The Decision Reliability of MAP, Log-MAP, Max-Log-MAP and SOVA Algorithms for Turbo Codes

Lucian Andrei Perişoară, and Rodica Stoian

Abstract—In this paper, we study the reliability of decisions of the MAP, Log-MAP, Max-Log-MAP and SOVA decoding algorithms for turbo codes, in terms of the a priori information, a posteriori information, extrinsic information and channel reliability. We also analyze how important an accurate estimate of channel reliability factor is to the good performances of the iterative turbo decoder. The simulations are made for parallel concatenation of two recursive systematic convolutional codes with a block interleaver at the transmitter, AWGN channel and iterative decoding with mentioned algorithms at the receiver.

Keywords—Convolutional Turbo Codes, Channel Reliability, Decision Reliability, Extrinsic Information, Iterative Decoding.

I. Introduction

IN communication systems, like cellular, satellite and computer fields, the information is represented as a sequence of binary digits. The binary message is modulated to an analog signal and transmitted over a communication channel affected by noise that corrupt the transmitted signal. The channel coding is used to protect the information from noise and to reduce the number of error bits.

One of the most used channel codes are convolutional codes, with the decoding strategy based on the Viterbi algorithm. The advantages of convolutional codes are used in Turbo Codes (TC), which can achieve performances within a 2 dB of channel capacity [1]. These codes are parallel concatenation of two Recursive Systematic Convolutional (RSC) codes separated by an interleaver. The performances of the turbo codes are due to parallel concatenation of component codes, the interleaver schemes and the iterative decoding using the Soft Input Soft Output (SISO) algorithms [2], [3].

In this paper we study the decision reliability problem for turbo coding schemes in the case of two different decoding strategies: Maximum A Posteriori (MAP) algorithm and Soft Output Viterbi Algorithm (SOVA). For the MAP algorithm we also consider two improved versions, named Log-MAP and Max-Log-MAP algorithms. The first one is a simplified algorithm which offers the same optimal performance with a reasonable complexity. The second one and the SOVA are less complex again, but give a slightly degraded performance.

The paper is organized as follows. In Section II, the turbo encoder is presented. In Section III, the turbo decoder is explained in detail, presenting firstly the iterative decoding principle (turbo principle), specifying the concepts of a priori information, a posteriori information, extrinsic information, channel reliability and source reliability. Then, we review the MAP, Log-MAP, Max-Log-MAP and SOVA decoding algorithms for which we discuss the decision reliability. In Section IV is analyzed the influence of channel reliability factor on decoding performances for the mentioned decoding algorithms. Section V presents some simulation results, which we obtained.

II. The Turbo Coding Scheme

The turbo encoder can use two different or identical Recursive Systematic Convolutional (RSC) codes, connected in parallel, see Fig. 1.

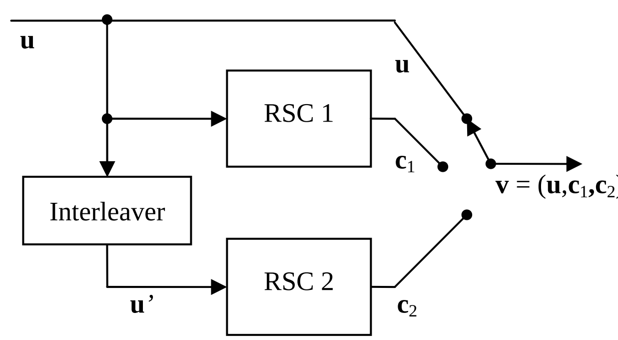

Fig. 1. The turbo encoder with rate 1/3.

The first encoder operates on the input bits represented by the frame **u**, in their original order, while the second encoder operates on the input bits which are permuted by the interleaver, frame **u**', [4]. The output of the turbo encoder is represented by the frame:

Manuscript received December 10, 2008. This work was supported in part by the Romanian National University Research Council (CNCSIS) under the Grant type TD (young doctoral students), no. 24.

L. A. Perişoară is with the Applied Electronics and Information Engineering Department, Politehnica University of Bucharest, Romania (e-mail: lucian@orfeu.pub.ro, lperisoara@yahoo.com, www.orfeu.pub.ro).

R. Stoian is with the Applied Electronics and Information Engineering Department, Politehnica University of Bucharest, Romania (e-mail: rodica@orfeu.pub.ro, rodicastoian2004@yahoo.com, www.orfeu.pub.ro).

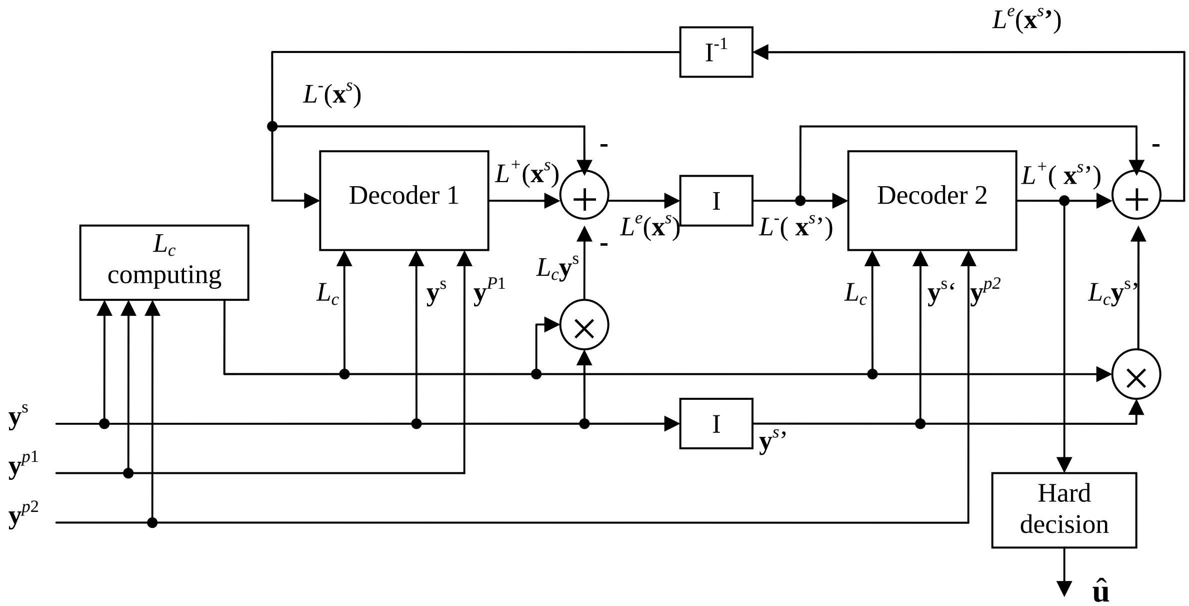

Fig. 2. The turbo decoder.

$$\mathbf{v} = (\mathbf{u}, \mathbf{c}_1, \mathbf{c}_2) = (u_1, c_{1,1}, c_{2,1}, u_2, c_{1,2}, c_{2,2}, ..., u_k, c_{1,k}, c_{2,k}), \qquad (1)$$

where frame $\mathbf{c}_1$ is the output of the first RSC and frame $\mathbf{c}_2$ is the output of the second RSC. If the input frame **u** is of length k and the output frame **x** is of length n, then the encoder rate is $R = k/n$.

For block encoding data is segmented into non-overlapping blocks of length k with each block encoded (and decoded) independently. This scheme imposes the use of a block interleaver with the constraint that the RSC's must begin in the same state for each new block. This requires either trellis termination or trellis truncation. Trellis termination need appending extra symbols (usually named tail bits) to the input frame to ensure that the shift registers of the constituent RSC encoders starts and ends at the same zero state. If the encoder has code rate 1/3, then it maps k data bits into $3k$ coded bits plus $3m$ tail bits. Trellis truncation simply involves resetting the state of the RSC's for each new block.

The interleaver used for parallel concatenation is a device that permutes coordinates either on a block basis (a generalized "block" interleaver) or on a sliding window basis (a generalized "convolutional" interleaver). The interleaver ensures that the set of code sequences generated by the turbo code has nice weight properties, which reduces the probability that the decoder will mistake one codeword for another.

The output codeword $\mathbf{v} = (\mathbf{u}, \mathbf{c}_1, \mathbf{c}_2)$ is then modulated, for example with Binary Phase Shift Keying (BPSK), resulting the sequence $\mathbf{x} = (\mathbf{x}^s, \mathbf{x}^{p1}, \mathbf{x}^{p2})$, which is transmitted over an Additive White Gaussian Noise (AWGN) channel.

It is known that turbo codes are the best practical codes due to their performance at low SNR. One reason for their better performance is that turbo codes produce high weight code words [4]. For example, if the input sequence **u** is originally low weight, the systematic **u** and parity $\mathbf{c}_1$ outputs may produce a low weight codeword. However, the parity output $\mathbf{c}_2$ is less likely to be a low weight codeword due to the interleaver in front of it. The interleaver shuffles the input sequence **u**, in such a way that when introduced to the second encoder, it is more likely probable to produce a high weight codeword. This is ideal for the code because high weight code words result in better decoder performance.

III. The Turbo Decoding Scheme

Let be the received sequence of length n, $\mathbf{y} = (\mathbf{y}^s, \mathbf{y}^{p1}, \mathbf{y}^{p2})$ where the vector $\mathbf{y}^s$ is formed only by the received information symbols $\mathbf{y}^s = (y_1^s, y_2^s, ..., y_n^s)$ and the vectors $\mathbf{y}^{p1}$ and $\mathbf{y}^{p2}$ represents the received parity symbols $\mathbf{y}^{p1} = (y_1^{p1}, y_2^{p1}, ..., y_n^{p1})$ and $\mathbf{y}^{p2} = (y_2^{p2}, y_2^{p2}, ..., y_n^{p2})$. These three streams are applied to the input of the turbo decoder presented in Fig. 2.

At time j, decoder 1 using partial received information y_j^s, y_j^{p1}, makes its decision and outputs the *a posteriori information* $L^+(x_j^s)$. Then, *the extrinsic information* is computed $L^e(x_j^s) = L^+(x_j^s) - L^-(x_j^s) - L_c y_j^s$. Decoder 2 makes its decision based on the extrinsic information $L^e(x_j^s)$ from decoder 1 and the received information $y_j^s{}', y_j^{p2}$. The term $L^+(x_j^s{}')$ is the a posteriori information derived from decoder 2 and used by decoder 1 as *a priori information* about the received sequence, noted with $L^-(x_j^s{}')$. Now, the second iteration can begin, and the first decoder decodes the same channel symbols, but now with additional information about the value of the input symbols provided by the second decoder in the first iteration. After some iterations, the algorithm converges and the extrinsic information values remains the same. Now the decision about the message bits u_j is made based on the a posteriori values $L^+(x_j^s)$.

Each constituent decoder operates based on the Logarithm Likelihood Ratio (LLR).

A. The Decision Reliability of MAP Decoder

Bahl, Cocke, Jelinek and Raviv proposed the Maximum A Posteriori (MAP) decoding algorithm for convolutional codes in 1974 [1]. The iterative decoder developed by Berrou et al. [5] in 1993 has a greatly increased attention. In their paper, they considered the iterative decoding of two RSC codes concatenated in parallel through a non-uniform interleaver and the MAP algorithm was modified to minimize the sequence error probability instead of bit error probability.

Because of its increased complexity, the MAP algorithm was simplified in [6] and the optimal MAP algorithm called the Log-MAP algorithm was developed.

The LLR of a transmitted bit is defined as [2]:

$$L(x_j^s) \overset{def}{=} \log\left(\frac{P(x_j^s=+1)}{P(x_j^s=-1)}\right) \overset{We\ noted}{=} L^-(x_j^s)\ , \tag{1}$$

where the sign of the LLR $L(x_j^s)$ indicate whether the bit x_j^s is more likely to be +1 or -1 and the magnitude of the LLR gives an indication of the correct value of x_j^s. The term $L^-(x_j^s)$ is defined as the *a priori information* about x_j^s.

In channel coding theory we are interested in the probability that $x_j^s = \pm 1$, based or conditioned on some received sequence y_j^s. Hence, we use the conditional LLR:

$$L\left(x_j^s \mid y_j^s\right) \overset{def}{=} \log\left(\frac{P\left(x_j^s=+1 \mid y_j^s\right)}{P\left(x_j^s=-1 \mid y_j^s\right)}\right) \overset{We\ noted}{=} L^+(x_j^s)\ . \tag{2}$$

The conditional probabilities $P\left(x_j^s = \pm 1 \mid y_j^s\right)$ are the a posteriori probabilities of the decoded bit x_j^s and $L^+(x_j^s)$ is the *a posteriori information* about x_j^s, which is the information that the decoder gives us, including the received frame, the a priori information for the systematic symbols y^s_j and the a priori information for symbol x^s_j. It is the output of the MAP algorithm.

In addition, we will use the conditional LLR $L\left(y_j^s \mid x_j^s\right)$ based on the probability that the receiver's output would be y_j^s when the transmitted bit x_j^s was either +1 or -1:

$$L\left(y_j^s \mid x_j^s\right) \overset{def}{=} \log\left(\frac{P\left(y_j^s \mid x_j^s=+1\right)}{P\left(y_j^s \mid x_j^s=-1\right)}\right). \tag{3}$$

For AWGN channel using BPSK modulation, we can write the conditional probability density functions, [7]:

$$P\left(y_j^s \mid x_j^s = \pm 1\right) = \frac{1}{\sqrt{\pi N_0}} \exp\left[-2\frac{E_b}{N_0}\left(y_j \mp a\right)^2\right], \tag{4}$$

where E_b is the transmitted energy per bit, a is the fading amplitude and $N_0/2$ is the noise variance.

We can rewrite the (3) as follows:

$$\begin{aligned} L\left(y_j^s \mid x_j^s\right) &= -\frac{E_b}{N_0}\left[\left(y_j^s - a\right)^2 - \left(y_j^s + a\right)^2\right] \\ &= 4a\frac{E_b}{N_0} y_j^s \overset{Noted}{=} L_c y_j^s, \end{aligned} \tag{5}$$

where $L_c = 4a\,E_b/N_0$ is defined as the *channel reliability factor* [8], [16] and $L_c y_j^s$ is the information about the reliability of the channel for the transmitted information symbol x_j^s. The term E_b is the transmitted energy per bit, a is the fading amplitude and $N_0/2$ is the noise power. For non fading AWGN channels a = 1 and $L_c = 4E_b/N_0$. The E_b/N_0 ratio is defined as the Signal to Noise Ration (SNR) of the channel.

The extrinsic information can be computed as [1], [2], [9]:

$$\begin{aligned} L^e(x_j^s) &= \log\left(\frac{P\left(x_j^s=+1 \mid y_j^s\right)}{P\left(x_j^s=-1 \mid y_j^s\right)}\right) \\ &\quad -\log\left(\frac{P\left(x_j^s=+1\right)}{P\left(x_j^s=-1\right)}\right) - \log\left(\frac{P\left(y_j^s \mid x_j^s=+1\right)}{P\left(y_j^s \mid x_j^s=-1\right)}\right) \\ &= L^+(x_j^s) - L^-(x_j^s) - L_c y_j^s. \end{aligned} \tag{6}$$

The a posteriori information defined in (2), can be written as the following [1], [10]:

$$L^+(x_j^s) = \log\frac{\sum_+ \alpha_{j-1}(s')\cdot\beta_j(s)\cdot\gamma_j^e(s',s)}{\sum_- \alpha_{j-1}(s')\cdot\beta_j(s)\cdot\gamma_j^e(s',s)}, \tag{7}$$

where $\sum_+$ is the summation over all possible transition branch pairs (s',s) in the trellis, at time j, given the transmitted symbol x^s_j = +1. Analog, $\sum_-$ is for transmitted symbol x^s_j = -1.

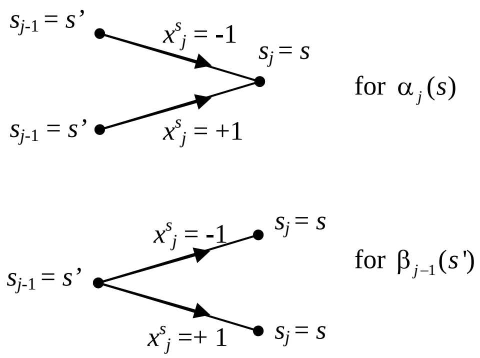

Fig. 3. Trellis states transitions.

The forward and backward terms, represented in Fig. 3 as transitions between two consecutive states from the trellis, can be computed recursively as following [7], [10], [11]:

$$\alpha_j(s) = \sum_{s'} \alpha_{j-1}(s')\gamma_j(s',s)\ , \qquad (8)$$

$$\beta_{j-1}(s') = \sum_{s} \beta_j(s)\gamma_j(s',s)\ . \qquad (9)$$

For systematic codes, which is our case, the branch transition probabilities $\gamma_j(s',s)$ are given by the relation:

$$\gamma_j(s',s) = \exp\left[\frac{1}{2}L^-(x_j^s)x_j^s + \frac{1}{2}L_c x_j^s y_j^s\right]\cdot\gamma_j^e(s',s)\ , \qquad (10)$$

where:

$$\gamma_j^e(s',s) = \exp\left[\frac{1}{2}L_c x_j^{p1} y_j^{p1} + \frac{1}{2}L_c x_j^{p2} y_j^{p2}\right]. \qquad (11)$$

At each iteration and for each frame **y**, $L^+(x_j^s)$ is computed at the output of the second decoder and the decision is done, symbol by symbol $j = 1\ldots k$, based on the sign of $L^+(x_j^s)$, original information bit u_j being estimated as [2], [3]:

$$\hat{u}_j = sign\left\{L^+(x_j^s)\right\}\ . \qquad (12)$$

In the iterative decoding procedure, the extrinsic information $L^e(x_j^s)$ is permuted by the interleaver and becomes the a priori information $L^-(x_j^s)$ for the next decoder. If $L^-(x_j^s)$ is a large (or small) positive number, then it would be difficult (or easier) to change the estimated symbol decision from +1 to -1 between to consecutive decoding stages.

For high SNR, the channel reliability value L_c will be high and this information symbol will have a large influence on $L^+(x_j^s)$. Conversely, for low SNR, the L_c is low and it's influence on $L^+(x_j^s)$ is insignificant.

B. *The Decision Reliability of Max-Log-MAP Decoder*

The MAP algorithm as described in previous section is much more complex than the Viterbi algorithm and with hard decision outputs performs almost identically to it. Therefore for almost 20 years it was largely ignored. However, its application in turbo codes renewed interest in this algorithm. Its complexity can be dramatically reduced without affecting its performance by using the sub-optimal Max-Log-MAP algorithm, proposed in [12]. This technique simplifies the MAP algorithm by transferring the recursions into the log domain and invoking the approximation:

$$\ln\left(\sum_i e^{x_i}\right) \approx \max_i(x_i)\ . \qquad (13)$$

where $\max_i(x_i)$ means the maximum value of x_i. If we note:

$$A_j(s) = \ln\left(\alpha_j(s)\right), \qquad (14)$$

$$B_j(s) = \ln\left(\beta_j(s)\right), \qquad (15)$$

and:

$$G_j(s',s) = \ln\left(\gamma_j(s',s)\right), \qquad (16)$$

then the equations (8), (9) and (10) can be written as:

$$\begin{aligned} A_j(s) &= \ln\left(\alpha_j(s)\right) = \ln\left(\sum_{s'}\alpha_{j-1}(s')\gamma_j(s',s)\right) \\ &= \ln\left(\sum_{s'}\exp\left(A_{j-1}(s') + G_j(s',s)\right)\right) \qquad (17) \\ &\approx \max_{s'}\left(A_{j-1}(s') + G_j(s',s)\right), \end{aligned}$$

$$\begin{aligned} B_{j-1}(s') &= \ln\left(\beta_{j-1}(s')\right) = \ln\left(\sum_{s}\beta_j(s)\gamma_j(s',s)\right) \\ &= \ln\left(\sum_{s}\exp\left(B_j(s) + G_j(s',s)\right)\right) \qquad (18) \\ &\approx \max_{s}\left(B_j(s) + G_j(s',s)\right), \end{aligned}$$

$$G_j(s',s) = C + \frac{1}{2}x_j^s L^-(x_j^s) + \frac{1}{2}L_c x_j^s y_j^s\ , \qquad (19)$$

where $C = \frac{1}{2}L_c x_j^{p1} y_j^{p1} + \frac{1}{2}L_c x_j^{p2} y_j^{p2}$ does not depend on the transmitted bits x_j^s and so can be considered a constant and omitted. Hence the branch metric is equivalent to that used in the Viterbi algorithm, with the addition of the a priori LLR term $x_j^s L^-(x_j^s)$.

Finally, the a posteriori LLR $L^+(x_j^s)$ which the Max-Log-MAP algorithm calculates is:

$$L^{+}(x_j^s) \approx \max_{(s',s)\, for\, u_j=+1} \left(A_{j-1}(s') + G_j(s',s) + B_j(s)\right) - \max_{(s',s)\, for\, u_j=-1} \left(A_{j-1}(s') + G_j(s',s) + B_j(s)\right). \tag{20}$$

In [12] and [13] the authors shows that the complexity of Max-Log-MAP algorithm is bigger than two times that of a classical Viterbi algorithm Unfortunately, the storage requirements are much greater for Max-Log-MAP algorithm, due to the need to store both the forward and backward recursively calculated metrics $A_j(s)$ and $B_j(s)$ before the $L^{+}(x_j^s)$ values can be calculated.

C. The Decision Reliability of Log-MAP Decoder

The Max-Log-MAP algorithm gives a slight degradation in performance compared to the MAP algorithm due to the approximation of (13). When used for the iterative decoding of turbo codes, Robertson found this degradation to result in a drop in performance of about 0.35 dB, [12]. However, the approximation of (13) can be made exact by using the Jacobian logarithm:

$$\begin{aligned}\ln(e^{x_1} + e^{x_2}) &= \max(x_1, x_2) + \ln\left(1 + \exp\left(-\,|\,x_1 - x_2\,|\right)\right) \\ &= \max(x_1, x_2) + f\left(|\,x_1 - x_2\,|\right) = g(x_1, x_2),\end{aligned} \tag{21}$$

where $f(\delta)$ can be thought of as a correction term. However, the maximization in (17) and (18) is completed by the correction term $f(\delta)$ in (21). This means that the exact rather than approximate values of $A_j(s)$ and $B_j(s)$ are calculated. For binary trellises, the maximization will be done only for two terms. Therefore we can correct the approximations in (17) and (18) by adding the term $f(\delta)$, where δ is the magnitude of the difference between the metrics of the two merging paths. This is the basis of the Log-MAP algorithm proposed by Robertson, Villebrun and Hoeher in [12]. Thus we must generalize the previous equation for more than two x_1 terms, by nesting the $g(x_1, x_2)$ operations as follows:

$$\ln\left(\sum_{i=1}^{n} e^{x_i}\right) = g\left(x_n, g\left(x_{n-1}, \ldots g\left(x_3, g(x_2, x_1)\right)\right)\right), \tag{22}$$

The correction term $f(\delta)$ need not to be computed for every value of δ, but instead can be stored in a look-up table. In [12], Robertson found that such a look-up table need contain only eight values for δ, ranging between 0 and 5. This means that the Log-MAP algorithm is only slightly more complex than the Max-Log-MAP algorithm, but it gives exactly the same performance as the MAP algorithm. Therefore, it is a very attractive algorithm to use in the component decoders of an iterative turbo decoder.

D. The Decision Reliability of SOVA Decoder

The MAP algorithm has a high computational complexity for providing the Soft Input Soft Output (SISO) decoding. However, we obtain easily the optimal a posteriori probabilities for each decoded symbol.

The Viterbi algorithm provides the Maximum Likelihood (ML) decoding for convolutional codes, with optimal sequence estimation. The conventional Viterbi decoder has two main drawbacks for a serial decoding scheme: the inner Viterbi decoder produces bursts of error bits and hard decision output, which degrades the performance of the outer Viterbi decoder [3]. Hagenauer and Hoeher modified the classical Viterbi algorithms and they provided a substantially less complex and suboptimal alternative in their Soft Output Viterbi Algorithm (SOVA). The performance improvement is obtained if the Viterbi decoders are able to produce reliability values or soft outputs by using a modified metric [14]. These reliability values are passed on to the subsequent Viterbi decoders as *a priori information*.

In soft decision decoding, the receiver doesn't assign a zero or a one to each received symbol from the AWGN channel, but uses multi-bit quantized values for the received sequence **y**, because the channel alphabet is greater than the source alphabet [3]. In this case, the metric derived from Maximum Likelihood principle, is used instead of Hamming distance. For an AWGN channel, the soft decision decoding produces a gain of 2÷3 dB over hard decision decoding, and an eight-level quantization offers enough performance in comparison with an infinite bit quantization [7].

The original Viterbi algorithm searches for an information sequence **u** that maximizes the a posteriori probability $P(\mathbf{s}\,|\,\mathbf{y})$, **s** being the states sequence generated by the message **u**. Using the Bayes theorem and taking into account that the received sequence **y** is fixed for the metric computation and it can be discarded, the maximization of $P(\mathbf{s}\,|\,\mathbf{y})$ is:

$$\max_{\mathbf{u}} \left\{P(\mathbf{s}\,|\,\mathbf{y})\right\} = \max_{\mathbf{u}} \left\{P(\mathbf{y}\,|\,\mathbf{s})P(\mathbf{s})\right\}. \tag{23}$$

For a systematic code, this relation can be expanded to:

$$\max_{\mathbf{u}} \left\{\prod_{j=1}^{k} P\left((y_j^s, y_j^{p1}, y_j^{p2})\,|\,s_{j-1}, s_j\right) P(s_j)\right\}. \tag{24}$$

Taking into account that:

$$\begin{aligned}P\left((y_j^s, y_j^{p1}, y_j^{p2})\,|\,s_{j-1}, s_j\right) &= \\ &= P\left(y_j^s\,|\,x_j^s\right)\cdot P\left(y_j^{p1}\,|\,x_j^{p1}\right)\cdot P\left(y_j^{p2}\,|\,x_j^{p2}\right),\end{aligned} \tag{25}$$

where (s_{j-1}, s_j) denotes the transitions between the states at time *j*-1 and the states at time *j*, the SOVA metric is obtained from (24) as [15]:

$$M_j = M_{j-1} + \sum_{*} x_j^* \log\left(\frac{P\left(y_j^*\,|\,x_j^* = +1\right)}{P\left(y_j^*\,|\,x_j^* = -1\right)}\right) + u_j \log\left(\frac{P(u_j = 1)}{P(u_j = 0)}\right), \tag{26}$$

where $x_j^* = (u_j, c_{1,j}, c_{2,j})$ is the RSC output code word at time

j, at channel input and $y_j^* = (y_j^s, y_j^{p1}, y_j^{p2})$ is the channel output. The summation is made for each pair of information symbols (u_j, y_j^s) and for each pair of parity symbols ($c_{1,j}, y_j^{p1}$) and ($c_{2,j}, y_j^{p2}$).

According [14] and [7], the relation (26) can be reduced as:

$$M_j = M_{j-1} + \sum_* L_c x_j^* y_j^* + u_j L(u_j), \tag{27}$$

where the source reliability $L(u_j)$, defined in (26), is the log-likelihood ratio of the binary symbol u_j. The sign of $L(u_j)$ is the hard decision of u_j and the magnitude of $L(u_j)$ is the *decision reliability*.

According [10], the SOVA metric includes values from the past metric M_{j-1}, the channel reliability L_c and the source reliability $L(u_j)$, as an a priori value. If the channel is very good, the second term in (27) is greater than the third term and the decoding relies on the received channel values. If the channel is very bad, the decoding relies on the a priori information $L(u_j)$.

If M^1_j, M^2_j are two metrics of the survivor path and concurrent path in the trellis, at time j, then the metric difference is defined as [7]:

$$\Delta_j^0 = \frac{1}{2}\left|M_j^1 - M_j^2\right|. \tag{28}$$

The probability of path m, at time j, is related as:

$$P(\text{path } m) = P(\mathbf{s}_j^m) = \exp\left(M_j^m / 2\right). \tag{29}$$

where $\mathbf{s}_j^m$ is a states vector and M_j^m is the metric. The probability of choosing the survivor path is:

$$P(\text{correct}) = \frac{P(\text{path } 1)}{P(\text{path } 1) + P(\text{path } 2)} = \frac{e^{\Delta_j^0}}{1 + e^{\Delta_j^0}}. \tag{30}$$

The reliability of this path decision is calculated as:

$$\log \frac{P(\text{correct})}{1\text{-}P(\text{correct})} = \Delta_j^0. \tag{31}$$

The reliability values along the survivor paths, for a particular node and time j, are denoted as Δ_j^d, where d is the distance from the current node at time j. If the survivor path bit for $d = j$ is the same with the associated bit on the competing path, then there would be no error if the competing path is chosen. The reliability value remains unchanged.

To improve the reliability values an updating process must be used, so the "soft" values of a decision symbol are:

$$L(u'_{j-d}) = u'_{j-d} \sum_{i=0}^{d} \Delta_j^i, \tag{32}$$

which can be approximated as:

$$L(u'_{j-d}) = u'_{j-d} \cdot \min_{i=0 \ldots d} \left\{\Delta_j^i\right\}. \tag{33}$$

The SOVA algorithm described in this section is the least complex of all the SISO decoders discussed in this section. In [12], Robertson shows that the SOVA algorithm is about half as complex as the Max-Log-MAP algorithm. However, the SOVA algorithm is also the least accurate of the algorithms described in this section and, when used in an iterative turbo decoder, performs about 0.6 dB worse than a decoder using the MAP algorithm. If we represent the outputs of the SOVA algorithm they will be significantly more noisy than those from the MAP algorithm, so an increased number of decoding iterations must be used for SOVA to obtain the same performances as for MAP algorithm.

The same results are reported also for the iterative decoding (turbo decoding) of the turbo product codes, which are based on two concatenated Hamming block codes not on convolutional codes [19].

IV. The Influence of L_C on Decoding Performance

In this section we analyze the importance of an accurate estimate of the channel reliability factor L_c is to the good performance of an iterative turbo decoder which uses the MAP, SOVA, Max-Log-MAP and Log-MAP algorithms.

In the MAP algorithm the channel inputs and the a priori information are used to calculate the transition probabilities from one state to another, that are then used to calculate the forward and backward recursion terms [2], [8]. Finally, the a posteriori information $L^+(x_j^s)$ is computed and the decision about the original message is made based on it.

In the iterative decoding with MAP algorithm, the channel reliability is calculated from the received channel values. At first iteration, the decoder 1 has no a priori information available (the $L^-(x_j^s)$ is zero) and the output from the algorithm is calculated based on channel values. If an incorrect value of L_c is used the decoder will make more decision errors and the extrinsic information from the output of the first decoder will have incorrect values, for the soft channel inputs [16].

In the SOVA algorithm the channel values $L_c y_j^*$ are used to recursively calculate the metric M_j for the current state s along a path from the metric M_{j-1} for the previous state along that path added to an a priori information term and to a cross-correlation term between the transmitted and the received channel values, x_j^* and y_j^*, using (27). The channel reliability factor L_c is used to scale this cross-correlation. When we use an incorrect value of L_c, e.g. $L_c = 1$, we are scaling the channel values applied to the inputs of component decoders by a factor of one instead of the correct value of L_c. This has the effect of scaling all the metrics by the same factor, see (8), and the metric differences are also scaled by the same factor, see (9). This scaling of the metrics do not affect the path chosen by the algorithm as a survivor path or as a Maximum Likelihood (ML) path, so the hard decisions given by the algorithm are not affected by using an incorrect value of L_c [16]-[18].

In the iterative decoding with SOVA algorithm, in the first iteration we assume that no a-priori information about the transmitted bits is available to the decoder (the a-priori information is zero), the first component decoder takes only the channel values. If channel reliability factor L_c is incorrect, the channel values are scaled, the extrinsic information will be also scaled by the same factor and the a-priori information for the second decoder will also be scaled. Because of the linearity of the SOVA, the effect of using an incorrect value of the channel reliability factor is that the output LLR from the decoder is scaled by a constant factor. The relative importance of the two inputs to the decoder, the a priori information and the channel information, will not change, since the LLRs for both these sources of information will be scaled by the same factor. In the final iteration, the soft outputs from the final component decoder will have the same sign as those that would have been calculated using the correct value of L_c. So, the hard outputs from the turbo decoder using the SOVA algorithm are not affected by the channel reliability factor [16].

The Max-Log-MAP algorithm has the same linearity that is found in the SOVA algorithm. Instead of one metric, now two metrics $A_j(s)$ and $B_j(s)$ are calculated, for forward and backward recursions, see (17), (18) and (19), were are used only simple additions of the cross-correlation of the transmitted and received symbols. But, if an incorrect value of the channel reliability value is used, all the metrics are simply scaled by a factor as in the SOVA algorithm. The soft outputs given by the differences in metrics between different paths will also be scaled by the same factor, with the sign unchanged and the final hard decisions given by the turbo decoder will not be affected.

The Log-MAP algorithm is identical to the Max-Log-MAP algorithm, except for a correction term $f(\delta) = \ln(1 + \exp(-\delta))$, used in the calculation of the forward and backward metrics $A_j(s)$ and $B_j(s)$, and the soft output LLRs. The function $f(\delta)$ is not a linear function, it decreases asymptotically towards zero as δ increases. Hence the linearity that is present in the Max-Log-MAP and SOVA algorithms is not present in the Log-MAP algorithm. This effect of non-linearity determines more hard decision errors of the component decoders if the channel reliability factor L_c is incorrect, and the extrinsic information derived from the first component decoder have incorrect amplitudes, which become the a-priori information for the second decoder in the first iteration. Both decoders in subsequent iterations will have incorrect amplitudes relative to the soft channel inputs.

In the iterative decoding with Log-MAP algorithm, the extrinsic information exchange from one component decoder to another, determines a rapid decrease in the BER as the number of iterations increases. When the incorrect value of L_c is used, no such rapid fall in the BER occurs due to the incorrect scaling of the a priori information relative to the channel inputs. In fact, the performance of the decoder is largely unaffected by the number of iterations used.

For wireless communications, some of them modeled as Multiple Input Multiple Output (MIMO) systems [23], the channel is considered to be Rayleigh or Rician fading channel. If the Channel State Information (CSI) is not known at the receiver, a natural approach is to estimate the channel impulse response and to use the estimated values to compute the channel reliability factor L_c required by the MAP algorithm to calculate the correct decoding metric.

In [20], the degradation in the performance of a turbo decoder using the MAP algorithm is studied when the channel SNR is not correctly estimated. The authors propose a method for blind estimation of the channel SNR, using the ratio of the average squared received channel value to the square of the average of the magnitudes of the received channel values. In addition, they showed that using these estimates for SNR gives almost identical performances for the turbo decoder to that given using the true SNR.

In [8], the authors proposes a simple estimation scheme for L_c from the statistical computation on the block observation of matched filter outputs. The channel estimator includes the error variance of the channel estimates. In [24], is used the Minimum Mean Squared Error (MMSE) estimation criterion and is studied an iterative joint channel MMSE estimation and MAP decoding.

None of above works requires a training sequence with pilot symbols to estimate the channel reliability factor. Other studies used pilot symbols to estimate the channel parameters, like [22] and [25].

In [22] it is shown that it is not necessary to estimate the channel SNR for a turbo decoder with Max-Log-MAP or SOVA algorithms. If the MAP or the Log-MAP algorithm is used then the value of L_c does not have to be very close to the true value for a good BER performance to be obtained.

V. SIMULATION RESULTS

This section presents some simulation results for the turbo codes ensembles, with MAP, Max-Log-MAP, Log-MAP and SOVA decoding algorithms. The turbo encoder is the same for the four decoding algorithms and is described by two identical RSC codes with constraint length 3 and the generator polynomials $G_f = 1 + D^2$ and $G_b = 1 + D + D^2$. No tail bits and no puncturing are performed. The two constituent encoders are parallel concatenated by a classical block interleaver, with dimensions variable according to the frame size. The binary input message is randomly generated and it is divided into frames of length *k*. The coded symbols are BPSK modulated and transmitted over an AWGN channel with signal to noise ratio E_b/N_0 variable between 0 ÷ 4 dB and without fading (the fading amplitude is $a = 1$).

A fixed number of iterations are used for the iterative decoding process. The Bit Error Rate (BER) is computed over 50 or 500 different frames. For comparison, the BER for uncoded system is calculated and plotted in the figures.

All the simulation parameters are mentioned in the figure caption.

A. Decision based on a posteriori information for MAP

In the iterative MAP decoder, the information exchange between one decoder and another determine a rapid decrease in BER of the turbo decoder as the number of decoding steps increase, see Fig. 4. The decisions about the transmitted bits are based on the a posteriori information after 1, 2, 3 and 10 decoding steps, which corresponds to the outputs of decoder 1 at iteration 1, decoder 2 at iteration 1, decoder 1 at iteration 2 and decoder 2 at iteration 5.

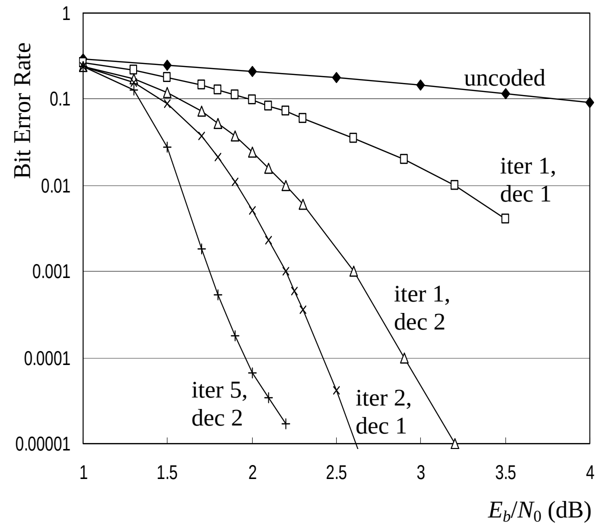

Fig. 4. BER(E_b/N_0) performance for different decoding steps and decision based on a posteriori information.
(frame size 4900 bits, 70x70 block interleaver, 500 frames, MAP).

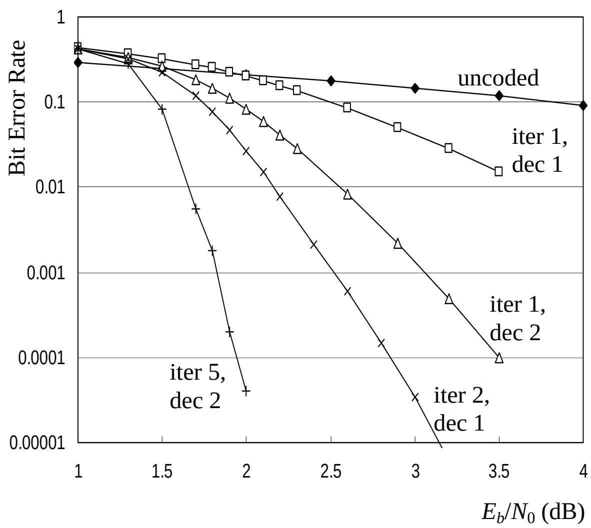

Fig. 5. BER(E_b/N_0) performance for different decoding steps and decision based on extrinsic info.
(frame size 4900 bits, 70x70 block interleaver, 500 frames, MAP).

B. Decision based on extrinsic information for MAP

If the decisions are based only on extrinsic information the simulation results for the MAP decoder are presented in Fig. 5. The performances obtained for this kind of decision are worse, with 0.5 dB for BER = 10^{-4} and for 3 decoding steps, compared with the decisions based on a posteriori information, from Fig. 4.

C. Incorrect values for channel reliability factor

Fig. 6 shows the performance of turbo decoder using a one and two iterations for the iterative decoder and the MAP algorithm for the constituent decoders. For the channel reliability factor L_c we suppose two cases: firstly, the L_c is calculated exactly using the known channel SNR at the decoder, and hence the correct value of L_c is used. Secondly, the real value of L_c is not known at the decoder and we consider a value of $L_c = 1$, which corresponds to a value of E_b/N_0 of -3 dB for the channel SNR. For MAP algorithm, the performances of iterative turbo decoder are drastically affected by the value of L_c used. If the incorrect value of L_c is used, no such rapid fall in the BER occurs and an increased number of iterations is needed for a certain BER. For a fixed value of L_c, correct or incorrect, if the number of iterations

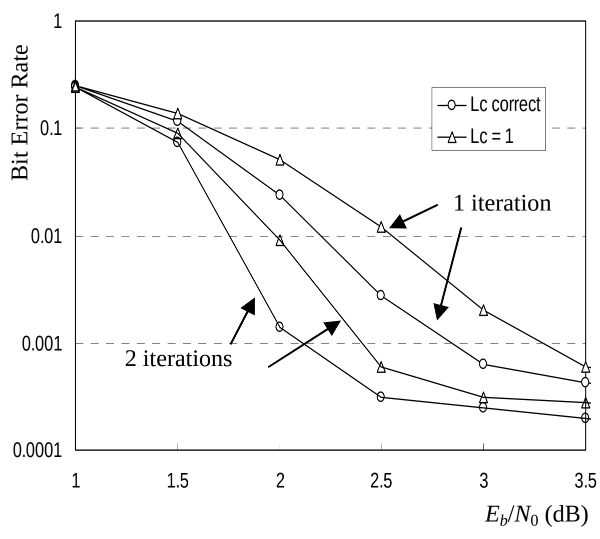

Fig. 6. BER(E_b/N_0) performance for different L_c values. (frame size 2000 bits, 40x50 block interleaver, 50 frames, MAP).

increases, the BER decreases. For higher E_b/N_0 (> 2.5 dB) and two decoding iterations, the BER dependence tends to be flat, because of a small number of bits (2000 bits x 50 frames = 100000 bits) which were coded in the simulated system.

D. *Log-MAP, Max-Log-MAP and SOVA performances comparison*

Fig. 7 shows the performances of a turbo decoder using different decoding algorithms (Log-MAP, Max-Log-MAP and SOVA) for a different number of iterations. As we discussed in Section IV, the decoding performances for the MAP and Log-MAP algorithms are the same, with some differences only in the complexity, so the MAP algorithm is not present in this figure.

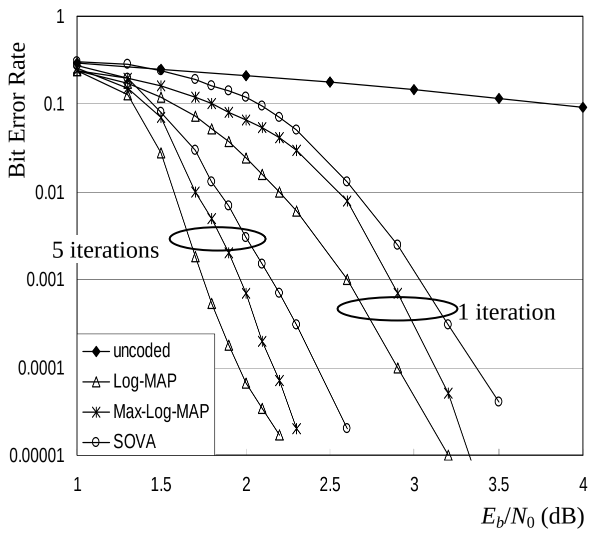

Fig. 7. BER(E_b/N_0) performance for variable number of iterations. (k=4900 bits, 70x70 block interleaver, 500 frames).

As the number of iterations increases from 1 to 5 iterations, the Log-MAP algorithm performs significantly better (about 1 dB at BER=10^{-4}). Similar results are obtained when using the Max-Log-MAP or SOVA algorithms. Also, it can be seen that the Max-Log-MAP and SOVA algorithms gives a degradation in performance of 0.3 dB, respectively 0.5 dB, compared to Log-MAP, at BER = 10^{-4}, independently of the decoding iterations.

So, the Log-MAP algorithm offers better error correction performances than the Max-Log-MAP and SOVA algorithms, but with increased complexity.

E. *The dependence of BER on channel reliability factor*

The simulation results for frame sizes k = 200 and k = 5000, in Fig. 8, concludes that the MAP an SOVA performances varies with channel reliability factor L_c.

For L_c < 6 dB the BER is almost the same. For BER between 6 dB and 10 dB, the differences between the algorithms, for a fixed frame size, increases. If the L_c is bigger that 10 dB, the difference remain constant and the dependence BER(L_c) tend to be flat. Also, both algorithms work better for bigger frame sizes. For the same frame size, and for $\mathrm{BER} = 10^{-4}$, the MAP algorithm perform about 1dB better than SOVA algorithm.

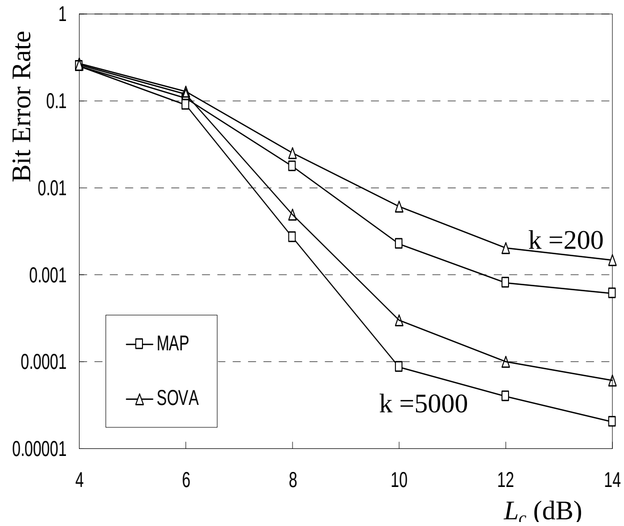

Fig. 8. BER(L_c) performance for variable frame size. (block interleaver, 3 decoding iterations, 200 frames).

VI. CONCLUSION

In this paper we have described the turbo coding system, with two parallel concatenated RSC codes at the transmitter side and with the iterative decoding which uses the MAP, Log-MAP, Max-Log-MAP and SOVA algorithms for constituent decoders, at the receiver side and we have analyzed and simulated the reliability of decision for each decoding algorithm.

It is shown that the hard decision based on a posteriori information is better than on extrinsic information. Also, the BER decrease as the number of decoding iterations increase.

Using a correct value of channel reliability factor, the BER performance increase and a decreased number of iterations is needed for a certain BER. Finally, is analyzed the influence of channel reliability factor on the BER performances of turbo coded system.

REFERENCES

[1] L. R. Bahl, J. Cocke, F. Jelinek and J. Raviv, “Optimal decoding of linear codes for minimizing symbol error rate,” *IEEE Trans. Inform. Theory*, vol. 20, pp. 284-287, 1974.
[2] C. Berrou and A. Glavieux, “Near optimum error correcting coding and decoding: turbo codes,” *IEEE Trans. Commun*., vol. 44, no. 10, pp.1261-1271, Oct. 1996.
[3] J. Hagenauer and P. Hoeher, “A Viterbi algorithm with soft decision outputs and its applications,” in *Proc. GLOBECOM 1989*, pp. 1680-1686, Dallas, Texas, Nov. 1989.
[4] H. Jin, R. J. McEliece, “Coding Theorems for Turbo Code Ensembles”, *IEEE Trans. Inform. Theory*, vol. 48, No. 6, pp. 1451-1463, June 2002.
[5] C. Berrou, A. Glavieux and P. Thitmajshima, “Near Shannon limit error-correcting coding: turbo codes,” in *Proc. of ICC ’93*, Geneva, Switzerland, pp.1064-1070, May 1993.
[6] J. Hagenauer, “The turbo principle: tutorial introduction and state of the art,” *International Symposium on Turbo Codes*, Brest, France 1997.
[7] F. Huang, “Evaluation of soft output decoding for turbo codes,” M.S. Thesis, Virginia Polytechnic Institute, 1997.
[8] Hyundong Shin, Jae Hong Lee, “Channel reliability estimation for turbo decoding in rayleigh fading channels with imperfect channel estimates,” *IEEE Communications Letters*, vol. 6, issue 11, pp. 503-505, Nov. 2002.

[9] Xuelan Zou, Guangzeng Feng, "Performance comparison of iterative decoding algorithm using the extrinsic information transfer chart," *6th Int. Conf. on Networking, ICN '07*, pp. 63-66, 22-28 April 2007.

[10] L. Hanzo, T. H. Liew, B. L. Yeap, *Turbo coding, turbo equalization and space-time coding*, John Wiley & Sons, UK, 2002.

[11] J. Hagenauer, P. Robertson and L. Papke, "Iterative turbo decoding of systematic convolutional codes with the MAP and SOVA algorithms," in *Proc. ITG*, pp. 21-29, Oct. 1994.

[12] P. Robertson, P. Hoeher, and E. Villebrun, "Optimal and sub-optimal maximum a posteriori algorithms suitable for turbo decoding," *European Transactions on Telecommunications*, vol. 8, pp. 119–125, March/April 1997.

[13] Stephan ten Brink, "Convergence behavior of iteratively decoded parallel concatenated codes," *IEEE Trans. Commun*., vol. 49, no. 10, pp. 1727-1737, Oct. 2001.

[14] C. Berrou, P. Adde, E. Angui and S. Faudeil, "A low complexity soft output viterbi decoder architecture," in *Proc. ICC 1993*, pp.737-740, Geneva, Switzerland, May 1993.

[15] J. Hagenauer, "Source controlled channel decoding," *IEEE Trans. Commun*., vol. 43, no. 9, pp. 2449-2457, Sept. 1995.

[16] L. A. Perişoară, Rodica Stoian, "Performances of turbo decoders in terms of channel reliability factor," in *Proc. of 7th WSEAS Int. Conf. on Data Networks, Communications, Computers (DNCOCO '08)*, Bucharest, Romania, pp. 46-50, 7-9 Nov., 2008.

[17] J. W. Lee and R.E. Blahut, „Convergence analysis and BER performance of finite-length turbo codes", *IEEE Trans. Commun*., vol. 55, no. 5, pp. 1033-1043, May 2007.

[18] J. Hokfelt, O. Edfors, and T. Maseng, „Turbo codes: correlated extrinsic information and its impact on iterative decoding performances," in *Proc. IEEE 49th Vehicular Technology Conference*, vol. 3, Houston, Texas, pp. 1871-1875, May 1999.

[19] Rodica Stoian, L. A. Perişoară, "Iterative decoding algorithms for turbo product codes", in *Proc. of the 5th WSEAS Int. Conf. on Applied Electromagnetics, Wireless and Optical Communications (Electroscience '07)*, Puerto de la Cruz, Spain, pp. 17-21, 14-16 Dec, 2007.

[20] Todd A. Summmers and Stephen G. Wilson, "SNR mismatch and online estimation in turbo decoding," *IEEE Trans. on Commun.*, vol. 46, pp. 421–423, April 1998.

[21] J. W. Lee, R. L. Urbanke, and R. E. Blahut, „Turbo codes in binary erasure channel," *IEEE Trans. Inform. Theory*, vol. 54, no. 4, pp. 1765-1773, April 2008.

[22] H. Shin, S. Kim, and J. H. Lee, "Turbo decoding in a Rayleigh Fading Channel with estimated channel state information," Proc. *IEEE VTC*, Fall, Boston, Ma, pp. 1358-1363, Sept. 2000.

[23] Rodica Stoian, L. A. Perişoară, "Information Capacity for a Class of MIMO Systems (revised)", *WSEAS Trans. on Commun.* Issue 9, Vol. 5, pp. 1648-1651, Sept. 2006.

[24] M. L. Ammari, H. T. Huynh, P. Fortier, "Joint iterative turbo decoding and estimation of correlated Rayleigh fading channel," the *2001 Canadian Workshop on Information Theory*, Vancouver, 3-6 June 2001, http://www.cce.com/cwit/papers/final_28_amma.pdf

[25] M. C. Valenti, B. D. Woerner, "A bandwidth efficient pilot symbol technique for coherent detection of turbo codes over fading channels," in *Military Communications Conference, MILCOM '99*, vol. 1, pp. 81-85, 1999.

[26] S. Asoodeh, "A New Constructive Approximation in Log-MAP Turbo Decoder," in Proc. of *7th WSEAS Int. Conf. on Electronics, Hardware, Wireless and Optical Communications (EHAC '08)*, Cambridge, UK, pp. 220 - 224, Feb. 20-22, 2008.

[27] M. A. U. Bhuiyan, M. S. Alam, M. S. Hosain, "BER and FER Performance of Third Generation (UMTS) Turbo Codes," *WSEAS Trans. on Commun.*, Issue 11, Vol. 4, pp. 1244-1252, Nov. 2005.

[28] D. Žagar, N. Falamić, S. Rimac-Drlje, "Comparison of Decoding Algorithms for Concatenated Turbo Codes," *WSEAS Trans. on Electronics*, Issue 4, Vol. 1, pp. 650-654, Oct. 2004.

Lucian A. Perişoară received the B.Sc. degree in Electronics Engineering and Telecommunications from the Politehnica University of Bucharest, Romania, in 2003. Currently, he is working toward his Ph.D. degree in Electronics Engineering at the Department of Applied Electronics and Information Engineering, at the same university.

Since 2003, he is Teaching Assistant at the Department of Applied Electronics and Information Engineering, Faculty of Electronics, Telecommunications and Information Technology, Politehnica University of Bucharest. His research interests include information theory and coding, random processes, decision and estimation, wireless communications, MIMO systems.

Mr. Perişoară is a student member of IEEE.

Rodica Stoian received the B.Sc. degree in Electronics and Telecommunications from the Politehnica University of Bucharest, Romania, in 1966, the M.Sc. degree in Informatics and Applied Mathematics for Research from the Faculty of Mathematics, University Bucharest, Romania and the Ph.D. degree in Electronics and Telecommunications from the Politehnica University of Bucharest, in 1985.

She is a Professor at the Department of Applied Electronics and Information Engineering, Faculty of Electronics, Telecommunications and Information Technology, Politehnica University of Bucharest. Her research interests covers a wide area of information transmission theory and its applications, such as data compression and coding, including compression techniques and error control coding for multimedia systems, burst error correcting codes for computer techniques, finite fields arithmetic architectures for test pattern generation, sources coding and channel capacity for multi-user systems. Her recent research interests are centered on information theory aspects applied to network systems (network coding).

Prof. Stoian is a member of IEEE and WSEAS.