Decoding M-binary turbo codes by the dual method

Yannick Saouter

Decoding M-binary turbo codes by the dual method

Yannick Saouter
Ecole Nationale Supérieure de
Télécommunications de
Bretagne, Technopole Brest-Iroise,
BP832, 29285 Brest Cédex, France.

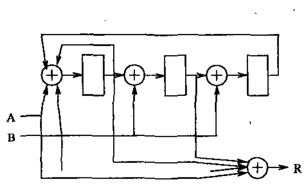
Abstract — **Joint effort by several researchers has led to the derivation of a maximum *a posteriori* decoding algorithm for classical binary turbo codes by the dual method. This algorithm leads to reduced complexity with regard to the classical MAP algorithm for high-rate binary turbo codes. In this article, we extend the result to duo-binary turbo codes and, more generally, to any M-binary turbo codes.**

I. Introduction

ETSI (European Telecommunications Standards Institute) is in the process of adopting new channel coding for interactivity in Digital Video Broadcasting (DVB) for satellite (RCS: Return Channel over Satellite) and terrestrial (RCT: Return Channel over Terrestrial) transmission. The error-correcting code is a duo-binary 8-state convolutional turbo code, specified for various block sizes (from 12 to 216 bytes) and coding rates (from 1/3 to 6/7).

II. Duo-binary codes

Duo-binary codes were introduced in the domain of turbo codes by Berrou et al. [1]. These codes are in fact binary codes with an unpunctured rate of 2/3: each pair of data bits produces one redundancy bit. Those codes exhibit better correcting performances and do not suffer from severe flattening at low error rates, unlike classical binary turbo codes. In this article, we consider the 8-state duo-binary recommended by the DVB datasheet standard [2], as an illustrative example. Figure 1 describes the layout of the encoder as well as its transition graph.



initial state / future state and redundancy				
S	A=0 B=0	A=1 B=0	A=0 B=1	A=1 B=1
0	S'=0 R=0	S'=6 R=1	S'=1 R=1	S'=7 R=0
1	S'=2 R=1	S'=4 R=0	S'=3 R=0	S'=5 R=1
2	S'=5 R=1	S'=3 R=0	S'=4 R=0	S'=2 R=1
3	S'=7 R=0	S'=1 R=1	S'=6 R=1	S'=0 R=0
4	S'=1 R=0	S'=7 R=1	S'=0 R=1	S'=6 R=0
5	S'=3 R=1	S'=5 R=0	S'=2 R=0	S'=4 R=1
6	S'=4 R=1	S'=2 R=0	S'=5 R=0	S'=3 R=1
7	S'=6 R=0	S'=0 R=1	S'=7 R=1	S'=1 R=0

Figure 1: Architecture and state lattice of the duo-binary convolutional encoder

III. MAP dual decoding

Following the works of Hartmann [3], Hagenauer introduced MAP dual decoding for binary turbo codes [4]. This work was extended to the general case of turbo codes defined over Galois fields by Berkmann [5]. In this article, we extend it to M-binary turbo codes which do not come within the scope of Berkmann's work and seem to have more practical applications. Indeed, although duo-binary turbo codes can be decoded by Hagenauer's method, our improvement is to extract duo-binary extrinsic values (instead of binary extrinsic values for Hagenauer). This leads to better performance in terms of bit error rate. For the sake of simplicity, we present here the equations for the case of duo-binary codes and then explain how they generalize to the general M-binary case.

Let C be a systematic duo-binary convolutional code. Let $\mathbf{x} = (\mathbf{x}_0, \mathbf{x}_1, ..., \mathbf{x}_{N-1})$ be a frame of N duo-binary information data symbols to be encoded by the encoder of C. The encoder of C will be denoted $\mathbf{C}$. We suppose that $\mathbf{x}$ is a terminated sequence (i.e. if the encoding of $\mathbf{x}$ begins in the zero state of $\mathbf{C}$, it also finishes in the zero state of $\mathbf{C}$). The output sequence produced by $\mathbf{C}$ is then $\mathbf{c} = (\mathbf{x}_0, y_0, \mathbf{x}_1, y_1, ..., \mathbf{x}_{N-1}, y_{N-1})$ where y_i denotes the i-th binary redundancy produced by the encoder. The codeword $\mathbf{c}$ is then modulated by a BPSK and transmitted over a memoryless channel with white Gaussian noise with zero mean. We suppose that the modulation is the conventional one in which a data bit 0 is emitted over the channel as -1 and a data bit 1 as $+1$.

From the decoder point of view, a frame $\mathbf{r} = (r_0, r_1, r'_0, r_2, r_3, r'_1, ..., r_{2N-2}, r_{2N-1}, r'_{N-1})$ with real numbers values is received, where r_{2i} (resp. r_{2i+1}) corresponds to the received sample for the first (resp. second) bit of the duo-binary symbol $\mathbf{x}_i$ and r'_i corresponds to the received sample for y_i. For any $0 \leq t < N$, any $s_1, s_2 \in \{0,1\}$ and any $s \in \{0,1\}$, we define $J(t, s_1, s_2) = Prob(r_{2t}, r_{2t+1} \mid s_1, s_2)$ and $I(t, s) = Prob(r'_t \mid s)$.

More precisely, if τ is the standard deviation of the noise of the transmission channel (computed from the $Eb/N0$ value and the rate of the code by conventional formulas), we have:

$$I(t,s) = \frac{1}{\sqrt{2\pi\tau}} exp(\frac{-((r'_t - m_s)^2)}{2\tau^2})$$

$$J(t,s_1,s_2) = \frac{1}{2\pi\tau} exp(\frac{-((r_{2t} - m_{s_1})^2 + (r_{2t+1} - m_{s_2})^2)}{2\tau^2})$$

if m_s is the noiseless modulation level of the bit s (i.e. $m_0 = -1$ and $m_1 = 1$).

Let $\tilde{C}$ be the reciprocal dual code of C (in the sense of convolutional codes), whose encoder is $\tilde{\mathbf{C}}$. We denote Σ the set of the states of $\tilde{\mathbf{C}}$. For any state $\sigma \in \Sigma$ we denote $Succ(\sigma)$ (resp. $Pred(\sigma)$) the set of states of $\tilde{\mathbf{C}}$ which can be reached from state σ (resp. which can reach state σ) after a single step of convolutional encoding. Let σ be any state in Σ and let σ' be any state in $Succ(\sigma)$. Then during a transition from σ to σ', encoder $\tilde{\mathbf{C}}$ will produce three bits as outputs that we will denote $z_1(\sigma,\sigma')$, $z_2(\sigma,\sigma')$ and $z_3(\sigma,\sigma')$. These three bits are not to be considered in any order: if you consider a codeword of C beginning and ending in the zero state of $\mathbf{C}$, then any codeword of $\tilde{C}$ with the same length beginning and ending in the zero state of $\tilde{\mathbf{C}}$ should have a null dot product over $\mathbf{GF}(2)$ with the codeword of C if $z_1(\sigma,\sigma')$ (resp. $z_2(\sigma,\sigma')$) is multiplied with the first (resp. second) bit of the data symbol of the codeword of C, and $z_3(\sigma,\sigma')$ is multiplied with the binary redundancy of the codeword of C. This defines $z_1(\sigma,\sigma')$, $z_2(\sigma,\sigma')$ and $z_3(\sigma,\sigma')$ uniquely and without ambiguity.

For any $\sigma \in \Sigma$, any $\sigma' \in Succ(\sigma)$ and $0 \leq t < N$ we define the branch metric at time t for a transition from σ to σ', that we will denote as $BM(t,\sigma,\sigma')$ by the following equation:

$$
\begin{aligned}
&BM(t,\sigma,\sigma') = \\
&\quad \Big(\sum_{s_1,s_2\in\{0,1\}} (-1)^{z_1(\sigma,\sigma').s_1+z_2(\sigma,\sigma').s_2}.J(t,s_1,s_2)\Big). \\
&\qquad\qquad \Big(\sum_{s\in\{0,1\}} (-1)^{z_3(\sigma,\sigma').s}.I(t,s)\Big)
\end{aligned}
$$

Now, we define the forward metrics $A_\sigma(t)$ for any $\sigma \in \Sigma$ and $0 \leq t < N$ and the backward metrics $B_\sigma(t)$ for any $\sigma \in \Sigma$ and $0 < t \leq N$ by the following recurrence equations:

$$
\begin{aligned}
&A_0(0) = 1, A_\sigma(0) = 0 \text{ if } \sigma \neq 0 \\
&B_0(N) = 1, B_\sigma(N) = 0 \text{ if } \sigma \neq 0 \\
&A_\sigma(t) = \sum_{\sigma'\in Pred(\sigma)} A_{\sigma'}(t-1).BM(t-1,\sigma',\sigma) \\
&B_\sigma(t) = \sum_{\sigma'\in Succ(\sigma)} B_{\sigma'}(t+1).BM(t,\sigma,\sigma')
\end{aligned}
$$

Then if $m = (s_1, s_2)$ is a duo-binary symbol, the extrinsic value at time t for symbol m, required by the turbo decoding process and that we will denote as $EXTR_m(t)$, is equal to:

$$
\begin{aligned}
&EXTR_m(t) = -log\Big[\\
&\sum_{\sigma\in\Sigma}\sum_{\sigma'\in Succ(\sigma)} ((-1)^{z_1(\sigma,\sigma').s_1+z_2(\sigma,\sigma').s_2}.A_\sigma(t).B_{\sigma'}(t+1). \\
&\qquad\qquad (I(t,0)+(-1)^{z_3(\sigma,\sigma')}I(t,1)))\Big]
\end{aligned}
$$

The proofs of these formulas are outside the scope of this paper but they will be published in a more detailed article.

IV. Generalization to M-binary codes

In the previous section, we have described computation in the case of duo-binary turbo codes only, in order to lighten the notation. However, these formulas generalize to any M-binary codes.

For instance, if we now consider a tri-binary code, the J terms now have four function indices and are defined by the following equation:

$$
\begin{aligned}
&J(t,s_1,s_2,s_3) = \\
&\Big(\frac{1}{2\pi\tau}\Big)^{3/2} exp\Big(\frac{-(\sum_{i=1}^{3}(r_{3t+i-1}-m_{s_i})^2)}{2\tau^2}\Big)
\end{aligned}
$$

The branch metrics $BM(t,\sigma,\sigma')$ have the same expression, with the difference that the summation of the first factor term is now over $\{s_1, s_2, s_3 \in \{0,1\}\}$ and the exponent now has an additional term $z_3(\sigma,\sigma').s_3$. The exponent of the second factor still concerns the redundancy of the code and thus is $z_4(\sigma,\sigma').s$ where z_4 is the fourth bit produced by the encoder of the dual reciprocal code.

The expression of the state metrics does not change and finally the expression of the extrinsic value is modified with an additional term in the first exponent, namely $z_3(\sigma,\sigma').s_3$ while the second exponent becomes $z_4(\sigma,\sigma')$.

V. Circular codes

In what follows, we will compare the performance of the dual duo-binary MAP decoding with that of the dual binary MAP decoding [6] for the case of the DVB-RCS code [2]. However, in what precedes, we have considered codes whose encoding begins and ends in the zero state. Practically, in applications, those codes require a set of termination symbols, in order to ensure that the end state is effectively the zero state.

In the DVB-RCS encoding scheme, the beginning and ending state are computed, given the information data, in order that they be equal. This kind of code is called circular. This allows space to be saved for information data, since there are no longer any terminating symbols. Moreover, since the frame is circular, there are no side effects due to the initialization of forward state metrics and backward state metrics at the beginning and the end of the frame, and any extrinsic value benefits from the decoding of the entire frame.

However, the orthogonality property is not preserved with such codes and this property is critical for the computation of extrinsic values by the dual method. In order to circumvent the problem, we use the forgetfulness property of convolutional decoding. Indeed, it is a well known fact in turbo decoding that the extrinsic values for a given symbol of the frame essentially depend on the channel values of its nearest neighbors and the dependence decreases exponentially. This property allows sliding window decoding and has also been validated for dual binary MAP decoding [7], which is in any case algebraically equivalent to classical binary MAP decoding [8]. Similarly, dual M-binary MAP decoding is equivalent to classical M-binary MAP decoding and thus the forgetfulness property also holds in our approach.

Thus, in order to deal with circular codes, we begin the decoding by a prologue which is in charge of estimating the initial state metrics of the backward and the forward phase. If N is the length of the frame and L is the length of the prologue, then the initial forward state metrics are computed by making a forward prologue over the channel samples of the interval $[N-L, N-1]$. At time $N-L$ all the states are considered *a priori* possible and thus all the state metrics are initialized to 1.0. Since the frame is circular, the state metrics computed at the end of the prologue after the samples of time $N-1$, are acceptable initializations for the regular forward phase beginning with the sample of time 0. Similarly, the initial backward metrics are computed by a backward prologue over the interval $[0, L]$.

VI. Implementation for the DVB-RCS code

Let us denote $A(D) = \sum_{-\infty}^{+\infty} a_i D^i$, $B(D) = \sum_{-\infty}^{+\infty} b_i D^i$ and $R(D) = \sum_{-\infty}^{+\infty} r_i D^i$, where a_i (resp. b_i) are the input bits of the DVB-RCS encoder depicted in figure 1 and r_i is the output redundancy. Then the transfer function of this code can be computed by the conventional method [9] and we find:

$$\begin{aligned}(A(D) \quad B(D) \quad R(D)) &= \\ &(A(D) \quad B(D)) \cdot \begin{pmatrix} 1 & 0 & \frac{D^3+D+1}{D^3+D^2+1} \\ 0 & 1 & \frac{D^3+D^2+D+1}{D^3+D^2+1} \end{pmatrix}\end{aligned}$$

The transfer matrix of the dual code is then defined as the matrix $H(D)$ such that $G(D).H^T(D) = 0$ where $G(D)$ is the transfer matrix of the code explicitly given in the previous equation. The matrix $H(D)$ is not uniquely defined and thus we can choose the one with the simplest expression. In fact, we have:

$$\begin{pmatrix} 1 & 0 & \frac{D^3+D+1}{D^3+D^2+1} \\ 0 & 1 & \frac{D^3+D^2+D+1}{D^3+D^2+1} \end{pmatrix} \cdot \begin{pmatrix} D^3+D+1 \\ D^3+D^2+D+1 \\ D^3+D^2+1 \end{pmatrix} = 0$$

Thus we may choose $H(D) = (D^3 + D + 1,\ D^3 + D^2 + D+1,\ D^3+D^2+1)$. Now the transfer matrix of the reciprocal dual code is $\tilde{H}(D) = H(D^{-1})$. We then have $\tilde{H}(D) = (1 + D^{-1} + D^{-3},\ 1+D^{-1}+D^{-2}+D^{-3},\ 1+D^{-2}+D^{-3})$. This code has a rate of $1/3$ since the initial code has a rate of $2/3$. Thus, for one input bit, we have three output redundancy bits. Let x_i be the samples of input bits and r_i^1, r_i^2 and r_i^3 be the samples of output redundancies. According to the expression of $\tilde{H}(D)$ we then have the recurrence equations:

$$\begin{aligned} r_i^1 &= x_i + x_{i-1} + x_{i-3} \\ r_i^2 &= x_i + x_{i-1} + x_{i-2} + x_{i-3} \\ r_i^3 &= x_i + x_{i-2} + x_{i-3} \end{aligned}$$

The corresponding encoder $\tilde{\mathbf{C}}$ is depicted in figure 2 and the reader may verify that any sequence of the initial DVB-RCS code beginning and ending in the zero state has a null dot product over $\mathbf{GF}(2)$ with any sequence produced by the encoder $\tilde{\mathbf{C}}$ of the same length also beginning and ending in the zero state if a_i (resp. b_i, r_i) meets r_i^1 (resp. r_i^2, r_i^3). This order is not arbitrary: it is imposed by the initial matrix conventions we chose.

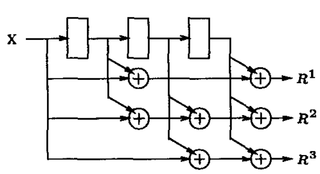


Figure 2: Architecture of the dual reciprocal duo-binary convolutional encoder

VII. Performance

In this section we compare our dual duo-binary MAP decoding with the dual binary MAP decoding described in [6] for a special case of the DVB-RCS encoding standard. We choose two different frame lengths (a long one with 188 information bytes and a short one with 12 information bytes) and the specifications of the permutations used are to be found in [2]. For both programs, the computations were done in floating point numbers and we use prologues at each iteration of the turbo decoding process. The number of iterations is the same in both programs and is equal to 8. The bit error ratio curves of each decoding are depicted in figure 3 for a frame length of 188 bytes and in figure 4 for a frame length of 12 bytes.

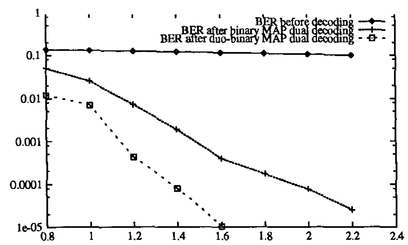


Figure 3: Performance of binary and duo-binary MAP decoding for DVB-RCS standard (frame length=188 bytes)

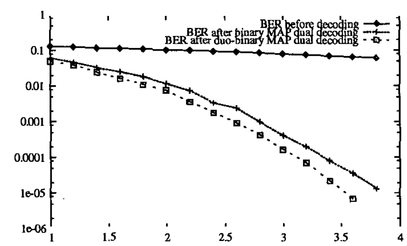


Figure 4: Performance of binary and duo-binary MAP decoding for DVB-RCS standard (frame length=12 bytes)

In the case of 188 byte frame, we see a great degradation of performance if we use binary MAP dual decoding instead of duo-binary MAP decoding. The gap attains 0.6 dB at a bit error ratio of 10^{-5}. In the case of 12 byte frames, the difference is only 0.2 dB at the same bit error ratio. The explanation is that, in longer frames, the duo-binary decoder takes advantage of a greater number of correctly evaluated extrinsic values while the binary decoder suffers from correlation between the extrinsic values of each bit of a duo-binary symbol. This correlation is eliminated in the duo-binary case since each symbol is treated as a pair, while the binary decoder treats each bit of a pair independently.

VIII. Conclusion

In this article, we have presented a new soft decision algorithm that makes use of duality to compute M-binary extrinsic values. We have shown that, in the case of the DVB-RCS code, our approach outperforms the dual binary MAP algorithm by more than 0.2 dB on an AWGN channel. The following step will be to derive simplified versions of this algorithm with a silicon integration.

IX. Acknowledgments

The author wishes to thank Prof. Claude Berrou at the Ecole Nationale Supérieure de Télécommunications de Bretagne (Brest, France) for his support and his helpful comments about the work presented in this article.

REFERENCES

[1] C. Berrou and M. Jezequel, "Non-binary convolutional codes for turbo coding," *Electronics Letters*, vol. 35, no. 1, pp. 39, January 1999.

[2] "Digital Video Broadcasting (DVB); Interaction channel for satellite distribution systems," ETSI EN 301 790 V 1.2.2, December 2000, http://pda.etsi.org/pda/home.asp?wki_id=8177.

[3] C.R.P. Hartmann and L.D. Rudolph, "An optimum symbol-by-symbol decoding rule for linear codes," *IEEE Trans. Inform. Theory*, vol. IT-22, pp. 514–517, September 1974.

[4] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutionnal codes," *IEEE Trans. Inform. Theory*, vol. IT-42, pp. 429–445, 1996.

[5] J. Berkmann, *Iterative Decoding of Nonbinary Codes*, Ph.D. thesis, Munich University of Technology, 2000.

[6] S. Riedel, "MAP Decoding of Convolutional Codes Using Reciprocal Dual Codes," *IEEE Trans. Inform. Theory*, vol. IT-44, no. 3, May 1998.

[7] Christian Weiss, *Error Correction with Tail-Biting Convolutional Codes*, Ph.D. thesis, Munich University of Technology, 2000.

[8] L.R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate," *IEEE Trans. on Information Theory*, vol. IT-20, pp. 284–287, March 1974.

[9] J.G. Proakis, *Digital Communications*, McGraw-Hill, 2000.