ROBUST TRANSMISSION OF HTML FILES : ITERATIVE JOINT SOURCE-CHANNEL DECODING OF LEMPEL ZIV-77 CODES

Zied Jaoua[*+]*, Anissa Zergaïnoh-Mokraoui*[+]*, Pierre Duhamel*[*]

[*]LSS/CNRS, SUPELEC, Plateau de Moulon, 91 192 Gif sur Yvette, France
[+]L2TI, Institut Galilée, Université Paris 13, Avenue Jean Baptiste Clément, 93 430 Villetaneuse, France
{jaoua,duhamel}@lss.supelec.fr, zergainoh@galilee.univ-paris13.fr

ABSTRACT

This paper concerns the error correction of corrupted compressed HTML pages during their transmission via a noisy mobile channel. The proposed receiver is based on an iterative joint source channel decoding approach similar to the turbo decoding of two serial concatenated codes. We propose a soft-Input soft-Output Lempel-Ziv inner decoder based on the modified version of the traditional sequential decoding M-algorithm. This new algorithm exploits the specific grammatical rules of the Lempel-Ziv-77 codes which are combined to the syntax of the HTML language. This source decoder is combined to a Soft-Input Soft-Output channel decoder of convolutional codes. Simulation results, over an additive white gaussian noise channel, show that the proposed method drastically reduces the number of files in error compared to any conventional channel decoding.

Index Terms— Entropy codes, Lempel-Ziv codes, Iterative methods, Sequential decoding, Hypertext systems

1. INTRODUCTION

The Hyper-Text Transfer Protocol (HTTP) defines specifications allowing the web clients (e.g.browser running on mobiles or computers) to request and to transfer web pages from web servers. Generally, a web page contains HTML (or XHTML) files and many referenced objects such as JPEG images, GIF images, Java applet, ...

The recent release HTTP1.1 standard [1] is designed to address the problems concerning the congestion control occurring in a network. One specific way of reducing congestion problems is to compress the HTML (or XHTML) files by some lossless code, thus reducing the required resources between the server and the client. In HTTP1.1, the HTML (or XHTML) files are compressed using a variant of Lempel-Ziv lossless compression algorithm optionally followed by Huffman encoding. The compressed files are then transmitted or downloaded according to the Gzip (rfc1952) [2] or Zip (rfc1950) [2] format. However, the compressed bitstream is very sensitive to any transmission error which may arise in a noisy channel. One can easily imagine how the loss or alteration of a single bit in the compressed bitstream could affect the decoding of the HTML (or XHTML) file. In that respect, the lack of error resilience in the Lempel-Ziv algorithm is a real problem. To our knowledge, this problem has been addressed only in reference [3]. The authors proposed a joint source channel encoding algorithm using Reed Solomon coder to protect against a number of errors. In this work, the authors change the classical Lempel-Ziv encoder in order to organize the inherent remaining redundancy (even if this redundancy is small) in such a way as to achieve error-resilience. Their approach thus requires specific encoders and decoders.

In our approach, we keep the encoder unchanged, and we rather use a joint source channel decoding approach (JSCD) of Lempel-Ziv-77 (LZ) codes by making use only of existing properties. More specifically, no additional information is introduced at the encoding side, the bitstream is exacly the same one as in a classical application. Only the decoder is improved. The proposed receiver is built on the turbo principle, just like the iterative decoding of two serially concatenated codes.

This paper is organized as follows. The next Section describes the Lempel-Ziv-77 entropy coding algorithm. Section 3 presents the Soft-Input Soft-Output (SISO) decoding algorithm based on a modified version of the sequential decoding M-algorithm adapted to the Lempel-Ziv-77 structure codes and to the syntax of the HTML language. Section 4 presents the proposed receiver based on the iterative joint source channel decoding approach. Simulation results are then provided in Section 5. Section 6 concludes our work.

2. LEMPEL-ZIV-77 ENCODING ALGORITHM

Among the large family of Lempel-Ziv encoding algorithms available in the literature, this paper focuses particularly on the Lempel-Ziv-77 algorithm since it is implemented in the HTTP1.1 protocol [1]. LZ is a universal lossless encoding algorithm developed initially to compress texts [4]. The Lempel-Ziv algorithm exploits the redundant nature of the HTML (or XHTML) file corresponding to an ASCII document.

Denote by T the HTML text of length n over a finite alphabet. The i-th symbol in T is denoted by $T[i]$. $T[i,j]$ represents the substring composed by the following symbols $T[i]T[i+1]T[i+2]...T[j]$. The LZ algorithm parses the text sequentially left to right and processes the data on line as it is read in order to adaptively build a dictionary. Suppose that $i-1$ symbols have been parsed providing $h-1$ substrings composing the dictionary denoted by $T[1,i-1] = s_1 s_2 ... s_{h-1}$, where s_k represents the k-th substring. At this step, the algorithm looks in the dictionary (i.e. $T[1,i-1]$) the largest h-th string that matches with the longest string available in $T[i, i+l_h-1]$ with $l_h \leq L$ and L corresponding to the fixed size of the search window.

The retained substring is encoded using a triplet denoted by $< p_i, l_i, c_i >$ where p_i is the pointer toward the dictionary indicating the beginning of the substring to be encoded, l_i the length of the new substring to be included in the dictionary and c_i corresponds to the next symbol $T[i+l_i]$ to be included in the dictionary. Therefore the text to be encoded by LZ is represented by the sequence of triplets as follows: $< p_0, l_0, c_0 >, < p_1, l_1, c_1 >, ..., < p_i, l_i, c_i >, ...$ In the deflate normalization (rfc1951)[2], the size of the dictionary and the

Thanks to French National Research Agency for funding this project part of the DITEMOI Project http://rnrt.ditemoi.free.fr/

search window are respectively fixed to 256 bytes and 32 K-bytes. Consequently the words p_i, l_i and c_i are respectively encoded on 15 bits (i.e. $p_i = p_i^{14}...p_i^j...p_i^0$), 8 bits (i.e. $l_i = l_i^7...l_i^j...l_i^0$) and 8 bits (i.e. $c_i = c_i^7...c_i^j...c_i^0$).

It should be noted that in some options, Huffman encoding algorithm is applied to the triplets of the LZ sequence. In the present work, we do not consider this stage, which is optional. Indeed, its usefulness is moderate, compared to the first step (LZ encoding). For example, a file of 47 K-bytes corresponding to our HTML laboratory page, is compressed by a factor of 7 by LZ, while the Huffman encoding provides a further improvement of about 30 % of the file size. In any case, the asymptotic properties of LZ are obtained without applying Huffman encoding: the improvement made by Huffman should logically decrease with the size of the file.

3. SEQUENTIAL DECODING M-ALGORITHM USING THE LEMPEL-ZIV-77 STRUCTURE CODES AND THE HTML SYNTAX

This Section concentrates on the Soft-Input Soft-Output (SISO) inner decoder, denoted SDLZ, of the proposed iterative receiver. The SISO decoder processes in two main steps. The first one estimates the LZ sequence using a modified version of the traditional sequential decoding M-algorithm and the second one derives the soft values which will be discussed in the next Section. Let briefly introduce at the following the traditional sequential decoding M-algorithm.

3.1. Traditional sequential decoding M-algorithm

The traditional sequential decoding M-algorithm, processes on a tree structure using at each depth of the tree the best M memorized paths. This algorithm belongs to the breadth first approaches [5]. At each iteration, the best M memorized paths at the same depth in the tree are extended one step forward. For each extended path, the cumulated metric (maximum likelihood criterion) is computed. Among these paths, only M best paths are selected in accordance with the cumulated metric. When the algorithm reaches the maximum depth, the best stored path in terms of the cumulated metric is considered as the best sequence estimate.

Denote by $\mathbf{x} = (x_1, ..., x_t, ..., x_N)$ the sequence to be transmitted over an additive white gaussian noise (AWGN) channel and $\mathbf{y} = (y_1, ..., y_t, ..., y_N)$ the sequence of the received symbols. The corresponding transmission scheme is depicted in Fig.1. The transmitted sequence is estimated according to the maximum likelihood (ML) criterion as follows: $\widehat{\mathbf{x}} = \arg\max_x \log(P(\mathbf{y}/\mathbf{x}))$ where $P(\mathbf{y}/\mathbf{x})$ is the maximum likelihood. The channel being memoryless, the maximum likelihood becomes:

$$P(\mathbf{y}/\mathbf{x}) = \prod_{k=1}^{N} P(y_k/x_k) = \prod_{k=1}^{N} \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{(y_k - x_k)^2}{2\sigma^2}) \quad (1)$$

where σ^2 represents the variance of the zero-mean Gaussian white noise.

The estimated sequence is thus given by: $\widehat{\mathbf{x}} = \arg\min_x \sum_{k=1}^{N} (x_k - y_k)^2$ where $\mu(N) = \sum_{k=1}^{N} (x_k - y_k)^2$ corresponds to the cumulated metric allowing to find the best path. The traditional sequential decoding M-algorithm is summarized below:

1. Read the received information y_k,
2. Extend all M best current nodes of the tree associated to this information y_k,
3. Compute the metrics of the extended branches,
4. Sort the $2M$ new nodes according to the cumulated metrics computed in 3,
5. Select among the $2M$ paths, the best M paths according to the cumulated metrics of each path,
6. Start again from 1 until reaching the maximal depth of the tree (i.e. reading the last information y_N).

3.2. Modified version of the sequential decoding M-algorithm

The traditional sequential decoding M-algorithm described above is generic, and does not take into account any property of the decoded sequence should meet. However, sequential algorithms easily allow to restrict the sequences to those meeting a given syntax, or set of rules. This restriction allows to reduce the computational complexity and to improve the decoding efficiency. In fact, it is possible to impose a set of rules to be satisfied during the decoding step of the received sequence, thus imposing the decoded sequence to be a *valid* sequence, both in terms of LZ and HTML code. Altogether, the required rules should describe the structure of the LZ codes combined to the syntax of the HTML language.

In order to benefit from the LZ code structure, we modify the traditional sequential decoding M-algorithm as follows. The proposed M-algorithm (SDLZ) takes a decision only after having read all the elements corresponding to one word such as the pointer (i.e. $p_i = p_i^{14}...p_i^j...p_i^0$), the length (i.e. $l_i = l_i^7...l_i^j...l_i^0$) or the character ($c_i = c_i^7...c_i^j...c_i^0$). In addition to the metric criterion, we check if the LZ grammatical rules are satisfied. If it is not the case, the corresponding branches are then dropped from the tree. The proposed rules are listed below: i) the decoded pointer does not exceed the number of symbols already read and inserted in the dictionary,ii) a null decoded pointer involves also a null decoded length, iii) the decoded character must belong to the specific range of the ASCII codes.

In addition, among the selected decoded triplets, the modified M-algorithm parses the current substring and checks the syntax of the HTML document. Indeed, an HTML document is a special document which uses rules of SGML (Standardized General Markup Language) [1]. The structure of any HTML document is then described by inserting tags into the text to be displayed by the browser. Depending on the HTML version, some HTML elements impose to start with a tag and to end with a tag to define the same HTML element. Moreover, specific attributes are associated to the tags according to the HTML version [2]. These various HTML rules are exploited to parse the validity of the HTML document after having decoded one or several triplets.

Any sequence which do not fulfill the LZ of html rules is discarded, and the corresponding metric is not even computed. Only the M best *valid* paths are kept.

Summarize the different steps of the sequential decoding M-algorithm adapted to the LZ codes and the syntax of the HTML language as follows:

1. Read the received information y_k,
2. For each received information, extend the M current branches of the tree until building $2^\alpha M$ paths (with $\alpha = 15$ for the pointer word, $\alpha = 8$ for the character word or length word),
3. Among the $2^\alpha M$ paths, drop the branches which do not satisfy the LZ rules and the HTML syntax. The number of branches selected is denoted by M',
4. Compute the metrics of the remaining branches,
5. Sort these branches according to the cumulated metric,

[1] http://www.w3.org/markup/sgml/

[2] Html specification, w3c recommendation

6. Among the M' retained paths in 5, select the M best paths,
7. Start again from 1 until attaining the final depth of the tree.

4. ITERATIVE JOINT SOURCE CHANNEL DECODING METHOD

The communication system developed in this paper is depicted by Fig.2. Based on this transmission model, the receiver task consists in estimating the sequence of the transmitted bits from the sequence of the received symbols.

4.1. System model and notations

Denote $\mathbf{d} = (d_1, ..., d_k, ..., d_N)$ the binary sequence generated by the LZ algorithm ($< p_i, l_i, c_i >$) where N is the length of the sequence and d_k is the k-th information bit. The information bits $\{d_k\}$ are assumed to be uniformly distributed. This LZ binary sequence is first permuted by a pseudo-random interleaver to break the error burst at the receiver, then used as an input to the convolutional encoder with a code rate equal to $\frac{1}{n}$. The convolutional encoder outputs are then modulated by a BPSK modulator and transmitted over memoryless AWGN channel. The transmitted data stream is denoted by $\mathbf{x}_1^N = (\mathbf{x}_1, ..., \mathbf{x}_k, ..., \mathbf{x}_N)$ where $\mathbf{x}_k = (x_{k,0}, x_{k,1}, ..., x_{k,n-1})$. The received data stream $\mathbf{y}$ is denoted by $\mathbf{y}_1^N = (\mathbf{y}_1, ..., \mathbf{y}_k, ..., \mathbf{y}_N)$ where $\mathbf{y}_k = (y_{k,0}, y_{k,1}, ..., y_{k,n-1})$.

Define two sets $\boldsymbol{R}$ and $\boldsymbol{D}$ of binary sequences $\mathbf{d}$ of length N. The first set $\boldsymbol{R} = \{[d_1, d_2, ..., d_N] \in \{0,1\}^N\}$ contains all possibles sequences. While the second one $\boldsymbol{D} = \{[d_1, d_2, ..., d_N] \in \{0,1\}^N\}$ contains only the sequences satisfying the LZ structure codes and the syntax of the HTML language.

4.2. Proposed iterative receiver

The receiver task consists in estimating the LZ sequence of the transmitted bits from the sequence of the received symbols. We propose to solve this problem using an iterative approach based on the turbo principle with serial concatenated codes as developed in the reference [6]. The first code corresponds to the channel outer decoder (i.e. SISO BCJR) and the second block concerns the source inner decoder (i.e. SISO SDLZ) described above (see Fig.2). Each decoding block is fed with soft inputs and can deliver soft outputs. This soft information is exchanged, in an iterative process, between the channel decoder and the source decoder. Classically, this soft information is the so-called *extrinsic probability* of each bit (i.e. the *a posteriori* probability of the bit, divided by its *a priori* probability).

The iterative receiver estimates the transmitted message $\mathbf{b}$ so as to iteratively optimize the maximum a posteriori (MAP) criterion within each block, and considering the extrinsic information provided by the other block as an *a priori* probability. Thus, at a given iteration I, the decoding algorithm proceeds in two main steps as described below.

4.2.1. Outer BCJR channel decoder

In the first step, the BCJR channel decoder computes the a posteriori probability $P_R(\mathbf{d}/\mathbf{y})$ for for every $\mathbf{d} \in \mathbf{R}$ (all possible sequences) which is also equivalent to the product of the marginals a posteriori probabilities (APP) $\prod_{k=1}^{N} P^I_{BCJR}(d_k/y_k)$ since the channel transmission is assumed to be memoryless and the $\{d_k\}$ are independent.

At any iteration I, the output of the outer BCJR channel decoder is given by the extrinsic probability associated to the coded information bit as follows:
$E^I_{BCJR}(d_k) = K_{BCJR} \frac{P^I_{BCJR}(d_k/y_k)}{P^I_{BCJR}(d_k)}$ where K_{BCJR} is the normalization factor such as $E^I_{BCJR}(d_k = 0) + E^I_{BCJR}(d_k = 1) = 1$; $P^I_{BCJR}(d_k)$ is the a priori probability associated to the coded information bit corresponding at the interleaved extrinsic information ($E^{I-1}_{DSLZ}(d_k)$) computed by the SDLZ decoder at the previous iteration (i.e. $I-1$). The de-interleaved extrinsic information $E^I_{BCJR}(d_k)$ is then sent as an input of the SDLZ decoder.

4.2.2. Inner SDLZ source decoder

The second step, related to the inner source decoder SDLZ, consists in a projection of the distribution of the APP evaluated on $\boldsymbol{R}$, on the set of the APP distribution compatibles with the LZ structure codes and the syntax of the HTML language. The distribution of the required APP, denoted by $P^+_{SDLZ}(\mathbf{d})$, is that which minimizes the Kullback-Leibler distance given by:

$$P^+_{SDLZ}(\mathbf{d}) = \arg \min_{P_{\boldsymbol{D}}(\mathbf{d})} dist(P_{\boldsymbol{D}}(\mathbf{d}), P_{\boldsymbol{R}}(\mathbf{d}/\mathbf{y})) \quad (2)$$

It has been shown in [6] that the required APP distribution is given by :

$$P^+_{SDLZ}(\mathbf{d}) = \begin{cases} \frac{P_{\boldsymbol{R}}(\mathbf{d}/\mathbf{y})}{\sum_{d \in \mathbf{D}} P_{\boldsymbol{R}}(\mathbf{d}/\mathbf{y})} & \text{if } \mathbf{d} \in \mathbf{D} \\ 0 & \text{if } \mathbf{d} \in \mathbf{R} \backslash \mathbf{D} \end{cases} \quad (3)$$

At any iteration I, the output of the inner source decoder (SDLZ) is given by the extrinsic probability associated to the coded information bit as follows: $E^I_{SDLZ}(d_k) = K_{SDLZ} \frac{P^I_{SDLZ}(d_k/y_k)}{P^I_{SDLZ}(d_k)}$ where K_{SDLZ} is the normalization factor such as $E^I_{SDLZ}(d_k = 0) + E^I_{SDLZ}(d_k = 1) = 1$; $P^I_{SDLZ}(d_k)$ is the a priori probability associated to the coded information bit corresponding at the interleaved extrinsic information ($E^{I-1}_{BCJR}(d_k)$) computed by the BCJR decoder at the previous iteration (i.e. $I-1$). The marginal APP is deduced from equation (2) as follows: $P^I_{SDLZ}(d_k/y_k) = \sum_{d_k:0,1} P^+_{SDLZ}(\mathbf{d})$
The de-interleaved extrinsic information $E^I_{SDLZ}(d_k)$ is then sent as an input of the BCJR decoder.

The SDLZ decoder focuses on the distribution of the APP that maximize $P^+_{SDLZ}(\mathbf{d})$ which is equivalent to preserve the optimal solutions with respect to the complete sequences and the individual bits as follows: $\widehat{d_k} = \arg \max_{d_k:0,1} \log(\sum_{\mathbf{d} \in \mathbf{D}} P^+_{SDLZ}(\mathbf{d}))$
Most of the required work is performed by a slight change in the modified M-algorithm described in Section 3.2 : the probabilities computed from the channel outputs are replaced by the extrinsic values provided by the other blocks. As an output, this version provides the set of M best paths which are compatible with the LZ and HTML syntax. The only additional computation is to marginalize the corresponding probabilities on these M paths rather than on all feasible sequences. Since (hopefully) these M paths are the likeliest, this a rather valid approximation.

5. SIMULATION RESULTS

The simulations are carried out on two HTML files. The first HTML file [3] is a small file of size 413 bytes, which contains 60.5 % of tags compared to the total number of characters in the HTML file. The second HTML file [4], corresponds to a medium file of size 3031 bytes, containing only 11.5 % of tags compared to the total number of characters in the HTML file. Obviously, since we aim at transmitting files to mobiles with small screens), the file is likely to be not too large in practical situations. Note also that, due to the nature of the files, the percentage of tags increases when the file size decreases. This impacts the efficiency of our method.

[3] samples of sdk database www.forum.nokia.com/tools, worldcup.html
[4] ftp://ftp.uu.net/graphics/png/documents/zlib/zdoc-index.html

The performance of the proposed receiver is measured in terms of symbol error ratio (SER) for various (E_b/N_0). The first simulation results are performed on the transmission chain without channel coding (see Fig.1) in order to show the importance of including the LZ structure codes and the HTML syntax in the traditional sequential decoding M-algorithm. The symbol error ratio (SER) performances are compared for the sequential decoding M-algorithm for different values of M when (i) only the LZ code structure is imposed to the decoded data stream, and (ii) the LZ structure codes and the syntax of the HTML language are imposed to the decoded data stream. The graphs of Fig.3 show that the SER performance of the modified algorithm is improved when using the structure of the LZ codes compared to a hard decoding decision.

The communication system presented by Fig.2 is used to evaluate the performance of the proposed iterative receiver. The simulations are carried out in an environment reminiscent of the IEEE 802.11.a standard using the first operating mode [5], which is intended to be used around 6dB. Therefore, the specified convolutional encoder has the following characteristics: a constraint length equal to 7, with a polynomial generator presented by its octal representation [171,133] and a code rate equal to 1/2. The graphs illustrated by Fig. 4, concerns the simulation results obtained on the two HTML files with $M = 10$ after three iterations. These results are compared to a hard decoding decision. For the first HTML file (Top) and for a given E_b/N_0 equal to $6dB$ (i.e. in the operating zone defined in the IEEE 802.a), with two iterations, the proposed iterative receiver reduces the SER by a factor 10 compared to the results achieved without the SDLZ decoder using only a hard decision at the output of the BCJR channel decoder. In addition, it is also clearly seen that more errors can be corrected when the HTML file contains more tags.

6. CONCLUSION

This paper proposes an iterative joint source channel decoding approach allowing the error correction of corrupted LZ compressed HTML pages during their transmission via a noisy mobile channel. The proposed receiver is based on the turbo principle, in a setting similar to the one used for decoding serially concatenated codes. The source decoder is based on a modified version of the traditional sequential decoding M-algorithm. This new algorithm exploits the structure of the LZ codes combined to the syntax of the HTML language. The provided simulation results, in a mobile environment, show that the proposed receiver improves the SER performances without introducing additional redundant information in the transmitted bitstream. We are currently working on the joint source and channel decoding of other variants of compression standards, involving Huffman codes on top of universal lossless codes.

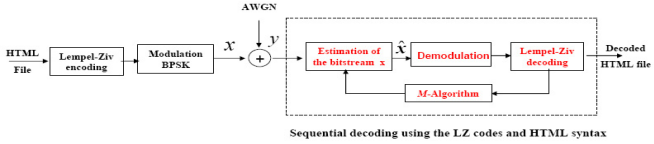

Fig. 1. Sequential decoding M-algorithm adapted to the LZ codes and the syntax of the HTML language (SDLZ)

7. REFERENCES

[1] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T.Berners-Lee, "Hypertext transfer protocol–http/1.," June 1999.

[5] http://www.ieee802.org/11/

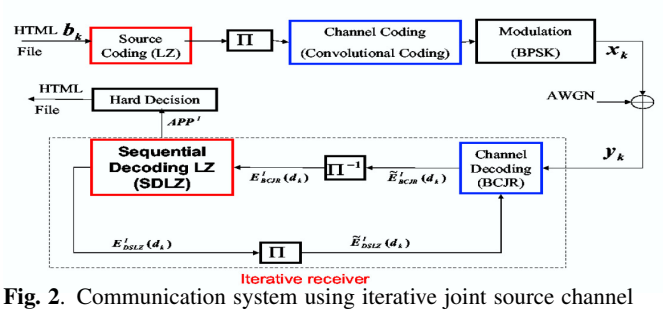

Fig. 2. Communication system using iterative joint source channel decoding

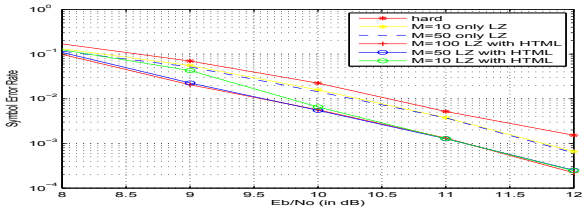

Fig. 3. SER performances of the SDLZ algorithm applied to the HTML file 1

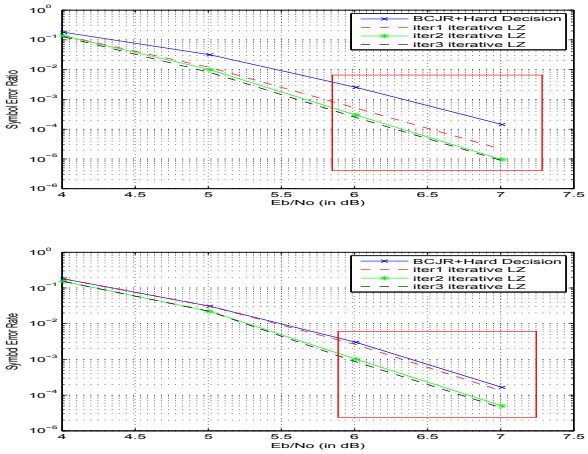

Fig. 4. SER performances of the iterative receiver applied to the HTML file tests: HTML file 1 (top) and HTML file 2 (bottom)

[2] L.P. Deutsch, "ZLIB Compressed data format specification," in *rfc1950*, May 1996.

[3] S. Lonardi, W. Szpankowski, and M. D. Ward, "Error resilient lz'77 data compression: Algorithms, analysis, and experiments," in *IEEE Transactions on Information Theory 53(5)*, May 2007, pp. 1799–1813.

[4] J. Ziv and A. Lempel, "A universal algorithm for sequential data compression," in *IEEE Transactions on Information Theory, 23(3)*, May 1977, pp. 337–343.

[5] J.B. Anderson and S. Mohan, "Source and channel coding: an algorithmic approach," in *Kluwer Academic Publishers, Norwell*, MA 1991.

[6] P. Magniez, B. Muquet, P. Duhamel, V. Buzenac, and M. de-Courville, "Optimal decoding of bit-interleaved modulations: Theoretical aspects and practical algorithms," in *2nd Intl. Symposium on Turbo Codes and Related Topics*, Sept. 2000, pp. 284–287.