm was derived in the context of Bayesian networks. In [6], we presented a new joint decoding algorithm that was shown to yield significant gains in performance along with a drastic reduction in computational complexity when compared with [5]. In this paper, we present a novel technique that allows further significant reductions in computational complexity albeit with small performance losses.

2 TURBO JOINT DECODING VIA BAYESIAN NETWORKS

Bayesian networks provide a graphical representation of statistical problems based on the *factoring* of their joint distribution into conditional distributions. The resulting graph may then be used to incorporate new knowledge as particular nodes (random variables) are instantiated. Belief Propagation (BP), which we henceforth refer to, essentially performs maximum-a-posteriori (MAP) estimation of each random variable in the graph by means of local message passing. BP may be either locally triggered by each node or *organized*, in two passes equivalent to the BCJR algorithm.

In [6], we derived a new algorithm for turbo joint decoding in the context of Bayesian networks. Here we briefly summarize our methodology. Let an order one, finite alphabet, stationary Markov source[1](MS) generate symbols $S = S_1 \dots S_N$. These symbols are subsequently mapped via a block length one[2], binary source coder (SC) into a sequence of information bits $U = U_1 \dots U_K$. Deriving the Bayesian network corresponding to the MS+SC model is not a difficult exercise and essentially requires deriving the product FSM model of the MS and SC. Let X represent the state variable of this product FSM and suppose that the information bits are sent to a channel coder (CC) yielding a sequence R of redundant bits. We may with no loss in generality assume the CC to take information bits one at a time and yield a number of redundant bits, possibly none. We denote by X', the corresponding state variable of such a model. The derived Bayesian network [6] of all three elements (MS,SC and CC) is shown in figure 1 where we have assumed a rate 1/2 systematic channel encoder. The node connected to X_K rep-

This research was supported in part by a grant from FQRNT

[1,2] these assumptions may be relaxed and are used here for simplicity

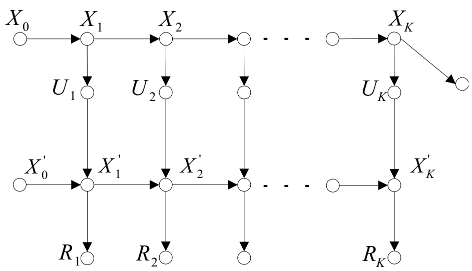


Fig. 1 Bayesian network corresponding to the MS+SC+CC [6].

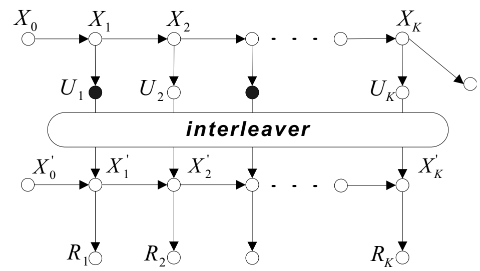


Fig. 2 Bit Simplification

resents a constraint on symbol termination and essentially ensures that the latter indeed corresponds to the end of a symbol. Such a graph may then be used to provide MAP estimates on the quantities of interest, given observations on R_k's and possibly U_k's in the case of a systematic channel code. This scheme would take into account three sources of redundancy: the source memory, the residual redundancy of the source coder and the artificial redundancy introduced by the channel coder. Unfortunately, the graph is not singly connected due to the presence of undirected cycles and hence the local message passing algorithm of BP is not guaranteed to converge to correct posterior probabilities.

Our proposed algorithm in [6] follows the lines of the algorithms in [5] in that an interleaver is inserted between the MS+SC model and the CC model in order that the graph be locally approximated by a singly connected network. However iterative decoding is simply achieved via a specific ordering of node activation: namely one iteration consists in activating $X'_0, \ldots, X'_K, \ldots, X'_0$, activating $U_1, \ldots, U_K$ and finally activating $X_0, \ldots, X_K, \ldots, X_0$. Our algorithm was shown to yield up to several decibels in gain in the symbol error rate (SER) along with a drastic reduction in computational complexity (up to 80%) when compared with [5]. Both the increase in performance and the decrease in complexity are essentially due to the derived graph's topology (figure 1) which exhibits far less undirected cycles and relaxes a stringent assumption with respect to the statistical dependence of certain nodes.

3 PROPOSED SIMPLIFICATION TECHNIQUE

Although our proposed algorithm in [6] has tractable complexity, it remains significantly more computationally expensive than its tandem decoding counterpart. Interestingly, it was noted that whilst effecting the iterative decoding, during which soft values are exchanged between neighboring nodes, the beliefs of most information bits rapidly converge and with high confidence to either of the two possible values of 0 or 1. With this context in mind, our proposed simplification entails declaring, early-on in the iterative process, certain information bits to be known and accordingly set the latter via corresponding hard decisions to values of 0 or 1. Given a threshold $\theta \in [0,1]$, this simply simply translates to the following: if $Pr\{U_k = 0\} > \theta$ then u_k is set to 0; if $Pr\{U_k = 1\} > \theta$, u_i is set to 1; else the soft information on U_k is kept. This is depicted conceptually in figure 2 where the simplified bits have been grayed. In terms of the algorithm of BP, our proposed simplification entails that we perform,

$$\hat{u_k} = \arg\max_{u_k} Bel(u_k)$$

$$Bel'(u_k) = \begin{cases} \delta_{u_k,\hat{u_k}} & \text{if } \max_{u_k} Bel(u_k) > \theta \\ Bel(u_k) & \text{otherwise} \end{cases}$$

prior to one of the iterations. Note that the quantity $Bel(\cdot)$ is simply shorthand notation for the posterior probability mass function of a variable given all observations.

The question of when to effect the bit simplification arises. Given the observed behavior of our joint decoding algorithm in [6], where an important gain is exhibited between the first and second iteration and very little gains for the subsequent iterations, the most natural choice is to effect the bit simplification prior to the second iteration.

3.1 Complexity

Let $|\mathcal{X}|$ and $|\mathcal{X}'|$ represent the cardinalities of the state-spaces of X and X' respectively. A *black box* implementation of BP for our algorithm in [6] leads to a complexity of $O(|\mathcal{X}|^2) + O(|\mathcal{X}'|^2)$. It is possible, however, to consider an efficient implementation of BP specifically adapted for the graph in question that leads to a complexity of $O(|\mathcal{X}|) + O(|\mathcal{X}'|)$. This is mainly, because the conditional probability measures quantifying the graph's directed links contain a significant number of zero entries which may be ignored.

The proposed technique will result in complexity reduction in both the black box case and the efficient implementation, although more so in the former than the latter. The reduction in complexity stems simply from the fact that those information bits that are declared known, would henceforth deliver constant messages to their neighbors and therefore need not be activated further. Further, the neighboring nodes also need not compute messages to those instantiated bits. Seen in the context of trellis decoding, our technique is essentially equivalent to trellis pruning.

Letting η represent the total number of bits simplified, such a scheme would result in a percentage operations (per iteration and for an efficient implementation),

$$\frac{15K|\mathcal{X}| - 7\eta|\mathcal{X}| + 16K|\mathcal{X}'| - 8\eta|\mathcal{X}'|}{15K|\mathcal{X}| + 16K|\mathcal{X}'|} \ (\times) \tag{1}$$

$$\frac{11K|\mathcal{X}| - 5\eta|\mathcal{X}| + 9K|\mathcal{X}'| - 2\eta|\mathcal{X}'|}{11K|\mathcal{X}| + 9K|\mathcal{X}'|} \ (+) \tag{2}$$

where $(\times)$ denotes multiplication and $(+)$ denotes addition operations. Note that equations 1 and 2 assume that the iterative procedure is carried out normally, namely by activating all of the graph's nodes in the aforementioned order under the assumption that constant messages are not to be re-computed.

3.2 Performance

It is clear that the performance of our proposed simplification will largely depend on the value of the chosen threshold θ. Indeed, simplifying bits entails the removal of information available to the joint decoder, a removal embodied by changing soft information to hard information. Hence with a lower threshold θ, performance is expected to be degraded.

However, one point merits particular attention. Indeed, the instantiation of a particular information bit has the beneficial effect of *blocking* a number of undirected cycles from the perspective of BP. Indeed, the local message passing algorithm of BP operates on the assumption that for each node, messages collected from that node's parents and children emanate from disjoint data sets and are hence statistically independent. Consider the case of the undirected cycle shown in figure 3. When node X_k is activated, it will assume that the message from U_k and the message from X_{k-1} are statistically independent which is in fact not the case. The same applies to all other nodes in the undirected cycles. However if node U_k were to be instantiated, the undirected

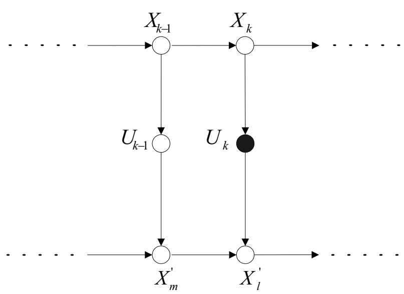


Fig. 3 Blocking the undirected cycle $X_{k-1}, U_{k-1}, X'_m, X'_l, U_k, X_k$. Note that due to the interleaver, node U_i is generally not connected to node X'_i.

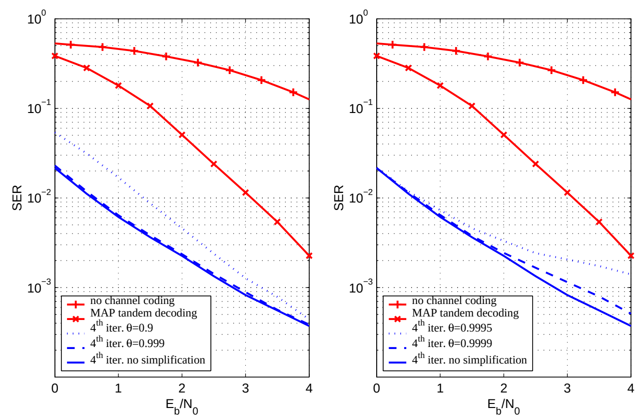
cycle is blocked in that for every node in the cycle, messages collected from its neighbors are now statistically independent. This idea was introduced by Pearl in [7] and was part of our original inspiration for our proposed simplification technique. For a rigorous proof, the reader is referred to [8].

Note that all undirected cycles that are formed with X_k, U_k, X'_l are also blocked. Hence every simplified (instantiated) information bit entails the removal of approximately K undirected cycles. This reduction in the number of undirected cycles necessarily implies that our graph may be better locally approximated by a singly connected graph. This in turn implies a better approximation of posterior probabilities of each node and therefore a better performance in the context of decoding.

4 RESULTS AND DISCUSSION

Experiments were performed on an order one stationary Gauss-Markov source with zero mean, unit variance and correlation coefficient 0.9. The source is quantized with a 3-bit uniform quantizer on the interval $[-3, 3]$. Blocks of $N = 50$ symbols were decoded at a time. The source coder is a Huffman code designed for the source statistics. The channel coder is a rate $3/4$ recursive systematic convolutional coder with 5 bits of memory. An additive white gaussian noise (AWGN) channel was assumed along with a BPSK modulation on the transmitted bits. In all the graphs to follow, it was assumed that the joint decoder has no knowledge of the total number $N = 50$ of transmitted symbols.

Figure 4 (a) shows the symbol error rates (SER) for different channel E_b/N_0 (coded) in the case where bit simplification is effected prior to the second iteration. The top curve shows the case when Hard Huffman decoding is used directly on the received bit stream. The second curve shows the performance of sequential decoding, namely soft-input MAP channel decoding followed by hard Huffman decoding. The last curve is the performance of our algorithm at the 4^{th} iteration and with no bit simplification. The intermediate curves show the performance at the 4^{th} iteration for a bit simplification prior to the second iteration and with thresholds $\theta = 0.9$ and $\theta = 0.999$. Note that both curves possess the same slope as the optimal performance. Whereas, for the $\theta = 0.9$, a performance degradation of approximately 1-dB is observed, the $\theta = 0.999$ case shows almost no loss in performance (0.05 dB). This indicates that those information bits whose beliefs converge with high confidence to either 0 or 1 after the first iteration will likely maintain the same beliefs throughout the iterative process.



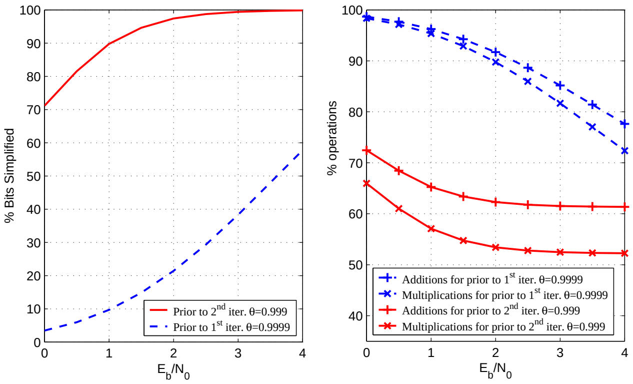
(a) Simplifying prior to the second iteration. Performance with thresholds $\theta = 0.9$ and $\theta = 0.999$.

(b) Simplifying prior to the first iteration. Performance with thresholds $\theta = 0.9995$ and $\theta = 0.9999$.

Fig. 4 Bit simplification prior to the second iteration and prior to the first iteration compared with no bit simplification.

Figure 4 (b) shows the the SER for different channel E_b/N_0 (coded) for a bit simplification prior to the first iteration. The order of the curves is as Figure 4 (a). The intermediate curves show the performance at the 4^{th} iteration for a bit simplification prior to the first iteration and with thresholds $\theta = 0.9995$ and $\theta = 0.9999$. Note that in this case, the bit simplification is entirely based on the pointwise observations on the information bits and the behavior of the algorithm is very different. Both curves follow the optimal performance at low E_b/N_0 before suffering from a performance degradation. It does appear however that the threshold θ dictates not only at which E_b/N_0 point performance will start degrading but also how fast performance will degrade.

Interestingly enough, performing bit simplification prior to the second iteration is also more beneficial in terms of computational savings. Figure 5 (a) shows the percentage of bits simplified for different channel E_b/N_0 (coded) for the case of simplifying prior to the first iteration with $\theta = 0.9999$ and for the case of simplifying prior to the second iteration with $\theta = 0.999$. Figure 5 (b) shows the corresponding percentage operations performed with respect to the optimal. Note that for a simplification prior to the second iteration



(a) Percentage of information bits simplified for different channel E_b/N_0

(b) Fraction of operations performed per iteration after simplification with respect to optimal and for different E_b/N_0 .

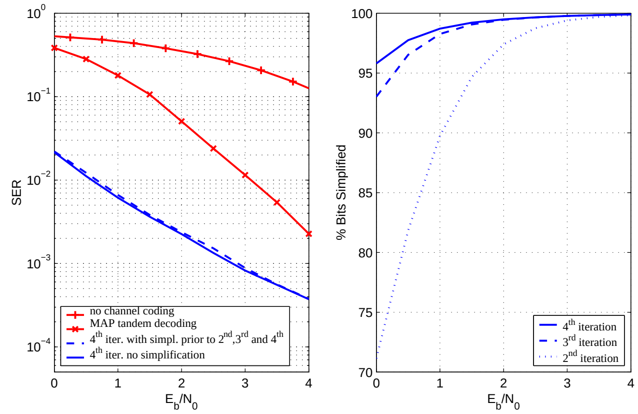
Fig. 5 Bit simplification prior to the first iteration with $\theta = 0.9999$ and for a simplification prior to the second iteration with $\theta = 0.999$.

with $\theta = 0.999$, savings of up to 48% in the multiplication and 39% in the addition operations are attainable at high E_b/N_0 for all subsequent iterations and with virtually no loss in performance.

It becomes clear that effecting the bit simplification prior to the second iteration presents the best solution and that those information bits whose beliefs converge with high confidence after the first iteration will likely maintain them. This naturally leads to the question of whether or not this behavior remains true for the subsequent iterations. In Figure 6(a) we simulated the case where bit simplification with $\theta = 0.999$ is effected at *every* iteration after the first. Note again that virtually no performance loss (0.05 dB) is observed at the fourth iteration, confirming our hypothesis. Figure 6(b) shows the percentage of bit simplified for such a method for every iteration. Note that the strategy of simplifying information bits at every iteration after the first provides with considerably more computational savings for the third and fourth iterations were the percentage of bits simplified is above 95% at every E_b/N_0 point.

5 CONCLUSION

A simplification technique for our algorithm in [6] was presented. This technique allows for considerable computational savings with virtually no computational losses and is simply based on declaring information bits, whose beliefs converge with high confidence, as known. The success of this technique indicates indeed that those simplified instantiated bits likely maintain their beliefs in any case throughout the iterative process. Most beneficial is the strategy of applying such a simplification after the first iteration and for every iteration to follow. It is important to note that this strategy exhibits such a high percentage of information bits simplified that it becomes possible to consider another alternative for



(a) SER for different channel E_b/N_0

(b) Percentage of information bits simplified per iteration for different channel E_b/N_0.

Fig. 6 Bit simplification prior to the second, third and fourth iterations with $\theta = 0.999$.

effecting the iterations. This would involve activating the neighborhoods of only those non-simplified bits and would result in computational savings of the same importance as the percentage bits simplified. Finally, it is interesting to note that these results also indicate that that our algorithm would function well if messages were to be quantized.

References

[1] C. E. Shannon, “A mathematical theory of communication”, *Bell Systems Technical Journal*, vol. 27, pp. 379-423, July-Oct. 1948.

[2] D. J. Miller, M. Park, “A sequence-based approximate MMSE decoder for source coding over noisy channels using discrete hidden Markov models”, *IEEE Journal Transactions on Communications*, vol. 46, pp. 222-231, Feb. 1998.

[3] A. H. Murad, T. E. Fuja , “Joint source-channel decoding of variable length encoded sources”, *in Proc. Inform. Theory Workshop, ITW*, June 1998, pp. 94-95.

[4] R. Bauer, J. Hagenauer , “Iterative source/channel decoding based on a trellis representation for variable length codes”, *in Proc. Int. Symp. Information Theory, ISIT* , June 2000, p. 238.

[5] A. Guyader, E. Fabre, C. Guillemot, M. Robert, “Joint source-channel turbo decoding of entropy-coded sources”, *IEEE Journal on Selected Areas in Communications*, Vol. 19, 9, Sept. 2001 pp. 1680-1696.

[6] K. Ali, F. Labeau,“Joint source-channel turbo decoding of entropy coded sources: a reduced complexity, higher performance algorithm”, *Vheciular Technology Conference*, Fall 2005.

[7] J. Pearl,“Fusion, propagation, and structuring in belief networks”, Artificial Intelligence, vol. 29, pp. 241-288, 1986.

[8] T. Verma, J. Pearl, “Causal networks: semantics and expressiveness”, *Uncertainty in Artificial Intelligence*, vol. 4, pp. 69–76, 1990.