4th INTERNATIONAL CONFERENCE ON ENGINEERING TECHNOLOGIES - ICET 2009
Novi Sad, April 28-30, 2009

BIT DECODING VERSUS SYMBOL DECODING IN MULTI-BINARY TURBO DECODERS

Horia Balta[1*], Maria Kovaci[2*], Ciprian Botiz[3], Cristian Poenaru[4]
[1,2,3,4]Department of Communications,
University "Politehnica" of Timişoara, Faculty of Electronics and Telecommunications,
Postal address, 30223 Timişoara, Romania,
*Contact person: maria.kovaci@etc.upt.ro, horia.balta@etc.upt.ro

Abstract: *The Maximum A Posteriori (MAP) decoding algorithms are used in convolutional decoders from the single-binary turbo-codes. They compute the log-likelihood ratio (LLR) at each iteration for the issued sequence. In the case of multi-binary turbo codes the decoding can be made in the same way as in the case of single-binary turbo codes. But, for the multi-binary turbo codes (MBTC) there is another possibility to compute a posteriori probabilities (APP) for each entering symbol: symbol decoding. This paper proposes to analyze the brought benefits from the point of view of BER and FER performance, of the symbol/character-wise decoding in comparison with the bit-wise decoding, in the case of convolutional turbo coders that use the MAP decoding types algorithms.*

Key Words: multi binary turbo codes, symbol decoding, MAP decoding algorithm

1 INTRODUCTION

In its original form, [1], the turbo code or the parallel concatenated convolutional code presents a single binary input. Formally, the structure of a Multi-Binary Turbo-Code (MBTC) is the same as a binary one, Fig. 1.

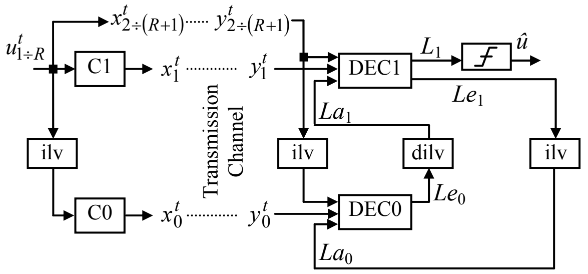

Fig. 1. Scheme of the multi-binary turbo code

However, there are few functional differences [2], differences that give some advantages [3]. The entry of the turbo-code consists of an R-bit bus, a block of data consists of a matrix of $N = R \times T$ bits, i.e. in a sequence of T symbols, each symbol consisting of R bits. Also, the interleaving implies both symbol mixing (inter-symbol interleaving) and bits mixing on the symbol (intra-symbol interleaving), [4]. Decoding can be done per bit, [5] or per symbol, [6]. Bit-wise decoding is the same as that used in uni-binary turbo codes (UBTC), [1].

Symbol-wise decoding is a "a finer analysis" of the decoder on the received sequence. Thus, considering a sequence of data of length N-bit, the uni-binary decoder, like the multi-binary decoder with bit-wise decoding, computes N LLR values. Definitely for the multi-binary encoder with R inputs the data sequence of $R \times T$ bits is organized in a matrix with the size $R \times T$. This means in a sequence of T symbols, each symbol contains R bits. In this case the symbol multi-binary decoding involves the finding of 2^R APP values for each symbol from those T. There will be a total amount of $T \cdot 2^R$ APP values for the whole bit sequence which represent „a finer analysis" of the received sequence, when R is bigger.

In this paper we have investigated the two methods of decoding, bit-wise decoding and symbol-wise decoding, in the case of multi-binary convolutional turbo codes which, in decoding, are using the Maximum A posteriori (MAP) algorithms through simulations, in order to establish the hierarchy on BER and FER performance.

The paper is organized as follows. Sections II and III present the bit-wise decoding and symbol-wise decoding, respectively, of the MBTCs. Section IV presents some simulation results and section V is dedicated to some concluding remarks.

2 THE BIT-WISE DECODING ALGORITHM

If we consider that the two decoders DEC1 and DEC0 implement the MAP iterative decoding algorithm, the bit-wise decoding of the multi-binary turbo codes is similarly with the case of uni-binary turbo codes.

The MAP algorithm, proposed by Bahl, Cocke, Jelinek and Raviv, [7], calculate the Log Likelihood Ratio, LLR, for each bit, u_r^t, with $1< t < T$, and $1< r < R$, of the each symbol of the original data sequence $u = [u^1\ u^2\ \ldots\ u^T]$, where $u^t = [\,u_1^t\ u_2^t \ldots u_R^t\,]^{\mathrm{T}}$, under the form:

$$L_r^t = \left(\frac{\sum\limits_{\substack{(\hat{s},s)\Rightarrow \\ u_r^t=+1}} \alpha_{t-1}(\hat{s})\cdot \gamma_t(\hat{s},s)\cdot \beta_t(s)}{\sum\limits_{\substack{(\hat{s},s)\Rightarrow \\ u_r^t=-1}} \alpha_{t-1}(\hat{s})\cdot \gamma_t(\hat{s},s)\cdot \beta_t(s)} \right), \qquad (1)$$

where $\alpha_{t-1}(\hat{s})$ is the probability that the encoder trellis was in $\hat{s}$ state at instant t-1 and the received channel sequence, before this moment, is $[y^1\ y^2\ \ldots\ y^{t-1}]$, with $y^j=[\,y_0^j\ y_1^j \ldots y_{R+1}^j\,]^{\mathrm{T}}$.

The alpha (or forward) coefficients can be calculated recursively as:

$$\alpha_t(s) = \sum_{\text{all}\,\hat{s}} \gamma_t(\hat{s},s)\cdot \alpha_{t-1}(\hat{s}), \qquad (2)$$

$\beta_t(s)$ is the probability that, having been given the trellis state s at instant t, the received channel sequence, after this moment, to be $[y^t\ y^{t+1}\ \ldots\ y^T]$,

$$\beta_{t-1}(\hat{s}) = \sum_{\text{all}\,s} \gamma_t(\hat{s},s)\cdot \beta_t(s), \qquad (3)$$

$\gamma_t(\hat{s},s)$ is the probability that the encoder trellis took the transition from state $\hat{s}$ to state s and the received channel sequence for this transition is y^t:

$$\gamma_t(\hat{s},s) = \exp\left(\sum_{r=1}^{R} u_r^t \cdot La_r^t / 2 + \sum_{r=0}^{R} L_c \cdot y_r^t \cdot x_r^t \right) \qquad (4)$$

In relation (4), La_r^t is the extrinsic information for the r-th bit of the t-th symbol and Lc is the reliability factor.

The iterative turbo-decoding process supposes the exchange of extrinsic information calculated at the output of the two decoders with the relations:

$$Le_{1,r}^{t,i} = L_{1,r}^{t,i} - La_{1,r}^{t,i} - L_c \cdot y_{1,r}^t / 2,$$
$$Le_{0,r}^{t,i} = L_{0,r}^{t,i} - La_{0,r}^{t,i} - L_c \cdot y_{0,r}^t / 2. \qquad (5)$$

The a priori information, at each i iteration, are obtained through interleaving of the sequences of the extrinsic information:

$$La_{1,r}^{t,i} = \pi^{-1}\left(Le_{0,r}^{t,(i-1)}\right),$$
$$La_{0,r}^{t,i} = \pi\left(Le_{1,r}^{t,(i-1)}\right). \qquad (6)$$

The operations $\pi(\ )$ and $\pi^{-1}(\ )$ signifies the interleaving and de-interleaving, respectively (marked with „ilv” and „dilv” in Fig. 1).

3 THE SYMBOL-WISE DECODING ALGORITHM

Symbol-wise decoding implies that for each character from data sequence to compute the set a posteriori probability (APP). In this case, the APP extrinsic and a priori probabilities refer, in every time moment t, to a character from the 2^R possible. Considering the notations from Fig. 1, we can write the equations:

$$L_1^{t,i}(d) = La_1^{t,i}(d) + Y_1^t + Le_1^{t,i}(d),$$
$$L_0^{t,i}(d) = La_0^{t,i}(d) + Y_0^t + Le_0^{t,i}(d), \qquad (7)$$

where $L_j^{t,i}(d)$ and $Le_j^{t,i}(d)$ are the probabilities (a posteriori – APP, extrinsic respectively, computed by the decoder j, j = 0 or 1, at iteration i and step t) that u^t takes the value $d \in J = \{0, 1 \ldots 2^R\text{-}1\}$. (A decimal expression is considered for all possible words u^t.)

So, a priory probability can be computed as:

$$La_1^{t,i}(d) = \pi^{-1}\left(Le_0^{t,(i-1)}(d)\right),$$
$$La_0^{t,i}(d) = \pi\left(Le_1^{t,(i-1)}(d)\right). \qquad (8)$$

Y_1^t and Y_0^t are terms which are computed based on the received sequence and on the noise dispersion σ^2:

$$Y_j^t = \frac{1}{\sigma^2}\cdot \sum_{r=2}^{R+1} x_r^t \cdot y_r^t + \frac{1}{\sigma^2}\cdot x_j^t \cdot y_j^t, \qquad (9)$$

with j = 0 or 1. Remark that the extrinsic probabilities set is numerical equal to the branches set which leaves from a node of the trellis (each branch corresponds of one possible value of the information word)

4 SIMULATION RESULTS

For simulations we have used the MBTC proposed in [4], having the component encoder as in Fig. 2, where:

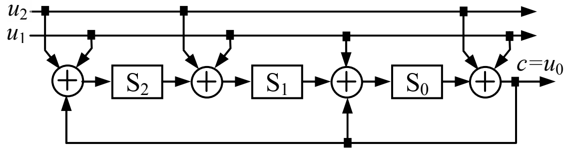

Fig. 2. Canonical configuration of a two-input rate-2/3 RSC convolutional encoder with generator matrix $G=[13\ 15\ 11]_{10}$

- S_0, S_1 and S_2 represent the encoder states, which correspond to 3-memory RSC encoders;
- u_1 and u_2 represent the input sequence bits;
- c represent the sequence of redundant bits.

For both, MBTC with bit-wise decoding and for MBTC with symbol-wise decoding, the random interleavers were used, [8]. The interleaver considered on MBTC which implies bit-wise decoding has the length equal to 1504 bits and the other one used in MBTC with symbol-wise decoding has the length equal to 752 bits.

In the case of bit-wise decoding the bits from symbol are separate decoded (for each bit one LLR is computed), so it is possible that the data block to be rearranged in one line vector having N=2T dimension and to be interleaved with an interleaver with this length (2T).

In the case of symbol-wise decoding this operation (rearranged in vector and binary interleaving) is not possible anymore at MBTC with symbol-wise decoding because the component encoders compute APPs for all symbols, so the component bits cannot be separated.

Through the simulation we have considered the AWGN channel and we have not used the puncturation of RSC codes so that the decoding rate is equal to 1/2.

In turbo decoding a maximum number of iterations, equal to 15, and a stop criterion of iterations, [9], were used.

The simulation results are shown in Fig. 3 and Fig. 4. To obtain the smallest BER and FER from the diagrams we have used 500000 blocks of 1504 bits (752 duo-binary symbols).

Obviously, the results show the superiority of symbol-wise decoding method versus bit-wise decoding method.

One remarkable fact about the obtained results is the following. If in the case of bit-wise decoding the difference between the MAP and MaxLogMAP algorithms is approximately equal to 0.5 dB, in the case of symbol-wise decoding this difference takes values smaller than 0.1 dB.

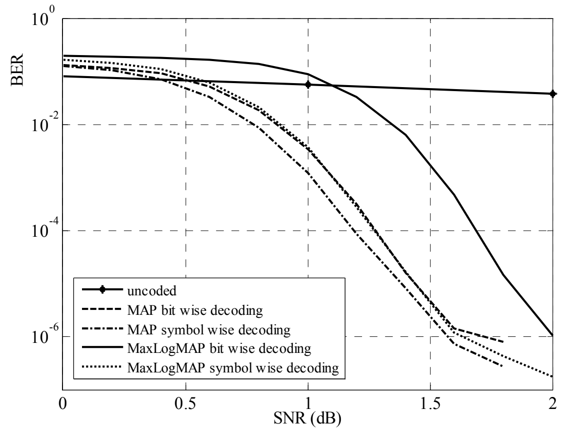

Fig. 3. BER performance of the multi-binary turbo code with RSC codes with memory 3 and with generator matrix G=[13 15 11]$_{10}$

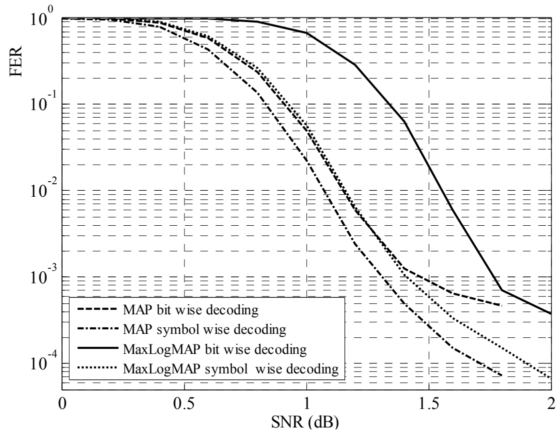

Fig. 4. FER performance of the multi-binary turbo code with RSC codes with memory 3 and with generator matrix G=[13 15 11]$_{10}$

Another important advantage of symbol-wise decoding versus bit-wise decoding is the diminution of the error floor. This effect is substantial on the FER curves, where the error floor appears immediately under a FER=10^{-3}, in the case of bit-wise decoding. But, the error floor in the case of symbol-wise decoding is practically inexistent.

5 CONCLUSION REMARKS

In this paper we analyzed the benefits brought from the point of view of BER and FER performance, of the symbol-wise decoding in comparison with the bit-wise decoding, in the case of convolutional turbo coders that use the MAP types decoding algorithms.

Analyzing the results obtained on the previous section, it results that the symbol-wise decoding method is superior to bit-wise decoding method.

As we can see on the two last figures we observe that it is better to use the MaxLogMAP algorithm with symbol-wise decoding, from a gain coding and an error floor effect point of view, instead of the MAP algorithm with bit-wise decoding.

6 REFERENCES

[1] C. Berrou, A. Glavieux, P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes", *Proc.ICC'93, Geneva*, Switzerland, May 1993, pp. 1064 – 1070.

[2] C. Berrou, M. Jézéquel, "Nonbinary convolutional codes for turbo coding", *Electron. Lett.*, vol. 35, no.1, pp.39–40, Jan. 1999.

[3] C. Berrou, M. Jézéquel, C. Douillard, and S. Kerouédan, "The advantages of nonbinary turbo codes," in *Proc. IEEE Inf. Theory Workshop, Cairns*, Australia, Sep. 2001, pp. 61–63.

[4] C. Douillard, C. Berrou, „Turbo Codes With Rate-m/(m+1) Constituent Convolutional Codes, *IEEE Transactions on Communications*, Vol. 53, No. 10, Oct. 2005, pp.1630-1638.

[5] Wey Xiang, Joint Source-Channel Coding for Image Transmission and Related Topics, www.library.unisa.edu.au/adt-root/uploads/approved/adt-SUSA-04112004-175228/public/02whole.pdf.

[6] J. Tan, G. L. Stüber, "A MAP equivalent SOVA for non-binary turbo codes", in *Proc. IEEE International Conference on Communications, ICC*, pp. 602-606, New Orleans, L. A., June 2000.

[7] L.R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal Decoding of Linear Codes for Minimising Symbol Error Rate", *IEEE Transactions on Information Theory*, Vol. 20, pp. 284-287, March 1974.

[8] S. Crozier, "New High-Spread High-Distance Interleavers for Turbo-Codes", 20th Biennial *Symposium on Communications, Kingston, Ontario*, Canada, May 28-31, 2000, pp.3-7.

[9] H. Balta, Catherine Douillard, M. Kovaci "The Minimum Likelihood APP Based Early Stopping Criterion for Multi-Binary Turbo Codes", *Buletinul Ştiinţific al Universităţii „Politehnica" Timişoara, Seria Electronică şi Telecomunicaţii*, Tom 51-65, Fascicola 2, Sept. 2006, pag.199-203.