Proc. IEEE Global Telecommun. Conf. (Dallas, TX), 172-178, Nov. 2004

Distance Measurement Method For Double Binary Turbo Codes and A New Interleaver Design For DVB-RCS

Youssouf Ould-Cheikh-Mouhamedou
Electrical and Computer Engineering
McGill University
Montreal, Quebec, Canada, H3A 2A7

Stewart Crozier
Communications Research Centre
3701 Carling Ave.
Ottawa, Ontario, Canada, K2H 8S2

Peter Kabal
Electrical and Computer Engineering
McGill University
Montreal, Quebec, Canada, H3A 2A7

***Abstract*— This paper presents a computationally efficient distance measurement method for double binary turbo codes, such as these used in the Digital Video Broadcast with Return Channel via Satellite (DVB-RCS) standard, based on Garello's method. Distance spectra for all standardized DVB-RCS packet sizes and all standardized code rates are presented. A new interleaver design for DVB-RCS based on the dithered relative prime (DRP) interleaving approach is also presented. A minimum distance ($d_{\min}$) of 36 has been achieved for an unpunctured ATM packet of 424 information bits with a DRP interleaver, whereas the $d_{\min}$ of the standardized DVB-RCS interleaver is 31. A $d_{\min}$ of 38 has been achieved for an unpunctured MPEG packet of size 1504 information bits with a DRP interleaver, whereas the $d_{\min}$ of the standardized DVB-RCS interleaver is 33. Simulation results for code rate 1/3 show an improvement at high signal to noise ratios of at least 0.15 dB and 0.25 dB for ATM and MPEG packets, respectively.**

I. Introduction

Consider the transmission of a linear binary code $C(N, \widetilde{K})$ (N is the codeword length, $\widetilde{K}$ is the number of information bits) over the additive white gaussian noise (AWGN) channel using binary phase-shift keying (BPSK) or quadrature phase-shift keying (QPSK). Applying Maximum-Likelihood (ML) decoding, the Frame Error Rate (FER) and Bit Error Rate (BER) are upper bounded by the union bounds [1]:

$$\text{FER} \leq \frac{1}{2} \sum_{d \geq d_{\min}} A_{\text{d}} \operatorname{erfc}\left(\sqrt{d \frac{\widetilde{K}}{N} \frac{E_{\text{b}}}{N_0}}\right) \tag{1}$$

$$\text{BER} \leq \frac{1}{2} \sum_{d \geq d_{\min}} \frac{W_{\text{d}}}{\widetilde{K}} \operatorname{erfc}\left(\sqrt{d \frac{\widetilde{K}}{N} \frac{E_{\text{b}}}{N_0}}\right) \tag{2}$$

Here, $d_{\min}$ is the minimum distance of the code, the multiplicity A_{d} is the number of codewords with Hamming weight d, the information bit multiplicity W_{d} is the sum of the Hamming weights of the A_{d} input sequences generating the codewords with Hamming weight d, $\operatorname{erfc}(x)$ is the complementary error function, E_{b} is the energy per information bit and N_0 is the one-sided noise power spectral density. At high signal to noise ratio (SNR) (i.e., low error rates), the FER and BER can be approximated by the first term or first few terms of equations (1) and (2). However, it is important to keep in mind that turbo codes [2] use iterative soft decoding [3], which is sub-optimal compared to Maximum-Likelihood (ML) decoding. See [4] for examples and further discussion.

The FER and BER can be obtained by software simulation. However, for applications with very low error rates (e.g., FER$<10^{-8}$), reliable software simulation could take months or may not be practical at all. An alternative, at least for low error rates, is to use the analytical approach described above. In order to use this approach, it is necessary to have a distance measurement method. It is important that such a method allows the computation of $d_{\min}$ in a reasonable time.

This paper is structured as follows. Past works on distance measurement methods are reviewed in the second section. A method for computing a lower bound on $d_{\min}$ is given in the third section. The fourth section presents a recursive method to compute the true $d_{\min}$. The fifth section presents some useful techniques to lower the computational complexity. Distance results of the standardized Digital Video Broadcasting with Return Channel via Satellite (DVB-RCS) interleavers and the new DRP interleavers are discussed in the sixth section and conclusions are given in the seventh section.

II. Background

According to [5], the minimum distance ($d_{\min}$) of a turbo code is expected to be equal to $d_{\min}(2)$, the distance due to input sequences of weight 2, when the interleaver is random and its size tends to infinity. This approach reduces the number of non-zero input sequences to be tested from $(2^{\widetilde{K}} - 1)$ to $\binom{\widetilde{K}}{2}$, which leads to a significant reduction in computational complexity. Unfortunately, this approach does not apply if the interleaver size is small to medium (for example, 1000 bits). Even for very large random interleavers, $d_{\min}$ can still be produced by an input sequence of any weight. The necessity of considering input-weights larger than two has been confirmed analytically in [6]. Thus $d_{\min}(2)$ gives a loose upper bound on $d_{\min}$. Furthermore, from a practical point of view, it is easy to increase $d_{\min}(2)$ by designing interleavers with high

[0]This work was supported by the Communication Research Centre (CRC) in Ottawa, Canada.

spread [7], [8], [9]. In this case $d_{\min}(2)$ tends to be a very loose upper bound on $d_{\min}$. This loose upper bound can still be used in practice to first discard bad interleavers (low $d_{\min}(2)$) and the remaining interleavers can then be tested by a true distance measurement method.

A number of brute force approaches computing $d_{\min}$ have been proposed. The basic idea is to consider all possible candidate sequences for one of the constituent encoders, then interleave and encode each sequence with the second constituent encoder to find the total turbo code distance. These brute force approaches are practical, even for large block sizes, as long as $d_{\min}$ remains small (i.e., it is close to the minimum possible $d_{\min}$). For well designed interleavers, however, the computational complexity quickly becomes unacceptable as the value of $d_{\min}$ increases. See [4] and [10] and the references therein for examples and further discussion.

Another method based on combining low input-weight patterns that lead to low-weight codewords has been presented in [8]. An essential aspect of this approach is to determine which combinations of low input-weight patterns should be considered. It has been observed that these combinations depend on the *spread*, defined as

$$S_p = \min_{(i,j \neq i)} (\mid \pi(i) - \pi(j) \mid + \mid i - j \mid), \forall i, j \in \{0, \ldots, \widetilde{K} - 1\}, \tag{3}$$

where π represents an interleaver permutation. A high spread constraint easily eliminates many of the worst input-weight combinations. However, some input-weight combinations do not improve with spread. Improving the distance for these cases requires specific distance tests to be performed. Fortunately, many of the remaining worst-case input-weight combinations are fairly easy to test with reasonable computational complexity. See [8] for the recommended cases to test and further details. This method has been found to give a fairly tight upper bound for the true $d_{\min}$ and is thus very useful in designing good interleavers in a reasonable time. The approach is very efficient, demonstrated by the fact that it was possible to find a distance upper bound of 110 for a 16-state single binary turbo code with $\widetilde{K}$=32768 bits. Also, for large interleavers with sufficiently high spread, the upper bounds are guaranteed to be the true minimum distances for all cases up to and including an input weight of 6. Unfortunately, the higher the upper bound that is achieved, the less likely the bound is to be tight, that is, the more likely the true $d_{\min}$ will be caused by one of the cases not tested.

Recently, a fast method based on the ability of a soft-in decoder to overcome error impulse inputs has been presented in [11]. This method gives the true $d_{\min}$ if ML decoding is used. A short description of this method follows. Define $\boldsymbol{x} = (-1, -1, \ldots, -1)$ as the modulated codeword generated by the *all zero* input sequence and $\boldsymbol{y} = (-1, -1, \ldots, -1, -1 + E_i, -1 \ldots, -1)$ as the input to the decoder (E_i is called the error impulse at position i and is a real number). Assuming that $d_{\min}$ lies in the interval $[d_0, d_1]$, where d_0 and d_1 are two integers and a ML decoder is used, then $d_{\min}$ can be determined with the following algorithm, where $\widetilde{K}$ is the number of information bits:

‖ set $E_{\min} = d_1 + 0.5$;
‖ for $i = 0$ to $(N - 1)$ do
- $E = d_0 - 0.5$;
- set $[(\hat{\boldsymbol{x}} = \boldsymbol{x}) = \text{TRUE}]$;
- while $[(\hat{\boldsymbol{x}} = \boldsymbol{x}) = \text{TRUE}]$ and $(E \leq E_{\min} - 1.0)$ do
  - $E = E + 1.0$;
  - $\boldsymbol{y} = (-1, \ldots, -1, -1 + E, -1 \ldots, -1)$; where $(-1 + E)$ is in position i;
  - ML decoding of $\boldsymbol{y} \Rightarrow \hat{\boldsymbol{x}}$;
  - If $(\hat{\boldsymbol{x}} \neq \boldsymbol{x})$ then $[(\hat{\boldsymbol{x}} = \boldsymbol{x}) = \text{FALSE}]$;

  end while
- $E_{\min} = E$

end for
‖ $d_{\min}$ is the integer part of E_{min}

Unfortunately, for non-ML iterative soft decoding the relationship between the distance obtained with this method and the true $d_{\min}$ remains uncertain. It has been observed that this method usually gives a lower bound on $d_{\min}$, but distances higher than $d_{\min}$ have also been found. Even so, the approach may prove to be very useful for finding good interleavers.

A novel and efficient method to compute the true $d_{\min}$, the true multiplicity $A_{d_{\min}}$ and the true information bit multiplicity $W_{d_{\min}}$ based on the notion of constrained subcodes has been presented by Garello [10] for single binary turbo codes. This method assumes that both encoders start in the zero state and are forced to the zero state at the end of the encoding stage by adding termination bits, which are then sent to the decoder. This results in a reduction in code rate (especially noticeable for short block lengths) and the minimum distance obtained is usually less than the $d_{\min}(\text{TB})$ obtained with tail-biting if a structured interleaver is used.

This paper extends Garello's distance measurement method for single binary turbo codes to double binary turbo codes, such as the one used in the DVB-RCS [12] standard. Some techniques used in Garello's distance measurement routine are explained. Furthermore, a new effective early stopping technique that reduces the computational complexity by a factor of 2 is presented. This reduction is significant, noting that the computation of high $d_{\min}$ may take a month or more. The new distance measurement method is applied to DVB-RCS turbo codes, which use tail-biting [13]. Distance spectra including multiplicity and information bit multiplicity are presented for all standardized interleavers and code rates of DVB-RCS turbo codes. A new interleaver design for DVB-RCS based on dithered relative prime (DRP) interleavers is also presented. The distance spectra, FER and BER for the DVB-RCS standard are compared to those for the new DRP interleavers.

III. Computing a lower bound on minimum distance

This lower bound is a key element in lowering the computational complexity and will be used in the next section to determine the exact $d_{\min}$ in a recursive manner.

DVB-RCS turbo codes use tail-biting, where the first encoder (ENC1) must start and stop in the same state. The second encoder (ENC2) must also start and stop in the same state. Any combination of starting states for the two encoders is allowed. Thus, for an 8-state turbo code there are 64 possible starting state combinations. In double binary turbo codes each symbol consists of 2 bits, thus $K = \frac{\tilde{K}}{2}$. Define $\boldsymbol{u} = (u_0, \ldots, u_j, \ldots, u_{K-1})$ and $\boldsymbol{u}_\pi = (u_{\pi(0)}, \ldots, u_{\pi(j)}, \ldots, u_{\pi(K-1)})$ as the input sequences into ENC1 and ENC2 respectively, where π is an interleaver of length K. In other words, any information symbol u_j entering ENC1 at time j will enter ENC2 at time $\pi^{-1}(j)$. Turbo codes are linear codes, thus the minimum distance ($d_{\min}$) is given by the codeword with minimum Hamming weight, where Hamming weight is the number of non-zero bits in a binary sequence. There are $(4^K - 1)$ possible non-zero input sequences that may generate $d_{\min}$. The goal here is to find *all* input sequences from this set that generate $d_{\min}$. Define $\overline{w}(\boldsymbol{u}) = \overline{WE1} + \overline{WE2}$ as the Hamming weight of the codeword generated by the input sequence $\boldsymbol{u}$, where $\overline{WE1}$ is the sum of Hamming weights of $\boldsymbol{u}$ and its corresponding parities generated by ENC1 and $\overline{WE2}$ is the Hamming weight of parities generated by ENC2.

Assume an input sequence $\boldsymbol{u}^j$, where only the first j information symbols $\boldsymbol{u}^{<j} = (u_0, \ldots, u_{j-1})$ are known and the other $(K-j)$ information symbols $\boldsymbol{u}^{\geq j} = (u_j, \ldots, u_{K-1}) = (\times, \ldots, \times)$ are unknown ($\times$ can be 00, 01, 10 or 11). The aim is to find the unknown information symbols $\boldsymbol{u}^{\geq j}$ in $\boldsymbol{u}^j$ that minimize the weight $\overline{W}^j = \overline{WE1}^j + \overline{WE2}^j$. Define $\overline{MWE1}^j$ and $\overline{MWE2}^j$ as the minimum weight-outputs generated respectively by the input sequences $\boldsymbol{u}^j$ into ENC1 and $\boldsymbol{u}^j_\pi$ into ENC2. Note that the input symbols at positions $(j, \ldots, K-1)$ in $\boldsymbol{u}^j$ are not necessarily the same input symbols at positions $(\pi^{-1}(j), \ldots, \pi^{-1}(K-1))$ in $\boldsymbol{u}^j_\pi$, because $\overline{MWE1}^j$ and $\overline{MWE2}^j$ are computed separately based on the common knowledge of the known symbols $\boldsymbol{u}^{<j}$. If the symbols in $\boldsymbol{u}^{<j}$ agree with the first j information symbols of any input sequence $\boldsymbol{u}_{\min}$ that generates $d_{\min}$, then $\overline{MW}^j = \overline{MWE1}^j + \overline{MWE2}^j$ is a lower bound for $d_{\min}$, i.e.,

$$\overline{MW}^j \leq (\overline{w}(\boldsymbol{u}_{\min}) = d_{\min}). \tag{4}$$

A. The computation of $\overline{MWE2}^j_{tb}$

The computation of $\overline{MWE2}^j_{tb}$ for tail-biting turbo codes is obtained by applying the following modified *forward* Viterbi algorithm (MVA) [10]. Each branch of the trellis of ENC2 is labelled with the corresponding weight except the irrelevant branches at sections $(\pi^{-1}(0), \ldots, \pi^{-1}(j-1))$ of ENC2 that are labelled with an effectively infinite value (in practice, it is enough to set this value to N). The irrelevant branches are the branches associated with symbols different from the known symbols at trellis sections $(\pi^{-1}(0), \ldots, \pi^{-1}(j-1))$. Furthermore assume that the encoder starts and ends in the state s_x.

1) Initialize $t = 0$; $w(s_x) = 0$; $w(s \neq s_x) = \infty$.
2) Increase t by 1
   - Compute all weights of each state s by adding the weight of the branch entering s from state s' and the weight of the state s' at time $(t-1)$, then set $w(s)$ to the smallest weight. Repeat until $t = K$.
3) The value $\overline{MWE2}^j_{tb}$ is $w(s_x)$.

B. The computation of $\overline{MWE1}^j_{tb}$

The computation of $\overline{MWE1}^j_{tb}$ for tail-biting turbo codes consists of three parts:

- The output-weight ENC1($\boldsymbol{u}^{<j}$) resulting from encoding the known input sequence $\boldsymbol{u}^{<j}$ with ENC1. ENC1 starts encoding $\boldsymbol{u}^{<j}$ at time $t = 0$ in state s_x and ends it at time $t = j$ in state s_y. The computation of ENC1($\boldsymbol{u}^{<j}$) and s_y is straightforward.
- The minimum output-weight ENC1($\boldsymbol{u}^{\geq j}$) resulting from encoding the unknown input sequence $\boldsymbol{u}^{\geq j}$ with ENC1, which starts encoding $\boldsymbol{u}^{\geq j}$ at time $t = j$ in the state s_y and ends it in the state s_x at time $t = K$. Applying the *backward* MVA ($t = K$; initial state=s_x; $w(s_x) = 0$; $w(s \neq s_x) = \infty$; decreasing by 1 until $t = j$) gives a minimum weight at time $t = j$ for the state s_y. This weight is the needed minimum output-weight ENC1($\boldsymbol{u}^{\geq j}$).
- The $\overline{MWE1}^j_{tb}$ is the sum of ENC1($\boldsymbol{u}^{<j}$) and ENC1($\boldsymbol{u}^{\geq j}$).

IV. Recursive construction of minimum distance

The search for $d_{\min}$ consists of a recursive construction of input sequences that generate codewords of weight $d_{\min}$. Assume d^* is an upper bound for $d_{\min}$ and IW^* is the maximum allowed input-weight that can generate $d_{\min}$. Any input sequence $\boldsymbol{u} = \boldsymbol{u}^j$ fulfilling the criteria (weight of $\boldsymbol{u}^j$) $\leq IW^*$ and $\overline{MW}^j \leq d^*$ may generate $d_{\min}$ and will be the basis for the next iteration $\boldsymbol{u}^{j+1}_{a\in\phi^0}$, where $\phi^0 = \{00, 01, 10, 11\}$ and $\boldsymbol{u}^{j+1}_{a\in\phi^0} = (\boldsymbol{u}^{<j}, a, \times, \ldots, \times) = (\boldsymbol{u}^{<j}, 00/01/10/11, \times, \ldots, \times)$. If more than a single $\boldsymbol{u}^{j+1}_i$ fulfill the criteria, then set $\boldsymbol{u} = \boldsymbol{u}^{j+1}_i$ as the current basis for the next iteration $\boldsymbol{u}^{j+2}_{a\in\phi^0} = (\boldsymbol{u}^{<j+1}, a, \times, \ldots, \times)$ and keep the other sequences $\boldsymbol{u}^{j+1}_{a\neq i}$ that fulfill the criteria to be used as a bases for further iterations later. If only a single $\boldsymbol{u}^{j+1}_i$ fulfills the criteria, then set $\boldsymbol{u} = \boldsymbol{u}^{j+1}_i$ as the current basis for the next iteration $\boldsymbol{u}^{j+2}_{a\in\phi^0} = (\boldsymbol{u}^{<j+1}, a, \times, \ldots, \times)$. The iterations continue until $\boldsymbol{u}^{K}_{a\in\phi^0} = (\boldsymbol{u}^{<K-1}, a)$. If the $\overline{MW}^K$ resulting from $\boldsymbol{u}^K_{00}$, $\boldsymbol{u}^K_{01}$, $\boldsymbol{u}^K_{10}$ or $\boldsymbol{u}^K_{11}$ is lower than d^*, then set d^* to $\overline{MW}^K$. This leads to a reduction in the number of bases to be considered in the next iterations and thus fewer input sequences must be tested, leading to lower computational complexity. To make sure that all possible $(4^K - 1)$ non-zero input sequences are tested, the input sequences $\boldsymbol{u}^1 = (\otimes, \times, \ldots, \times) = (01/10/11, \times, \ldots, \times)$, $\boldsymbol{u}^2 = (00, \otimes, \times, \ldots, \times)$, $\boldsymbol{u}^3 = (00, 00, \otimes, \times, \ldots, \times)$,..., $\boldsymbol{u}^{K-1} = (00, \ldots, 00, \otimes, \times)$ must be used as bases for the next iterations, where $\otimes \in \phi = \{01, 10, 11\}$. The unique offsets corresponding to $\boldsymbol{u}^{<1} = (\otimes)$, $\boldsymbol{u}^{<2} = (00, \otimes)$, $\boldsymbol{u}^{<3} =$

$(00,00,\otimes)$,…,$\boldsymbol{u}^{<K-1} = (00,\ldots,00,\otimes)$ must be used to guarantee the exact values for distance d, multiplicity A_d and information bit multiplicity W_d.

To find the exact distance spectrum for tail-biting turbo codes the algorithm should be run $(\delta_1 \cdot \delta_2)$ times, where δ_1 and δ_2 are the number of states of ENC1 and ENC2, respectively. As expected, it has been observed for tail-biting that most of the computation is used in finding $d_{\min}$ for the case where both encoders are assumed to start in state zero, because the number of surviving paths to be tested is significantly higher than for other starting state combinations.

V. Techniques to reduce the computational complexity

Some of Garello's techniques that reduce the computational complexity are explained. A new efficient early stopping rule that significantly reduces the computational complexity is also presented.

A. Garello's modified definition for $\overline{WE1}$ and $\overline{WE2}$

Garello's modified definition for $\overline{WE1}$ and $\overline{WE2}$ are explained. The computational complexity of the distance measurement method presented here depends strongly on the number of bases that must be considered for later testing. This number in turn depends on the value of $\overline{MW}^j$ for each basis $\boldsymbol{u}^j, j \in \{1,\ldots,K-1\}$. The aim now is to increase the value of $\overline{MW}^j$, because the higher $\overline{MW}^j$ is, the lower the number of bases that must be considered and thus fewer input sequences must be tested.

Note that any recursive convolutional double binary code of M delays can be driven from any state s_x to any state s_y by an input sequence of length less than or equal to M symbols ($2M$ bits) and weight less than or equal $2M$. Given $\boldsymbol{u}^j$, ENC1 is guaranteed to be driven into the initial state with M input symbols. Also, the weight of the systematic symbols resulting from the $(K-j)$ unknown consecutive input symbols in $\overline{MWE1}^j$ will not bring any significant weight. It is better to consider their weight in $\overline{MWE2}^j$, because the $(K-j)$ unknown consecutive symbols get scattered and enter ENC2 in non-consecutive order leading to higher $\overline{MWE2}^j$. The modified definition for weight of ENC1 is

$$WE1 = \left[\begin{array}{l} \text{-Weight of both systematic and parity of ENC1} \\ \text{for known input symbols, plus} \\ \text{-Weight of parity of ENC1 for unknown input} \\ \text{symbols.} \end{array}\right. \quad (5)$$

This modified definition (*WE*1) leads to a lower weight for $MWE1^j$ compared to $\overline{MWE1}^j$ from the previous definition ($\overline{WE1}$), i.e.,

$$MWE1^j < \overline{MWE1}^j. \quad (6)$$

The modified definition for weight of ENC2 is *WE*2

$$WE2 = \left[\begin{array}{l} \text{-Weight of parity of ENC2 for known input} \\ \text{symbols, plus} \\ \text{-Weight of both systematic and parities of ENC2} \\ \text{for unknown input symbols.} \end{array}\right. \quad (7)$$

For the case $(K-j) >> M$, the modified definition (*WE*2) leads to significantly higher weight for $MWE2^j$ compared to $\overline{MWE2}^j$ in the previous definition ($\overline{WE2}$), i.e.,

$$MWE2^j >> \overline{MWE2}^j. \quad (8)$$

The lost weight in equation (6) is more than compensated for by the gained weight in equation (8). The effect of this compensation becomes especially clear if the number of unknown symbols $(K-j)$ in $\boldsymbol{u}^j$ is significantly larger than M ($(K-j) >> M$), which is the case for all interleavers of practical length. The modified definitions increase the minimum weight $MW^j = MWE1^j + MWE2^j$ resulting from the sum of minimum weights of ENC1 and ENC2, which leads to a *tighter* lower bound on $d_{\min}$ and eliminates the test of many input sequences, which in turn significantly reduces the computational complexity. That is,

$$\overline{MW}^j \le MW^j \le (\overline{w}(\boldsymbol{u}_{min}) = d_{\min}). \quad (9)$$

B. Other useful techniques to reduce the complexity

The following techniques have also been used to reduce the computational complexity in Garello's distance measurement routine:

1. Finding a tight upper bound for $d_{\min}$ at the beginning of the recursive process reduces the number of input sequences to be tested and thus lowers the computational complexity. Hence, use $\boldsymbol{u}^{j=K-1} = (00,\ldots,00,\otimes,\times)$, $\boldsymbol{u}^{j=K-2} = (00,\ldots,00,\otimes,\times,\times),\ldots,\boldsymbol{u}^{j=1} = (\otimes,\times,\ldots,\times)$ successively as the basis for the next iteration $u^{j+1}_{a\in\phi^0}$, where $\otimes \in \phi = \{01,10,11\}$ and $\times \in \phi^0 = \{00,01,10,11\}$. This will quickly lower the upper bound d^* because the first bases tested will have large numbers of consecutive leading zero symbols and very few trailing unknown symbols.

2. To reduce the computational complexity of $MWE1^j$ for all $j \in \{K-1,K-2,\ldots,1\}$, ENC1($\mathbf{u}^{\ge j} = (\times,\ldots,\times)$) can be computed *offline* for all states and all trellis sections $\{K-1,K-2,\ldots,1\}$ by applying the backward MVA. Similarly, the value ENC2($\mathbf{u}^{\ge j} = (\times,\ldots,\times)$) for all $j \in \{K-1,K-2,\ldots,1\}$ can also be computed offline. The values of ENC2($\mathbf{u}^{\ge j} = (\times,\ldots,\times)$) are used by the early stopping during the computation of the forward MVA (see below).

3. The $MWE2^{j+1}_{a\in\phi^0}$ resulting from the input sequence $\boldsymbol{u}^{j+1}_a$ can be computed separately by applying the forward MVA for each individual $a = 00,01,10$ or 11, but this leads to unnecessarily repeated computation over trellis sections $(0,\ldots,\pi^{-1}(j-1)$ and $(\pi^{-1}(j+1),\ldots,K-1)$. To lower the computational complexity, $MWE2^{j+1}_{00}$, $MWE2^{j+1}_{01}$, $MWE2^{j+1}_{10}$ and $MWE2^{j+1}_{11}$ can be computed in a single run by applying (See Fig 1):

a. Forward MVA from $t=0$ until $t=(\pi^{-1}(j)-1)$.

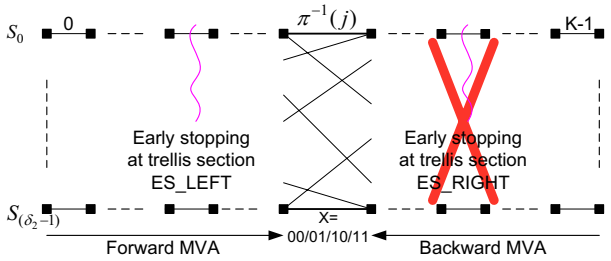


Fig. 1. This Figure illustrate how to obtain $MWE2_{00}^{j+1}$, $MWE2_{01}^{j+1}$, $MWE2_{10}^{j+1}$ and $MWE2_{11}^{j+1}$. It also shows the early stopping rule.

b. Backward MVA from $t = K$ until $t = (\pi^{-1}(j) + 1)$.
c. Combining the results of (a) and (b) at time $t = \pi^{-1}(j)$ for cases $\times = 00, 01, 10$ and 11 to get $MWE2_{00}^{j+1}$, $MWE2_{01}^{j+1}$, $MWE2_{10}^{j+1}$ and $MWE2_{11}^{j+1}$ respectively.

C. A new effective early stopping rule

As illustrated in Fig 1, the computation of the forward MVA can be stopped early at the trellis section ES_LEFT if the minimum weight over all states resulting from the sum of the current obtained weights at trellis section ES_LEFT and the offline computed ENC2 $\left(\mathbf{u}^{\geq \text{ES\_LEFT}}\right)$ weights is higher than d^*.

During the computation of $MWE2_{a\in\phi^0}^{j+1}$ the forward MVA starts at trellis section LEFT=0 and ends at trellis section ES_LEFT or $(\pi^{-1}(j) - 1)$ and the backward MVA starts at trellis section RIGHT=K-1 and ends at trellis section $(\pi^{-1}(j) + 1)$. The computation of $MWE2_{a\in\phi^0}^{j+2}$ will also use LEFT=0 and RIGHT=K-1, which leads to a re-computation of backward and forward MVA over many trellis sections resulting in higher computational complexity. To avoid this re-computation the following strategy is proposed:

- For all bases $\boldsymbol{u}^{j=K-1} = (00, \ldots, 00, \otimes, \times)$, $\boldsymbol{u}^{j=K-2} = (00, \ldots, 00, \otimes, \times, \times), \ldots, \boldsymbol{u}^{j=1} = (\otimes, \times, \ldots, \times)$, (previously kept input sequences $\boldsymbol{u}^j$), the computation of $MWE2_{a\in\phi^0}^{j+1}$ is done by setting LEFT=0 and RIGHT=K-1. The computed weights for all states at each trellis section must be stored in a matrix (i.e., MVA_MATRIX[K+1][δ_2]) .
- The values LEFT and RIGHT for the computation of $MWE2_{a\in\phi^0}^{j+2}$ for the next iteration depend on the positions $\pi^{-1}(j)$ and ES_LEFT from the previous iteration and the current position $\pi^{-1}(j+1)$ and can be obtained using the following simple structure
  - LEFT=$\min(\pi^{-1}(j+1), \pi^{-1}(j), \text{ES\_LEFT})$.
  - RIGHT=$\max(\pi^{-1}(j+1), \pi^{-1}(j))$.

The forward and backward MVA are initialized with MVA_MATRIX[LEFT] at trellis section LEFT and MVA_MATRIX[RIGHT] at trellis section RIGHT, respectively. The values of the MVA_MATRIX must be updated for the trellis sections $(\text{LEFT}, \ldots, \text{RIGHT})$ during the computation of $MWE2_{a\in\phi^0}^{j+2}$, so they can be used for the computation of $MWE2_{a\in\phi^0}^{j+3}$, if needed. It has been observed that this strategy leads to an average computational complexity reduction by a factor of 2.

The computation of the backward MVA can also be stopped earlier at the trellis section ES_RIGHT as shown in (Fig 1), if the minimum weight over all states resulting from the current obtained weights at trellis section ES_RIGHT is higher than d^*. However, it has been observed that an early stopping during the computation of the backward MVA does not lower the average computational complexity. In fact, the computational complexity is the same for a short interleaver (e.g., $K <= 100$ symbols) and is significantly higher for medium or large interleavers. Early stopping during the backward MVA is therefore not recommended.

VI. Distance Results

The new interleaver design for the DVB-RCS turbo codes is based on the dithered relative prime (DRP) approach [7], [8]. DRP interleavers are highly structured and ideal for designing low-memory interleaver banks for turbo codes. Each interleaver can be stored and implemented using only a few parameters. These parameters can be computed at run-time, if desired. The interleaver bank resolution is determined by the dither window size (WS). A WS of 4 works well for short blocks (e.g., $K < 200$) and a WS of 8 is better for medium blocks (e.g., $200 \leq K \leq 1000$). A value of WS=16 or higher is recommended for larger blocks. The DRP approach is applied to the interleaving of the double binary symbols. The bits within the symbols can also be manipulated, for example as per the original DVB-RCS standard. A number of different bit manipulations were investigated.

Using tail-biting turbo codes and structured interleavers, such as DRP interleavers, the distance spectra must repeat every Z symbols, where Z is the least common multiple of WS and all the puncturing mask lengths of the systematic and parity symbols. Also, the distance spectra must be a multiple of K/Z. This nice property of tail-biting turbo codes using structured interleavers is useful in testing the distance spectra obtained from a distance measurement method.

Table I shows the distance spectra of the 12 standardized DVB-RCS interleavers for the 7 standardized code rates. K is the packet length in symbols, R_c is the code rate and the 3 values $(d/A_{\text{d}}/W_{\text{d}})$ represent distance, multiplicity and information bit multiplicity, respectively. The lowest distance in each case is $d_{\min}$. Some of the distance results shown in Table I have also been independently computed by Rosnes, *et al.* [14] using a different modified version of Garello's original algorithm. The results in Table I agree exactly with the subset of cases considered in [14].

Table II shows additional distance results for the first 9 standardized interleaver sizes for DVB-RCS. These minimum distances were obtained using DRP interleavers that were found with an exhaustive search over all possible dither patterns with WS=4 (except some high code rates for K=752 symbols). All 4 repeating possibilities of swapping the 2 bits within a symbol over 2 consectutive symbols were also considered. For all the packet sizes and code rates, the minimum distances shown in Table II are as good or better than those shown in Table I for the standard interleavers.

TABLE I

DISTANCE RESULTS ($d/A_{\mathrm{d}}/W_{\mathrm{d}}$) FOR THE 12 STANDARDIZED DVB-RCS INTERLEAVERS.

K in Symbols	$R_c = 1/3$	$R_c = 2/5$	$R_c = 1/2$	$R_c = 2/3$	$R_c = 3/4$	$R_c = 4/5$	$R_c = 6/7$
48	21/72/240	17/48/192	13/72/168	8/120/360	4/8/32	4/12/36	3/16/32
64	25/192/1248	18/32/192	14/32/128	8/64/256	5/4/13	4/16/64	3/2/5
212	31/106/954 32/265/1643 33/106/901	25/159/1325 26/159/954 27/159/1219	18/159/954 19/159/1431 20/530/3551	11/159/901 12/265/1325 13/1802/11342	7/10/50 8/85/375 9/486/2335	6/159/742 7/530/2544 8/2544/12985	4/9/27 5/194/719 6/1228/5371
220	31/110/990	25/165/1265	19/165/1265	11/220/1210	7/10/35	6/110/550	4/2/8
228	30/114/855	24/57/342	18/171/1197	10/57/342	7/19/57	6/171/798	5/247/836
424	30/212/1696	24/212/1696	18/212/1696	13/530/3710	8/21/84	7/212/954	5/80/287
432	31/108/972	27/324/3132	18/108/972	12/432/2160	8/36/144	6/108/324	5/72/288
440	28/110/1100	22/110/1100	16/110/1100	12/110/440	8/27/108	8/1100/5500	4/10/40
752	33/376/3384 35/376/3760 36/752/6392	27/376/3384 28/376/3008 29/376/3384	19/376/3384 20/376/3008 22/752/6768	12/188/1316 14/752/5264 15/1504/12220	9/27/171 10/148/1025 11/1462/9674	9/3572/20680 10/8836/56212 11/31020/212252	6/199/826 7/1578/7269 8/9144/49558
848	36/848/7420	28/636/5088	20/636/5088	13/212/1272	9/1/4	8/212/848	5/67/176
856	33/428/3852	27/428/3852	19/428/3852	12/214/1498	9/8/40	9/3210/17762	5/16/64
864	36/864/7560	28/648/5184	20/648/5184	13/216/1296	9/72/144	8/648/2160	6/288/1008

TABLE II

DISTANCE RESULTS ($d_{\min}/A_{\mathrm{d}}/W_{\mathrm{d}}$) WITH EXHAUSTIVE SEARCH FOR DRP INTERLEAVERS OF WS=4. THE STANDARDIZED PUNCTURING WERE USED.

K in Symbols	$R_c = 1/3$	$R_c = 2/5$	$R_c = 1/2$	$R_c = 2/3$	$R_c = 3/4$	$R_c = 4/5$	$R_c = 6/7$
48	24/24/144	20/192/960	14/96/288	8/24/60	6/64/224	5/108/420	4/156/520
64	28/384/2528	22/64/384	17/256/1920	9/64/256	6/16/67	5/16/48	3/1/2
212	36/1007/7420	28/53/530	22/1908/14416	12/106/318	8/24/96	8/2173/9858	4/1/4
220	36/715/6380	28/110/880	21/220/1210	12/55/275	8/9/42	8/1815/8305	4/1/3
228	36/342/1710	29/627/3933	22/1995/14649	12/285/1197	9/133/684	8/2337/12084	6/1216/5092
424	36/106/636	30/848/5088	23/530/4770	14/318/2014	9/2/11	9/1802/9434	5/1/4
432	36/108/648	30/864/5184	23/756/7020	14/540/3348	10/72/360	9/2160/10584	6/36/180
440	36/330/2970	30/880/5280	23/880/7260	14/330/2090	9/1/2	9/1760/9130	6/246/1155
752	36/3196/24064	30/1504/9024	22/3760/28388	14/188/1692	10/137/793	10/7332/41924	6/108/479

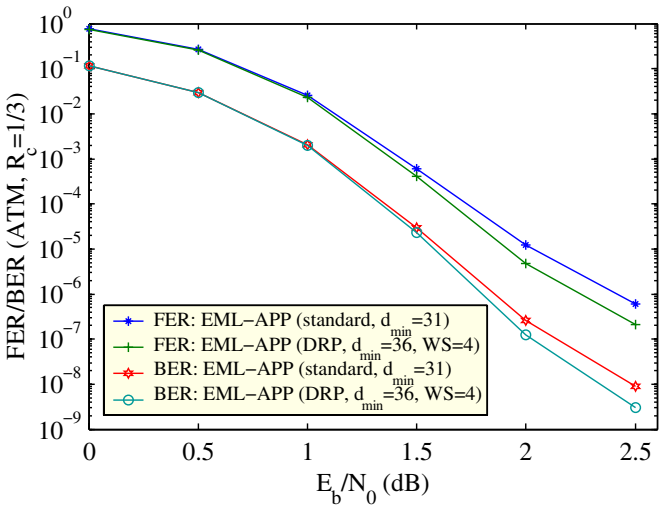
TABLE III

MINIMUM DISTANCES FOR CODE RATE 1/3 WITH AN EXHAUSTIVE SEARCH FOR DRP INTERLEAVERS WITH WS=1,2 AND 4.

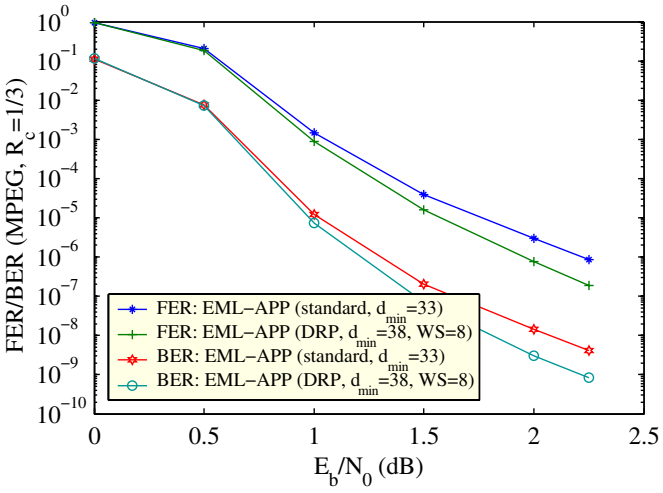
WS	K=48	K=64	K=212	K=220	K=228	K=424	K=432	K=440	K=752
1	24	27	28	28	28	28	28	28	28
2	24	27	32	32	32	32	32	32	32
4	24	28	36	36	36	36	36	36	36

Results were also generated for DRP interleavers with WS values of 1 and 2 and a code rate of 1/3. Table III shows that the upper bounds on $d_{\min}$ with WS=1,2 and 4 are 28, 32 and 36, respectively. Note that DRP interleavers with WS=1 correspond to simple relative prime interleavers. Results for WS=1,2 and 4 were obtained with an exhaustive search considering the four repeating possibilities of swapping the 2 bits within symbol over 2 consecutive symbols. To get a $d_{\min}$ higher than 36, the dither window size must be greater than WS=4. Unfortunately, an exhaustive search with WS=8 is impossible in a reasonable time due to the very large number of dither patterns to be tested. Thus, the search was limited to randomly selected dither patterns. A $d_{\min}$ of 38 with $A_{\mathrm{d}_{\min}}$=94 and $W_{\mathrm{d}_{\min}}$=752 was obtained for MPEG packets (K=752 symbols) with DRP and WS=8, whereas the $d_{\min}$ of the standardized DVB-RCS interleaver is 33.

Figure 2 shows simulated error rate results for rate 1/3 turbo codes using 8 full iterations. The constituent decoders used enhanced max-log APP decoding [15] with an extrinsic scale factor of 0.75. To reduce the statistical differences, the same noise sequence was used for the two interleaver designs at each SNR value. Fig 2(a) shows, for an ATM



(a)



(b)

Fig. 2. FER and BER for ATM and MPEG packets of code rate R_c=1/3 (QPSK/AWGN). The size of overlap is 50 symbols (100 bits) and 75 symbols (150 bits) for ATM and MPEG packets, respectively. The number of full iterations is 8. Enhanced max-log APP (EML-APP) decoding with a fixed scale factor of 0.75 for the extrinsic was used. For ATM packets, 100 million packets were simulated for both interleavers at 2.5 dB. For MPEG packets, 300 million packets were simulated for both interleavers at 2.25 dB.

packet with K=212 symbols, an improvement with the DRP interleaver of greater than 0.15 dB at high SNRs compared to the standardized DVB-RCS interleaver. Fig 2(b) shows, for an MPEG packet with K=752 symbols, an improvement with the DRP interleaver greater than 0.25 dB at high SNRs compared to the standardized DVB-RCS interleaver.

VII. Conclusion

Garello's distance measurement method for single binary turbo codes was extended to double binary tail-biting turbo codes. An efficient early stopping rule that reduces the computational complexity of the distance measurement method by a factor of 2 was presented. This reduction in computational complexity is significant because the measurement of high distances (i.e., $d_{\min} > 50$) can take weeks. Distance results for the 12 standardized DVB-RCS interleavers over all standardized code rates were presented. A new interleaver design for DVB-RCS based on the dithered relative prime (DRP) approach was also presented. The new interleaver design achieves, for a code rate of 1/3, an improvement of at least 0.15 dB and 0.25 dB at high SNRs for ATM and MPEG packets, respectively, compared to the standardized DVB-RCS interleavers.

VIII. Acknowledgements

The authors thank Dr. John Lodge, Ken Gracie and Pascal Chahine at the Communication Research Centre in Ottawa for their encouragement. Special thanks to Professor Roberto Garello at the Dipartimento di Elettronica of Politecnico di Torino for graciously elaborating on some finer points related to his distance measurement method.

References

[1] B. Vucetic and J. Yuan, *Turbo Codes: Principles and Applications*. Kluwer, 2000.

[2] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near shannon limit error-correcting coding and decoding: Turbo-codes," in *Proc. IEEE Int. Conf. Commun. (ICC'93)*, pp. 1064–1070, May 1993. (Geneva, Switzerland).

[3] J. Lodge, R. Young, P. Hoeher, and J. Hagenauer, "Using seperable MAP 'filters' for the decoding of product and concatinated codes," in *Proc. IEEE Int. Conf. Commun. (ICC'93)*, pp. 1740–1745, May 1993. (Geneva, Switzerland),(www.crc.ca/fec).

[4] L. C. Perez, J. Seghers, and D. J. Costello, Jr., "A distance spectrum interpretation of turbo codes," *IEEE Trans. Inform. Theory*, vol. 42, pp. 1698–1709, Nov. 1996.

[5] S. Benedetto and G. Montorsi, "Design of parallel concatenated convolutional codes," *IEEE Trans. Commun.*, vol. 44, pp. 591–600, May 1996.

[6] M. Breiling and J. Huber, "Combinatorial analysis of the minimum distance of turbo codes," *IEEE Trans. Inform. Theory*, vol. 47, pp. 2737–2750, Nov. 2001.

[7] S. Crozier and P. Guinand, "High-performance low-memory interleaver banks for turbo-codes," in *Proc. 54th IEEE Vehicular Tech.*, pp. 2394–2398, Oct. 2001. (Atlantic City, New Jersey, USA),(www.crc.ca/fec).

[8] S. Crozier and P. Guinand, "Distance bounds and the design of high-distance interleavers for turbo-codes," in *Proc. 21st Biennial Symposium Commun.*, pp. 10–14, June 2002. (Kingston, Ontario, Canada),(www.crc.ca/fec).

[9] S. Crozier and P. Guinand, "Distance upper bounds and true minimum distance results for turbo-codes designed with DRP interleavers," in *Proc. 3rd Int. Symp. Turbo codes*, pp. 169–172, Sept. 2003. (Brest, France).

[10] R. Garello, P. Pierleoni, and S. Benedetto, "Computing the free distance of turbo codes and serially concatenated codes with interleavers: Algorithms and applications," *IEEE J. on Selected Areas Commun.*, vol. 19, pp. 800–812, May 2001.

[11] C. Berrou, S. Vaton, M. Jézéquel, and C. Douillard, "Computing the minimum distance of linear codes by the error impulse method," in *Proc. IEEE Globecom*, pp. 10–14, Nov. 2002. (Taipei, Taiwan).

[12] "Interaction channel for satellite distribution systems." ETSI EN 301 790, V1.3.1, Mar. 2003.

[13] C. Berrou, C. Douillard, and M. Jézéquel, "Multiple parallel concatenation of circular recursive convolutional (CRSC) codes," *Annals Telecommun.*, vol. 54, No. 3-4, pp. 166–172, March-April 1999.

[14] E. Rosnes and O. Ytrehus, "An efficient algorithm for tailbiting turbo code weight distribution calculation," in *Proc. 3rd Int. Symp. Turbo codes*, pp. 439–442, Sept. 2003. (Brest, France).

[15] Y. Ould-Cheikh-Mouhamedou, P. Guinand, and P. Kabal, "Enhanced Max-Log-APP and enhanced Log-APP decoding for DVB-RCS," in *Proc. 3rd Int. Symp. Turbo codes*, pp. 259–269, Sept. 2003. (Brest, France).

