



HAL
open science

Multi-non-binary turbo codes

Horia Balta, Catherine Douillard, Radu Lucaciu

► **To cite this version:**

Horia Balta, Catherine Douillard, Radu Lucaciu. Multi-non-binary turbo codes. EURASIP Journal on Wireless Communications and Networking, 2013, 2013 (2013:279), 10.1186/1687-1499-2013-279 . hal-00931015

HAL Id: hal-00931015

<https://hal.science/hal-00931015v1>

Submitted on 14 Jan 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

RESEARCH

Open Access

Multi-non-binary turbo codes

Horia Balta^{1,2}, Catherine Douillard³ and Radu Lucaci^{1*}

Abstract

This paper presents a new family of turbo codes called multi-non-binary turbo codes (MNBTCs) that generalizes the concept of turbo codes to multi-non-binary (MNB) parallel concatenated convolutional codes (PCCC). An MNBTC incorporates, as component encoders, recursive and systematic multi-non-binary convolutional encoders. The more compact data structure for these encoders confers some advantages on MNBTCs over other types of turbo codes, such as better asymptotic behavior, better convergence, and reduced latency. This paper presents in detail the structure and operation of an MNBTC: MNB encoding, trellis termination, Max-Log-MAP decoding adapted to the MNB case. It also shows an example of MNBTC whose performance is compared with the state-of-the-art turbo code adopted in the DVB-RCS2 standard.

Keywords: Turbo codes; Non-binary code; Multi-non-binary recursive systematic convolutional code; Galois field

1 Introduction

Two decades after their introduction in [1], turbo codes (TCs) have found their utility in numerous communication systems. TCs can be found in LTE, DVB or in deep space communications standards [2-4]. This TC success was possible because of the numerous studies in the past years, leading to the diversification of TC families and to the emergence of new codes, whose decoding is based on the principle of iterative or turbo decoding. Thus, the evolution of TC in terms of component convolutional encoder after the now classic single binary turbo codes (SBTC) [1], was followed by double/multibinary turbo codes (D/MBTCs) [5,6] and non-binary turbo codes (NBTCs) [7-9]. These new TC families aim to improve SBTC performance, especially by lowering the error floor. Proposing a new family of multi-non-binary turbo codes (MNBTCs), this paper may be seen as a continuation of these concerns. The MNBTC was firstly introduced in [10], but this paper presents a more complete description of MNBTCs and also an example of more efficient MNBTCs. An MNBTC has recursive and systematic convolutional component encoders, with several non-binary inputs. This evolution from single-binary to (multi-)non-binary is found at low-density parity-check (LDPC) codes [11], but it happened much earlier [12].

The first benefit brought forth by the new family of MNBTCs is latency reduction, the data block being

more compact. Another benefit, argued by practical results, is a lower error floor. Basically, simulations show that the waterfall region extends below 10^{-8} of frame error rate (FER). Furthermore, MNB turbo decoders have a better convergence at high SNRs.

In addition, MNBTCs as non-binary LDPCs (NB-LDPCs) can be easily combined with high-order modulations, yielding increased bandwidth efficiency. The price for these benefits is the increased complexity of the component code trellis and, therefore, of the computational effort of the decoder. But, due to the evolution of the processing capacity, it is expected that the complexity of non-binary codes to no longer pose a disadvantage in relation to binary codes. This is confirmed by the extensive studies of NB-LDPC codes, [13-17].

The structure of the paper is as follows: in Section 2, we describe the MNBTC encoder and decoder, as well as component codes, the MNBTC structure, the encoding process and trellis termination, the Max-Log-MAP algorithm adjusted to the MNB case, and details of interleaving. In Section 3, we propose a memory-3 MNBTC with two inputs offering a good trade-off between performance and complexity. The process of component encoder selection is explained and performance is assessed through simulations in both the AWGN and the Rayleigh fading channels, in conjunction with BPSK and 16-QAM modulations. Performance is compared with the DBTC of the DVB-RCS2 standard [3] and also with some NB-LDPC from literature. Section 4 concludes the paper.

* Correspondence: radu.lucaci@etc.upt.ro

¹Faculty of Electronics and Telecommunications, University POLITEHNICA of Timisoara, V. Parvan 2, 300223, Timisoara, Romania

Full list of author information is available at the end of the article

2 The structure of MNBTCs

2.1 Multi-non-binary convolutional encoders

In this section, we describe the encoding scheme of the constituent codes of MNBTCs. Each constituent encoder has R non-binary inputs taking values in the Galois field $GF(2^Q)$, and is then referred to as a multi-non-binary (MNB). In Figure 1, we present the general scheme of a memory- M recursive systematic MNB convolutional encoder (MNBCE), with a rate $R_c = R/(R + 1)$. This scheme is known as the observer canonical form [18]. Each cell of the register in Figure 1 stores a vector of Q bits at a time. All the links are supposed to have a width equal to Q in order to carry Q -bit vectors. Each block labeled $g_{m,r}$, with $0 \leq m < M$, $0 \leq r \leq R$, represents a multiplier in $GF(2^Q)$ by the generator polynomial coefficient $g_{m,r}$. Adders also performs the sum in $GF(2^Q)$. At time n , the R encoder inputs consist of a word of symbols $u^n = [u_R^n \dots u_2^n u_1^n]$ with $u_r^n = [u_{r,Q-1}^n \dots u_{r,1}^n u_{r,0}^n]$, $1 \leq r \leq R$, $0 \leq n < N$, where N is the interleaver length and $u_{r,q}^n$ are binary coefficients. The MNB convolutional encoder generates $R + 1$ outputs, of which the first R symbols correspond to the R inputs (systematic part) and the last symbol is a redundant symbol. The current encoder state at time n is given by the content of the shift register, $S_0^n, S_1^n, \dots, S_{M-1}^n$. We define the encoder state vector by $s^n = [s_{M-1}^n \dots s_1^n s_0^n]$.

To simplify the notations, we understand by u^n both its vector expression $[u_R^n \dots u_2^n u_1^n]$ and its scalar one u^n being an element of $GF(2^{Q \cdot R})$, the distinction being made in context. Similarly, u_r^n is in $GF(2^Q)$ and s^n is in $GF(2^{Q \cdot M})$. Also, we note that we have chosen to place temporal indices in the top right and the rest of the indices in the bottom right.

The relation between the encoder state at time $n + 1$ and the encoder state at time n can be expressed in the compact form:

$$(s^{n+1})_{1 \times M} = (u^n)_{1 \times R} \cdot (G_T)_{R \times M} + (s^n)_{1 \times M} \cdot (T)_{M \times M} \quad (1)$$

$$u_0^n = u^n \cdot G_L + s^n \cdot W \quad (2)$$

where $G_T = G_L \cdot G_F + G_0^T$ and $W = ([0 \ 0 \ \dots \ 0 \ 1]_{1 \times M})^T$, with T denoting the transpose operation and $G_0 = [g_{m,r}]_{M \geq m \geq 1, R \geq r \geq 1}$, $G_F = [g_{M,0} \ g_{M-1,0} \ \dots \ g_{1,0}]$, $G_L = [g_{0,R} \ g_{0,R-1} \ \dots \ g_{0,1}]^T$, $T = \begin{bmatrix} 0_{(M-1) \times 1} & I_{M-1} \\ G_F & \end{bmatrix}$. In the definition of T and in (1), we marked the size of the matrices in question by pairs of indices of the form $x \times y$.

Applying the D transform similarly to the binary case [19] ($X(D) = \sum_{k=-\infty}^{+\infty} x^k \cdot D^k$) to Equations 1 and 2, we obtain

$$D^{-1} \cdot S(D) = U(D) \cdot G_T + S(D) \cdot T \quad (3)$$

$$U_0(D) = U(D) \cdot G_L + S(D) \cdot W \quad (4)$$

After some basic manipulations, it can be shown that

$$U_0(D) = \sum_{r=1}^R \frac{g_r(D)}{g_0(D)} \cdot U(D) \quad (5)$$

where $g_r(D) = \sum_{m=0}^M g_{m,r} \cdot D^m$, $0 \leq r \leq R$, are the generator polynomials. Because of the systematic nature of the involved encoders, (5) can be seen as the encoding relationship: $U_0(D) = G(D) \cdot U(D)$, where

$$G(D) = \begin{bmatrix} g_R(D) & \dots & g_1(D) \\ g_0(D) & \dots & g_0(D) \end{bmatrix} \quad (6)$$

This encoding matrix uniquely identifies a MNB convolutional encoder from the whole encoders family. For

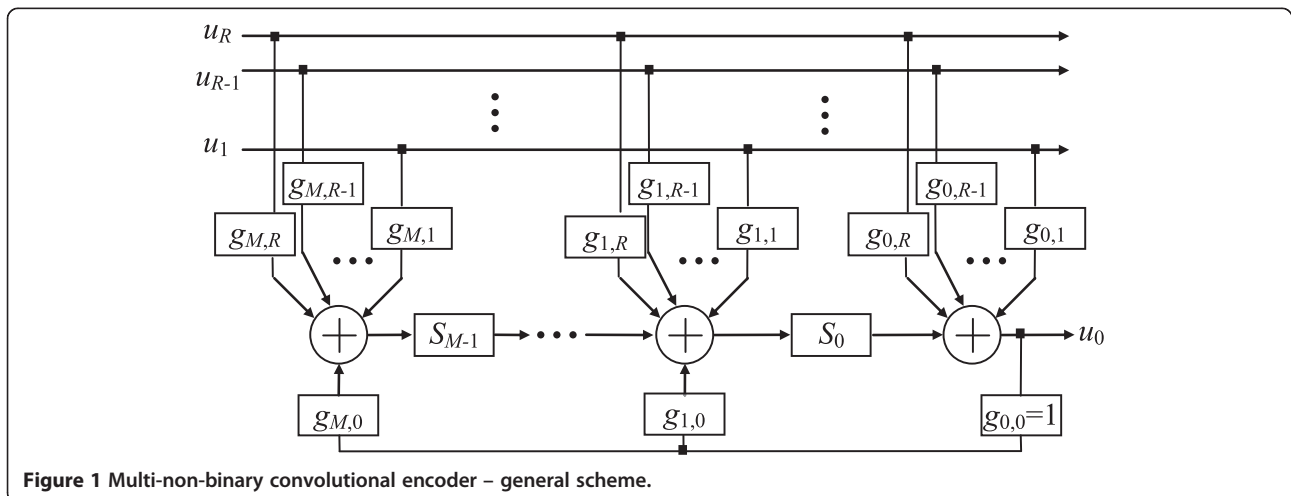


Figure 1 Multi-non-binary convolutional encoder – general scheme.

the sake of simplicity, we will use the form $G = [g_{m,r}]_{M \geq m \geq 0, R \geq r \geq 0}$ to refer to the encoding matrix. Hence, we have

$$G = \begin{bmatrix} G_0 & G_F^T \\ G_L^T & 1 \end{bmatrix} \quad (7)$$

2.2 The trellis termination for an MNBCE

In this section, we show how the already known techniques for trellis termination can be adapted to the MNB case: tail-biting [20] and zero padding [21,22]. Although the form of the equations is similar to the binary case, for the MNB case, all operations are performed in $GF(2^Q)$.

Starting from Equation 1 we can develop consecutively^a

$$\begin{aligned} s^N &= u^{N-1} \cdot G_T + s^{N-1} \cdot T = \\ &= u^{N-1} \cdot G_T + (u^{N-2} \cdot G_T + s^{N-2} \cdot T) \cdot T \\ &= \dots \dots = \sum_{n=0}^{N-1} u^n \cdot G_T \cdot T^{N-1-n} + s^0 \cdot T^N \end{aligned} \quad (8)$$

Thus, using the notation

$$s^u = \sum_{n=0}^{N-1} u^n \cdot G_T \cdot T^{N-1-n} \quad (9)$$

we have

$$s^N = s^u + s^0 \cdot T^N \quad (10)$$

The tail-biting technique involves a pre-encoding stage. The goal of pre-encoding is to determine s^u . Thus, starting with the null state $s^0 = 0$, it results that $s^u = s^N$. Actual coding starts with state $s^0 = s^x$ and finally reaches the same state $s^N = s^x$. We have $s^x = s^u + s^x \cdot T^N$, hence

$$s^x = s^u \cdot (I_M + T^N)^{-1} \quad (11)$$

provided that the matrix $(I_M + T^N)$ is non-singular. If N_f is the period of the polynomial $g_0(D)$ associated to the feedback, we have

$$T^{N_f} = I_M \quad (12)$$

Therefore, $(I_M + T^N)$ is non-singular and hence the tail-biting technique can be applied if N is not a multiple of the period N_f . The maximum possible period of a degree- M feedback polynomial with coefficients in $GF(2^Q)$ is

$$N_f = (2^Q)^M - 1 \quad (13)$$

In practice, the transition from s^N to s^x can be achieved using a look-up table with $2^{Q \cdot M}$ values.

The tail-biting technique has the advantage of not requiring the insertion of additional redundancy. Using tail-biting, the coding rate of the MNBCE in Figure 1 is

$$R_{cTB} = \frac{R}{R+1} \quad (14)$$

Zero padding can be achieved in two ways: interleaved termination or uninterleaved termination. Zero padding closes the trellis at null state^b at both ends: $s^{\text{end}} = s^0 = 0$. If $s^0 = 0$, (8) results in

$$s^N = s^u = \sum_{n=0}^{N-1} u^n \cdot G_T \cdot T^{N-1-n} \quad (15)$$

Zero padding with uninterleaved termination does not require pre-encoding. For zero padding with uninterleaved termination, K redundant symbols are added to the input block $[u^n]_{0 \leq n < N}$ so that

$$s^{\text{end}} = s^{N+K} = s^N \cdot T^K + \sum_{n=0}^{K-1} u^{N+n} \cdot G_T \cdot T^{K-1-n} = 0 \quad (16)$$

The matrix Equation 16 should be regarded as a system of M scalar equations on $GF(2^Q)$, where the unknowns are redundant symbols u^N, \dots, u^{N+K-1} with parameters the entries of s^N . For (16) to be a compatible system, uniquely determined, it is necessary and sufficient to insert $K = M$ redundant symbols.

Similar to the tail-biting case (Equation 11), Equation 16 also can be solved *a priori*, then using a look-up-table.

The coding rate for zero padding with uninterleaved termination is

$$R_{cZPU} = \frac{R \cdot N}{(R+1)(N+M)} \quad (17)$$

Zero padding with interleaved termination is done, dislodging M information symbols and attributing their places to redundant symbols. Assuming that the positions of redundant symbols are $J_{ZPI} = \{n_1, \dots, n_M\}$, we have

$$\begin{aligned} s^{\text{end}} = s^N &= \sum_{\substack{n=0 \\ n \notin J_{ZPI}}}^{N-1} u^n \cdot G_T \cdot T^{N-1-n} \\ &+ \sum_{n \in J_{ZPI}} u^n \cdot G_T \cdot T^{N-1-n} = 0 \end{aligned} \quad (18)$$

As for tail-biting, Equation 18 can be solved in two steps. By a pre-encoding stage, in which $(u^n)_{n \in J_{ZPI}}$ is set to 0, we obtain

$$s^u = \sum_{\substack{n=0 \\ n \notin J_{ZPI}}}^{N-1} u^n \cdot G_T \cdot T^{N-1-n} \quad (19)$$

Knowing s^u , the matrix Equation 18 is a system of M scalar equations with M unknowns that can be solved a

priori, and for which a look-up table can be used. One can note that the pre-encoding step can be eliminated if $J_{ZPI} = \{N-M, \dots, N-1\}$. In this case, the calculation of s^u , identifying the redundant symbols for the trellis termination and the encoding of these symbols using a look-up table are two phases of a single encoding.

The coding rate for zero padding with interleaved termination is

$$R_{cZPI} = \frac{R \cdot (N-M)}{(R+1) \cdot N} \quad (20)$$

2.3 Turbo encoding of an MNBTC

The left part of Figure 2 describes the structure of an MNBTC encoder. Input data block $u = [u_R \dots u_2 u_1]$, with $u_r = [u_r^0 \ u_r^1 \ \dots \ u_r^{N-1}]$, $1 \leq r \leq R$, is encoded by MNB encoder C1, providing the redundant symbols sequence $x_1 = [x_1^0 \ x_1^1 \ \dots \ x_1^{N-1}]$. At the same time, after being permuted by the block interleaver π , the input data sequence $\pi(u)$ is encoded by MNB encoder C0 which generates the second redundant symbols sequence $x_0 = [x_0^0 \ x_0^1 \ \dots \ x_0^{N-1}]$. So, the output of the turbo encoder is $x = [x_{R+1} \dots x_2 \ x_1 \ x_0]$, where $[x_{R+1} \dots x_2] = [u_R \dots u_2 \ u_1]$ is the systematic part of the output. In this way, the turbo coding rate is $R_c = R/(R+2)$. Of course, if a higher turbo coding rate is desired, puncturing can be performed.

Figure 3 shows the general structure of a block generated by an MNB turbo encoder at bit-wise in Figure 3a, and at symbol-wise in Figure 3b, respectively. All bits of

a block form the MNBTC word. Columns in both figures contain the bits/symbols generated at the turbo encoder output in the same moment of time (tact). The entire bit sequence contained in a column was named coded word (different from the MNBTC word). These coded words can be identified by the temporal indices: x^i , $0 \leq i < N$. The structure of bit-wise encoded words is visible in Figure 3a, of symbol-wise encoded words in Figure 3b. Also, Figure 3b shows that a coded word contains both information (symbols) and redundancy (symbols). The same block (turbo-coded word) can be seen/regrouped in Q -symbol sequences generated by the $R+2$ outputs of the MNBTC, denoted x_r , $0 \leq r \leq R+1$. Sequences x_2, x_3, \dots, x_{R+1} form the data block, and x_0 and x_1 form the redundancy generated by the MNBTC. Note that a branch through the trellis encoder C1 is given by the pair $[u^n \ x_1^n]$ and a branch in C0 encoder trellis is given by the pair $[\pi(u)^n \ x_0^n]$.

Trellis termination in MNBTC may be done primarily in four ways:

i) Dual tail-biting. Both encoders, C1 and C0, use the tail-biting technique to terminate their trellises. Turbo-coding rate in this case is

$$R_{cDTB} = \frac{R}{R+2} \quad (21)$$

ii) Semi-termination. C1 terminates its trellis according to the zero padding technique with interleaved termination with $J_{ZPI} = \{N-M, \dots, N-1\}$ while C0 does

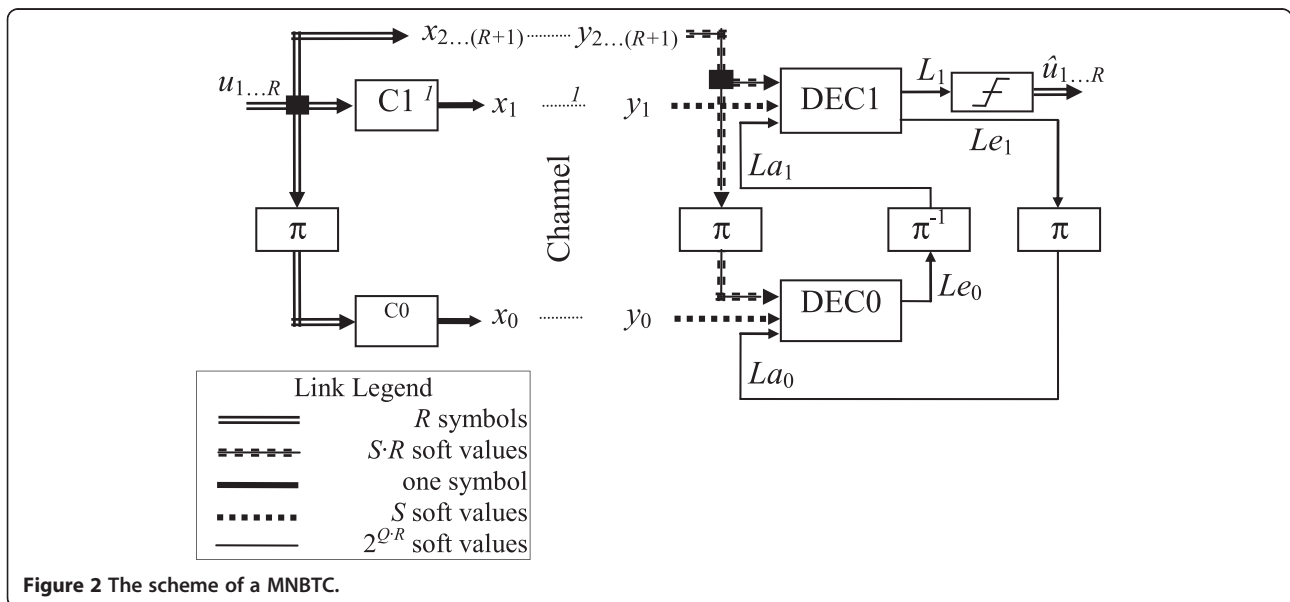


Figure 2 The scheme of a MNBTC.

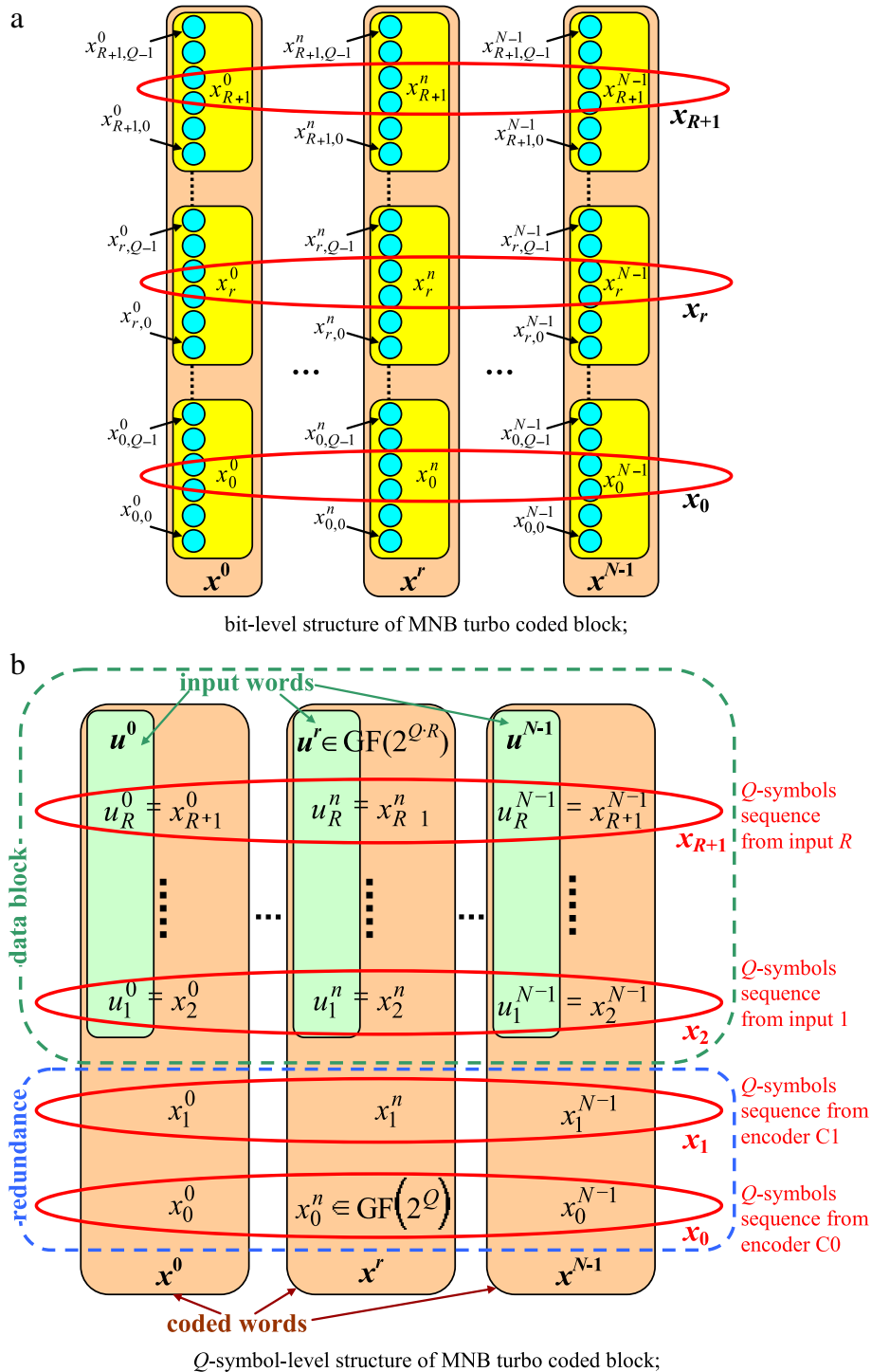


Figure 3 Structure of MNB turbo-coded block. (a) Bit-level structure of MNB turbo-coded block; **(b)** Q -symbol-level structure of MNB turbo-coded block.

not terminate the trellis. For the turbo-coding rate, the following value results

$$R_{cSCL} = \frac{R \cdot (N - M)}{(R + 2) \cdot N} \quad (22)$$

iii) Interleaved dual termination. Both encoders use the zero padding technique with interleaved termination. Positions of redundant symbols for C1 trellis termination can be chosen like in the previous case $J_{ZPI-1} = \{N - M, \dots, N - 1\}$, but for C0 they result from interleaving:

$J_{ZPI-2} = \{\pi(N-M), \dots, \pi(N-1)\}$. Thus, at least C0 performs a pre-coding stage. The coding rate is

$$R_{c\text{IDT}} = \frac{R \cdot (N - 2 \cdot M)}{(R + 2) \cdot N} \quad (23)$$

iv) Uninterleaved dual termination. Both encoders use the zero padding technique with uninterleaved termination. The coding rate is

$$R_{c\text{UDT}} = \frac{R \cdot N}{(R + 2) \cdot (N + M) + R \cdot N} \quad (24)$$

The uninterleaved dual termination technique is easy to accomplish, but subsequently raises the issue of restructuring the encoded block as sequences x_2, \dots, x_{R+1} contain $N + 2 \cdot M$ symbols, while sequences x_0 and x_1 only consist of $N + M$ symbols.

2.4 Modulation

For a 2^P -order modulation, if Q is a multiple of P , i.e., $Q = P \cdot B$, the Q bits of a turbo-coded Q -symbol can be grouped into B groups of P bits that can be transmitted in B -modulated P -symbols $[z_{r,B-1}^n \dots z_{r,1}^n z_{r,0}^n]$, $z_{r,b}^n \in GF(2^P)$. For the mapping between (the bits^c of) the coded word and modulated symbol, we use the notation

$$z_{r,b}^n = M([x_{r,b,P+P-1}^n \dots x_{r,b,P}^n]) \quad 0 \leq r \leq R + 1, 0 \leq b < B. \quad (25)$$

2.5 Turbo decoding of MNBTCs

In this section, we describe the possibilities for decoding MNBTCs.

2.5.1 Channel output and the decoding strategy

We will start by identifying the channel sequences arriving at the (turbo) decoder.

The sequences retrieved at the channel output, after demodulation, are denoted by $y = [y_{R+1} \dots y_2 y_1 y_0]$, with $y_r = [y_r^0 \ y_r^1 \ \dots \ y_r^{N-1}]$, $y_r^n = [y_{r,B-1}^n \ \dots \ y_{r,1}^n \ y_{r,0}^n]$, $0 \leq r \leq R + 1$, $0 \leq n < N$. Considering a Rayleigh flat fading channel^d, $y_{r,b}^n$ can be expressed as $y_{r,b}^n = a \cdot z_{r,b}^n + w_{r,b}^n$, $0 \leq b < B$, where $w_{r,b}^n$ is a sample of the additive white Gaussian noise (AWGN) and a is the fading amplitude. The decoding of these sequences can be performed bit-wise, symbol-wise, or word-wise. Of course, dedicated decoding algorithms can be used: maximum *a posteriori* (MAP) [23], logarithmic MAP (LogMAP) [24] or maximum logarithmic MAP (MaxLogMAP) [25,26]. Due to the known advantages of the MaxLogMAP algorithm with respect to practical implementation, we further present its adaptation to the MNB case. Let us note that although we adopted the strategy of the Max-Log-MAP algorithm, the changes needed to adapt MAP and Log-MAP algorithms to the MNB case are similar to the one shown.

2.5.2 Decoding over trellises of the component codes

To describe the operations performed by component decoders, we will first describe their operands.

At iteration i , each decoder DEC j processes the channel soft values $[y_{R+1} \dots y_2 y_1 y_0]$ and the *a priori* information La_j , $j = 0, 1$, to calculate the anch metrics (the gamma coefficients), the nodes metrics by forward recurrence (the alpha coefficients), the nodes metrics by backward recurrence (the beta coefficients), *a posteriori* information^e L , and extrinsic information Le .

The decoder tries to reconstruct the encoder's path through the trellis. In this respect, it assigns reliability values to each node and each branch in the trellis according to the values of received sequences, y_r^n . A trellis branch is identified by the triplet s^n , s^{n+1} , and u^n . However, the *possible* values for the variables s^n , s^{n+1} , and u^n respectively have a timeless character (do not depend on n or i). For possible values, we will use the notation: $\hat{\theta}$ for s^n , θ for s^{n+1} , and d for u^n . We will also denote by \tilde{x} the generic word obtained by encoding d , and with $\tilde{z}_{r,b}$, the generic symbol obtained by modulating the word \tilde{x} :

$$\tilde{z}_{r,b} = M([\tilde{x}_{r,b,P+P-1} \ \dots \ \tilde{x}_{r,b,P}]) \quad 0 \leq r \leq R + 1, 0 \leq b < B. \quad (26)$$

Obviously, there is interdependence between the current state $\hat{\theta}$, future state θ , and the value of the input word^f d : any two of them uniquely determine the third one. For instance, the current state $\hat{\theta}$ of the encoder and the input word value d determines the future state θ of the encoder. We will mark it through notation $(\hat{\theta}, d) \rightarrow \theta$. Given this interdependence property, the indexing of branches can be done by any two of these parameters. Therefore, the following branch metric notations are equivalent:

$$\gamma_j^{n,i}(\hat{\theta}, d) = \gamma_j^{n,i}(\hat{\theta} \cdot 2^{Q-R} + d) = \gamma_j^{n,i}(\hat{\theta}, \theta) \quad (27)$$

with $0 \leq \hat{\theta}, \theta < 2^{Q-M}$, $0 \leq d < 2^{Q-R}$, $j = 0, 1$.

Both branch metric γ and information processed during decoding (L , Le , or La) depend on how the input word d is *regarded* by the decoder. Thus we have

- word-wise decoding: d is a scalar in $GF(2^{Q-R})$;
- Q -symbol-wise decoding: d is a vector $[d_{R-1} \ \dots \ d_1 \ d_0]$ with entries in $GF(2^Q)$ (obviously, the scalar equivalent value of d remains the same in $GF(2^{Q-R})$;
- B -symbol-wise decoding: d_r from the previous expression of d is also a $[d_{r,B-1} \ \dots \ d_{r,1} \ d_{r,0}]$ vector with entries in $GF(2^P)$ with $P = Q/B$;
- bit-wise decoding. This case results as a degeneration of the previous case for $P = 1$ and $B = Q$. We have $d_r = [d_{r,Q-1} \ \dots \ d_{r,1} \ d_{r,0}]$.

2.5.3 Recurrences of component decoders

In this section, we present calculations performed by each component decoder in each iteration.

Regardless of the adopted decoding way (bit-/symbol-/word-wise), forward and backward recurrences have the same form:

$$\alpha_j^{n+1,i}(\theta) = \sum_{(\hat{\theta},d) \rightarrow \theta} \alpha_j^{n,i}(\hat{\theta}) \cdot \gamma_j^{n,i}(\hat{\theta}, d) \quad (28)$$

$$\beta_j^{n,i}(\hat{\theta}) = \sum_{\hat{\theta} \leftarrow (d,\theta)} \beta_j^{n+1,i}(\theta) \cdot \gamma_j^{n,i}(\hat{\theta}, d) \quad (29)$$

Branch metrics γ , *a posteriori* information L , and extrinsic information Le are calculated by the formulas

$$\gamma_j^{n,i}(\hat{\theta}, d) = Ex - Y_j^n - \frac{1}{\sigma^2} \sum_{b=0}^{B-1} (a \cdot \tilde{z}_{j,b} - y_{j,b}^n)^2 \Big|_{(\hat{\theta},d) \rightarrow \tilde{z}_{j,b}} \quad (30)$$

$$L_j^{n,i}(\delta) = \max_{(\hat{\theta},\theta) \rightarrow \delta} (\alpha_j^{n,i}(\hat{\theta}) + \gamma_j^{n,i}(\hat{\theta}, \theta) + \beta_j^{n+1,i}(\theta)) \quad (31)$$

$$Le_j^{n,i}(\delta) = \left(\alpha_j^{n,i}(\hat{\theta}) - \sum_{b=0}^{B-1} (a \cdot \tilde{z}_{j,b} - y_{j,b}^n)^2 \Big|_{(\hat{\theta},\theta) \rightarrow \tilde{z}_{j,b}} + \beta_j^{n+1,i}(\theta) \right) \Big|_{(\hat{\theta},\theta) \rightarrow \{L_j^{n,i}(\delta) = \max\}}$$

where σ^2 is the noise variance and

$$Y_1^n = \frac{1}{\sigma^2} \sum_{r=2}^{R+1} \sum_{b=0}^{B-1} (a \cdot \tilde{z}_{r,b} - y_{r,b}^n)^2 \quad (33)$$

$$Y_0^n = \frac{1}{\sigma^2} \sum_{r=2}^{R+1} \sum_{b=0}^{B-1} (a \cdot \tilde{z}_{r,b} - \pi(y_r^n)_b)^2$$

δ and Ex depend on the adopted decoding way:

a) Word-wise decoding: $\delta=d$ and $Ex = Le_j^{n,i}(d)$. The decoder calculates $N_{La} = 2^{Q \cdot R}$ values for L , Le , and La .

b) Q-symbol-wise decoding: $\delta = d_r$ and $Ex = \sum_{r=1}^R Le_j^{n,i}(d_r) \Big|_{(\hat{\theta},d) \rightarrow d_r}$. The decoder calculates $N_{Lb} = R \cdot 2^Q$ values for L , Le , and La .

c) B-symbol-wise decoding: $\delta=d_{r,b}$ and $Ex = \sum_{r=1}^R \sum_{b=0}^{B-1} Le_j^{n,i}(d_{r,b}) \Big|_{(\hat{\theta},d) \rightarrow d_{r,b}}$. The decoder calculates $N_{Lc} = R \cdot B \cdot 2^P$ values for L , Le and La .

d) Bit-wise decoding: $\delta=d_{r,q}$ and $Ex = \sum_{r=1}^R \sum_{q=0}^{Q-1} Le_j^{n,i}(d_{r,q}) \Big|_{(\hat{\theta},d) \rightarrow d_{r,q}}$. The decoder calculates $N_{Ld} = 2 \cdot R \cdot Q$ values for L , Le , and La .

We have $N_{Ld} \leq N_{Lc} \leq N_{Lb} \leq N_{La}$. Obviously, a higher value of N_L requires a higher computational effort, but also a decoder finer 'resolution'. In fact, this is the essence of the difference between SBTC, MBTC, NBTC, and MNBTC. A simple block data restructuring would not bring any advantage without *rethinking* of the decoding. Considering the decoding modes from d) to a), one can note a decoding optimization *shift* from bit-wise to symbol-wise and then word-wise. Word-wise decoding optimization accredits the MNBTC with a higher potential in decoder performance⁸. Other details that define the structure and operation of a MNBTC remain to be optimized. An example of this is interleaving.

2.5.4 Turbo decoding

Knowing the outputs of component decoders, we can specify the equations that govern turbo-decoding. At the turbo decoder level, we have the relations

$$L_0^{n,i}(\delta) = L_{a0}^{n,i}(\delta) - Y_0^n + L_{e0}^{n,i}(\delta) \quad (34)$$

$$L_1^{n,i}(\delta) = L_{a1}^{n,i}(\delta) - Y_1^n + L_{e1}^{n,i}(\delta)$$

where δ is defined above. Each decoder generates extrinsic information Le_{ej} that is transformed by interleaving in *a priori* information for the other component decoder

$$L_{a1}^i(\delta) = \pi^{-1}(L_{e0}^i(\delta)) \quad (35)$$

$$L_{a0}^i(\delta) = \pi(L_{e1}^{i-1}(\delta))$$

with $L_{e1}^0 \equiv 0$.

The iterative process continues until all iterations are performed or until a stop condition is fulfilled, if the turbo decoder is equipped with a mechanism to stop the iterations.

2.5.5 The hard decision

After performing all iterations, decoding is completed by a hard decision on *a posteriori* information values provided by one of the two component decoders. Thus, for word-wise decoding, we have

$$\hat{u}^n = \arg \max (L_1^{n,i_{\max}}(d)) \quad 0 \leq d < 2^Q \quad (36)$$

For other decoding methods, \hat{u}^n and d are replaced with \hat{u}_r^n and d_r , $0 \leq d_r < 2^Q$, $0 \leq r < R$, for Q-symbol-wise decoding;

- $\hat{u}_{r,b}^n$ and $d_{r,b}$, $0 \leq d_{r,b} < 2^P$, $0 \leq r < R$, $0 \leq b < B$, for B-symbol-wise decoding; and
 - $\hat{u}_{r,q}^n$ and $d_{r,q}$, $0 \leq d_{r,q} < 2$, $0 \leq r < R$, $0 \leq q < Q$, for bit-wise decoding.

2.6 Interleaving for MNBTCs

Similarly, to the interleavers used in MBTCs [5], for MNBTCs we also perform an intra-symbol interleaving step and an inter-symbol interleaving step. Inter-symbol

interleaving permutes the input words u^n in the information data sequence and may be done using already known interleaving techniques [27-29]. An advantage brought by MNBTCs is the essential reduction of the interleaver length N . The MNBTC data block structure is three-dimensional: $R \times Q \times N$. For SBTC, we have a linear data block structure ($R = Q = 1$) and for MBTC and NBTC planar structures ($R > Q = 1$ for MBTC and $Q > R = 1$ for NBTC) as well.

Interleaver size reduction implicitly leads to latency decrease. This is an advantage offered by MNBTCs. Latency is generally defined as the time from the first bit of a block input to the device to the time that the same bit is output from the device. Since this time is proportional to the length of the interleaver (not the size of the data block), a decrease in the latency results, proportional to the reduction of N , both the MNB encoder and the MNB decoder. Latency reduction is possible because both (MNB encoder and decoder) operate with symbols and not bits.

MNBTCs also constitute a compromise in complexity (placed) between MBTCs and NBTCs. Thus, considering the same data block of size $R \times Q \times N$ bits and the same memory M for all three (MBTC, NBTC, and MNBTC), in Table 1, we present the total number of ways from component encoders' trellises (MBCE, NBCE, and MNBCE). We considered that each encoder has an extra output (minimum redundancy). Then, the MNB encoder's trellis has $2^{(Q-1) \times (R+1) \times M} / Q$ more branches than the MB encoder's trellis and $2^{(R-1) \times Q \times M}$ less branches than the NB encoder's trellis. However, an accurate comparison is difficult, given that data block size and coding rate for the encoder in question cannot be aligned simultaneously. The fact however is that the MNB encoder has a low complexity compared to the NB encoder due to a lower number of bits per symbol.

Table 2 compares the volume of the calculation made by the convolutional decoder for SB, MB, and MNB. For each case, we present details about the calculation of the recurrent coefficients α and β , of the coefficients γ (trellis branch metrics), and the values of the coefficients L . 'No.' is the total number of coefficients calculated. To calculate each coefficient, we require a number of 'prod' multiplications and a number of 'sum' additions or 'cmp' comparisons. For a *fair* analysis, it is necessary to use the same number of bits of information, the same

encoding rate and the same constraint length (binary equivalent) for all three cases. The encoding rate is adjustable through puncturing. But puncturing does not change the 'volume' of decoding calculation. Thus, for comparison, we considered the natural rate of encoding in each case. In order to preserve the size of the input data block, we considered that a data block has the same number of bits ($R \times Q \times N$) in all three cases.

'Preserving' the constraint length (binary equivalent) requires that between M' (SB and MB encoders' memory) and M (MNB encoder's memory) there be the formula: $M' = Q \times M$.

As expected, the volume of the calculation for the MNB decoder is significantly higher than for SB and MB. However, a comparison between computational efforts made in turbo-decoding for the three cases must take into account the convergence of the iterative process (see Section 3.2, Equation 38).

By imposing the 'preservation' of the number of bits in the data block results an advantage for MNB: reduced latency. Thus, the numbers of recurrent steps necessary to be taken in the SB, MB, or MNB decoders are $R \times Q \times N + 1$, $Q \times N + 1$, and $N + 1$, respectively. If the other theoretical calculations can be parallelized, recurrences require successive calculations. Thus, latency becomes proportional to the number of recurrent steps and, of course, there is a reduction in the case of MNB.

Intra-symbol interleaving, used in the DVB-RCS and WiMAX standards, operates on the $R = 2$ bits ($Q = 1$) of each input symbol, reversing their positions for even indexes n . Since for MNBTCs, an input word consists of at least 4 bits, intra-symbol interleaving can be made in many more ways. A study of these modes is beyond the scope of this paper. The intra-symbol permutation implemented in the simulation results presented in Section 3 was not specifically optimized: we performed *helical* intra-symbol interleaving, i.e., the $R \times Q$ bits of the u^n word are cyclically permuted by one position in the same direction.

3 An example of MNBTC

This section presents an instance of MNBTC offering a good trade-off between performance and complexity. It is the outcome of a search among the MNB component encoders family with $R = 2$ (double-binary input), $Q = 2$, and memory $M = 3$. The next sections provide some

Table 1 Components of MB, NB and MNB encoders' trellises

Type of encoder	Interleaver length	Coding rate R_c	Bits/symbol	Memory	States/tact	Branches/tact	Total trellis branches
MB	$Q \times N$	$R/(R+1)$	1	M	2^M	$2^{(R+1) \times M}$	$Q \times N \times 2^{(R+1)M}$
NB	N	1/2	$R \times Q$	M	$(2^R \times Q)^M$	$(2^R \times Q)^{2 \times M}$	$N \times (2^R \times Q)^{2 \times M}$
MNB	N	$R/(R+1)$	Q	M	$(2^Q)^M$	$(2^Q)^{(R+1) \times M}$	$N \times (2^Q)^{(R+1)M}$

Table 2 Complexity of the Max-Log-MAP decoding algorithm

		SB	MB	MNB
a, β	No.	$(R \times Q \times N + 1) \times 2^M$	$(Q \times N + 1) \times 2^M$	$(N + 1) \times 2^{Q \times M}$
	Prod	2	2^R	$2^{Q \times R}$
	Sum	1	$2^R - 1$	$2^{Q \times R} - 1$
γ	No.	$R \times Q \times N \times 2^{M \cdot 2}$	$Q \times N \times 2^{M \cdot 2^R}$	$N \times 2^{Q \times M} \times 2^{Q \times R}$
	Prod	2	$R + 1$	$(R + 1) \times 2^Q$
	Sum	2	$R + 1$	$(R + 1) \times 2^Q$
L	No.	$R \times Q \times N \times 2$	$Q \times N \times 2^R$	$N \times 2^{Q \times R}$
	Prod	2	2×2^M	$2 \times 2^{Q \times M}$
	Cmp	1	$2^M - 1$	$2^{Q \times M} - 1$

Comparison between SB, MB, and MNB.

details on the MNB component encoder family's cardinality and on the selection criteria. Then the BER/FER performance of the selected MNBTC is compared with the memory-4 DBTC used in the DVB-RCS2 standard. We chose this TC as benchmark, since it is the most powerful standardized TC with a $\frac{1}{2}$ natural rate. But an absolute 'fair' comparison should be made between convolutional codes with the same 'binary equivalent constraint length'.

3.1 Cardinality of the two double-binary inputs MNB convolutional encoder family

A component of this family is identified by the following coding matrix:

$$G = [g_{32}g_{31}g_{30}, g_{22}g_{21}g_{20}, g_{12}g_{11}g_{10}, g_{02}g_{01}g_{00}], \quad (37)$$

where $g_{m,r} \in GF(4)$. The cardinality of the set of matrices in the form given by (10), verifying $g_{3,2}g_{3,1}g_{3,0} \neq 0$ is $N_{G_{3 \times 2 \times 2}} = 4^{11} - 4^8$, i.e. over 4 million. Comparatively, the number of memory-4 convolutional encoders amounts to $N_{G_{4 \times 1 \times 1}} = 2^9 - 2^7$ for single-binary codes and $N_{G_{4 \times 2 \times 1}} = 2^{14} - 2^{11}$ for double-binary codes. It is desirable to reduce the selection base by eliminating a priori poor performance matrices. Thus, we investigated only matrices with $g_0 = [g_{3,0}g_{2,0}g_{1,0} \ 1]$ as primitive polynomial (irreducible with maximum period 63). These are $[2 \ 1 \ 1 \ 1]$, $[2 \ 1 \ 3 \ 1]$, $[2 \ 2 \ 2 \ 1]$, $[2 \ 2 \ 3 \ 1]$, $[2 \ 3 \ 1 \ 1]$, $[2 \ 3 \ 2 \ 1]$, $[3 \ 1 \ 1 \ 1]$, $[3 \ 1 \ 2 \ 1]$, $[3 \ 2 \ 1 \ 1]$, $[3 \ 2 \ 3 \ 1]$, $[3 \ 3 \ 2 \ 1]$, and $[3 \ 3 \ 3 \ 1]$.

3.2 Search for the best MNBCE

Through practical observations, we found that an efficient turbo decoder in terms of BER/FER converges also rapidly, meaning that on average, it performs few iterations to correct a block. Thus, if the TC is equipped with a criterion for stopping the iterations, its convergence is defined as the ratio of the total number of iterations, iter

(Nb) necessary for the decoding of Nb turbo-coded blocks and Nb , when Nb is large

$$C_v = \lim_{Nb \rightarrow \infty} \frac{iter(Nb)}{Nb}. \quad (38)$$

For small SNR values, convergence C_v tends to $iter_{max}$, the maximum iteration number, which is set *a priori*. When the SNR increases, the weight of the non-converging blocks decreases.

The motivation of using convergence criterion is that C_v can be calculated with a quite good accuracy for small values of Nb . So, the turbo code functionality was simulated with each of the component codes. The simulations were stopped if the convergence C_v overcame a given limit

$$stop \ the \ simulation \ if \ C_v(Nb) > C_{v0} - Nb \cdot dC_v, \quad (39)$$

where dC_v represents the speed of threshold decreasing. The given threshold $C_{v0} - Nb \cdot dC_v$ decreases with Nb in order to limit the simulation duration for a specific code. In the search we made, the values of the parameters C_{v0} and dC_v were chosen equal to 4.5 and 15×10^{-4} respectively. For example, for a TC with a convergence $C_{v1} = 3$ iterations/block, $N_{b1} = 10^3$ blocks are simulated. The lower the convergence of a TC, the more blocks have to be simulated, which leads to a more precise estimate of its convergence.

We used the QPP inter-symbol interleaver set forth in the LTE standard for the length of 376. Thus, the data block size is 4×376 bits. For the most efficient component encoders in terms of convergence, we performed simulations in order to estimate BER and FER. Note that we obtained many component encoders with very close BER/FER performance.

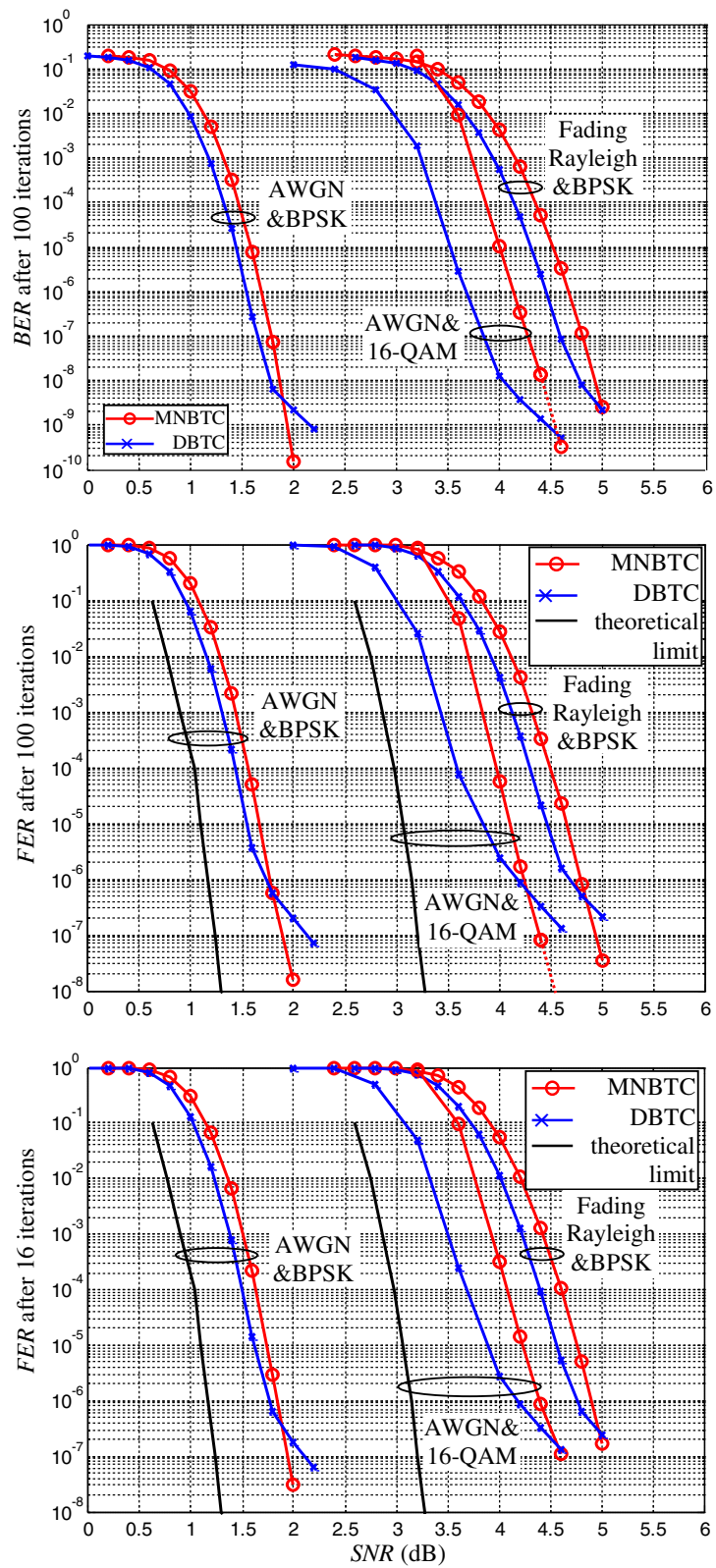


Figure 4 BER/FER versus SNR performance comparison. Between the MNBTC with component encoder defined by (40) and the memory-4 DBTC defined in the DVB-RCS2 standard [3].

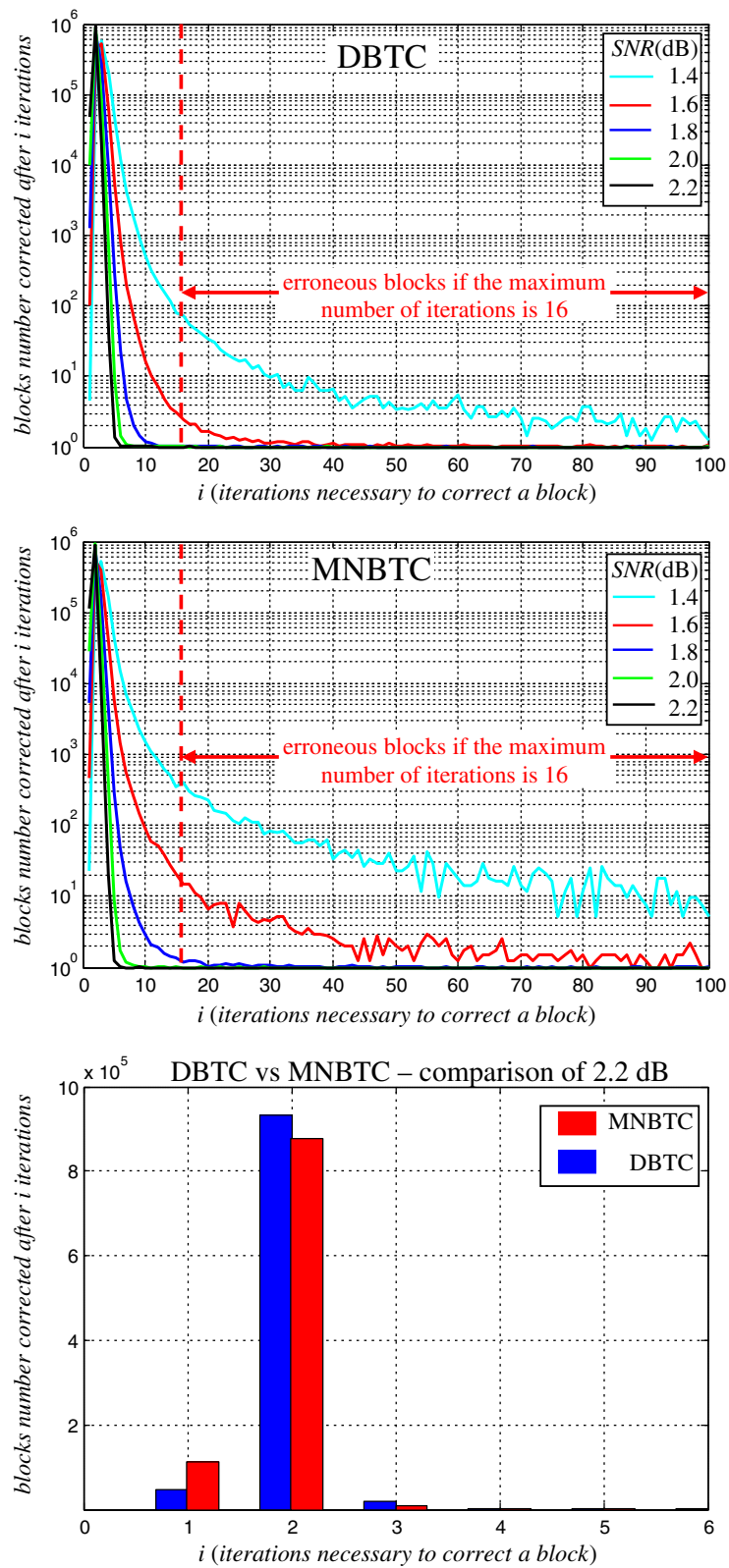


Figure 5 Histograms of 10^6 performed blocks over necessary iterations to correct entire block. In the case of AWGN transmission channel and BPSK modulation.

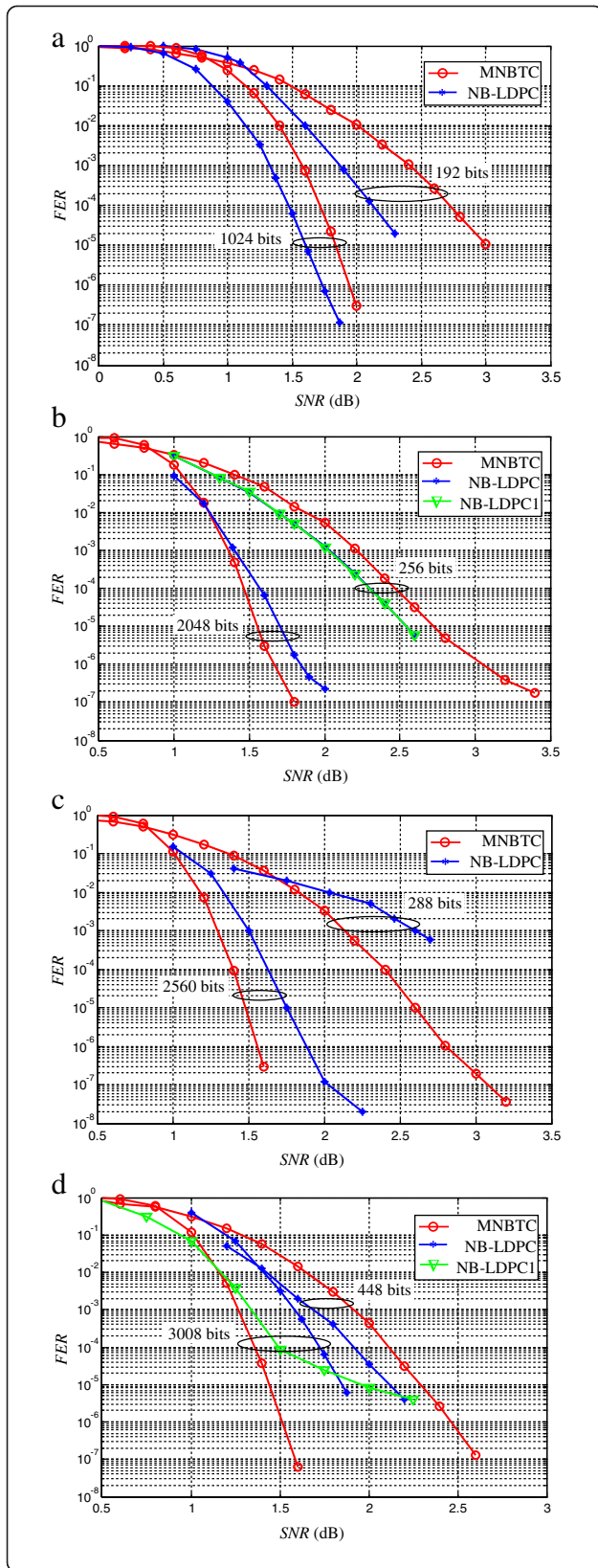


Figure 6 FER performance comparison between MNBTC and NB-LDPC for different lengths of the data block. (a) NB-LDPC curves were taken from Figure three “EMS” in [32] for 192 bits and Figure eleven ‘Punctured C1 R = 1/2’ in [33] for 1,024 bits; **(b)** NB-LDPC curves were taken from Figure three (c) ‘GF(256) Protograph LDPC Code’ in [34] - NB-LDPC, Figure two ‘LDPC GF(256)’ in [35] - NB-LDPC1 for 256 bits and Figure three ‘Hybrid LDPC 2 N = 2,048’ in [36] for 2,048 bits; **(c)** NB-LDPC curves were taken from Figure thirteen ‘BP floating point’ in [37] for 288 bits and Figure three ‘Multi-edge type LDPC N = 2,560’ in [36] for 2,560 bits; **(d)** NB-LDPC curves were taken from Figure three (d) ‘GF(256) Protograph LDPC Code, R = 1/2’ in [34] for 448 bits and Figure three ‘irregular GF(8) code’ in [38] - NB-LDPC, Figure three ‘hybrid G(8) – G(2) code’ in [38] - NB-LDPC1 for 3,008 bits.

3.3 Simulations results

The method described above led to the following generator matrix:

$$G_{\text{sample}} = [322, 111, 221, 111] \quad (40)$$

The MNBTC performance built with this component encoder is compared with the one of the benchmark DBTC from DVB-RCS2 in Figure 4. For a fair comparison, we used data blocks with sizes of $R \times Q \times N = 2 \times 2 \times 376$ bits for MNBTC and $2 \times 1 \times 752$ bits for DBTC. Also, for both TCs, we used tail-biting and the Max-Log-MAP decoding algorithm with the adaptation previously described for MNBs (word-wise decoding). To plot the curves, we performed simulations until 500 erroneous blocks were obtained or until a maximum number of blocks equal to 10^9 for DBTC and around 2×10^8 for MNBTC were simulated. Exceptions are simulations to determine the extreme points of the curves ‘AWGN&BPSK’ for MNBTC. For these points, simulations ran until we obtained 30 erroneous blocks (over 2.1×10^9 blocks transmitted). The maximum number of iterations is set to 16 or 100, combined with the genie stopping criterion [30]. The theoretical limit mentioned in Figure 4 is the sphere packing bound that takes the penalty due to the block size into account [31]. For the 16-QAM modulation, to perform the mapping between coded symbols and modulation symbols, in the MNBTC case, we used the values $B = Q = 2$, $P = 1$, and the relation

$$z_{r,b}^n = 2 \cdot x_{r,b}^n - 3, \quad (41)$$

where $x_{r,b}^n \in \text{GF}(4) = \{0, 1, 2, 3\}$, $0 \leq r < R + 1$, $0 \leq b < 2$, and $0 \leq n < N - 1$.

For the DBTC case, we used the relations

$$\begin{cases} z_1^n = (2 \cdot x_3^n - 1) \cdot (2 \cdot x_1^n + 1) \\ z_0^n = (2 \cdot x_2^n - 1) \cdot (2 \cdot x_0^n + 1) \end{cases}, x_r^n \in \text{GF}(2) = \{0, 1\}, 0 \leq n < N - 1, \quad (42)$$

where $[x_3^n \ x_2^n \ x_1^n \ x_0^n]$ is the turbo decoder output at time n . We do not use puncturing, nor quantization nor

bit interleaving between code and modulation for the Rayleigh case, since BPSK is assumed in this case.

Whatever type of channel involved and the type of modulation used, as argued by the curves in Figure 4, the MNBTC does not show the error floor effect in the value range for FER investigated. The MNBTC performance in the waterfall region is slightly weaker than for the DBTC: the observed convergence loss ranges from 0.1 to 0.4 dB, but that the error floor is much lower in all the simulated cases.

An advantage of MNBTCs over DBTCs is the continuous improvement of BER/FER performance when performing an average number of iterations greater than usual for TCs. Figure 4 shows that the FER performance of the DBTC does not improve from 16 to 100 iterations, whereas for the MNBTC, the improvement is clear. This is an interesting property for applications where the decoding latency/throughput is not an issue. Some additional details on this behavior are given in Figure 5. At a SNR = 1.4 dB, the histogram of blocks as a function of the number of iterations required for correction presents a much higher dispersion in MNBTCs than DBTCs. While the SNR increases, the histograms for MNBTCs becomes 'finer', and have become more compact than DBTC histograms at SNRs greater than 2 dB. For SNR > 2 dB, MNBTC shows better convergence than DBTC. This is argued by the third diagram in Figure 5.

In Figure 6, we compared the performance of the proposed MNBTC with that of the NB-LDPCs presented recently in the literature, [32-38]. In the simulations carried out for the construction of the curves for the MNBTCs in Figure 6, we used the stop criteria described in [39], with a threshold value of 20. The results in this figure clearly lead to the conclusion that the proposed MNBTC is inferior in terms of performance to the NB-LDPC in small size data block (under a thousand bits), but becomes superior at larger sizes (thousands of bits).

4 Conclusion

This work opens a new horizon for TCs: using of the multi-non-binary convolutional encoders by generalizing the concepts of multi-binary and non-binary codes. The sample of MNBTC presented in this paper successfully competes in performance with the DBTC recently standardized in DVB-RCS2. The main advantages (as for example latency reduction) of MNBTCs arise from the more compact form of data blocks. Many aspects still remain to be explored in this horizon, e.g., an optimization for intra-symbol interleaving. We intend also to investigate new solutions for decoding MNBTCs. The MAP adaptation algorithm, described above, is a rather complex solution. Due to the non-binary nature, it

is possible to find solutions less complicated, possibly by adapting the algorithms used in (turbo) decoding of the Reed-Solomon codes. Such solutions even if they are less effective than MAP type algorithms can be alternatives.

Regarding the iteration stopping criteria, stopping mechanisms dedicated to SBTCs abound in the literature [30,39-42]. A few studies on the stopping criteria for MBTCs can also be found [39,43]. In general, they can be adapted to the MNB case. Although a detailed study on this topic is beyond the scope of this paper, we have implemented the stopping criterion called minAPP in [39] for the MNB case. Performance in terms of BER/FER is similar to the application of the genie criterion [30]. However, the average number of iterations performed for decoding a block depends on the selected stopping technique. Thus, an MNB turbo-decoder equipped with a mechanism to stop the iterations based on the minAPP criterion (easily implementable in practice) performs on average 1.5 more iterations than the one based on the genie criterion (not applicable in practice).

The investigation of the possibilities of using the MNBTC in cooperative communications schemes, [44-46], is also of interest.

Endnotes

^aTo avoid confusion, let us point out that the matrix T is the only variable throughout the paper that appears exponentiated, i.e., for which the top right indexes are exponents.

^bObviously, the trellis can be closed in any other state, but this does not bring benefits.

^cEven if working with symbols and words, they exist physically still under binary form.

^dBy 'flat' we considered a frequency nonselective fading, whose amplitude remains constant for the duration of a transmitted symbol (which is P times greater than the length of a bit).

^eSince in the Max-Log-MAP algorithm operations are performed in the logarithmic domain, operands have an information dimension and not a probability one.

^fThe input word uniquely identifies a branch between all outgoing (or incoming) in a node.

^gOptimal decoding algorithm uses *a posteriori* probabilities for the entire sequence of data transmitted.

Abbreviations

APP: *A posteriori* probability; AWGN: additive white Gaussian noise; BER: bit error rate; BPSK: binary phase-shift keying; DBTC: Double binary turbo codes; DVB-RCS: Digital video broadcasting-return channel via satellite; FER: Frame error rate; GF: Galois field; Log-MAP: Logarithmic MAP; LTE: Long-term evolution; MAP: Maximum *a posteriori* probability; Max-Log-MAP: Maximum logarithmic MAP; MBTC: Multi-binary turbo codes; MNB: Multi-non-binary; MNBCE: Multi-non-binary convolutional encoder; MNBTC: Multi-non-binary turbo codes; NB-LDPC: Non-binary low-density parity check codes;

NBTC: Non-binary turbo codes; PCCC: Parallel concatenated convolutional codes; QAM: Quadrature amplitude modulation; QPP: Quadratic permutation polynomial; SB: Single binary; SBT: Single binary turbo codes; SNR: Signal-to-noise ratio; TC: turbo codes; WiMAX: worldwide interoperability for microwave access.

Competing interests

The authors declare that they have no competing interests.

Acknowledgements

This paper was partially supported by the project "Development and support for multidisciplinary postdoctoral programs in primordial technical areas of the national strategy for research - development - innovation" 4D-POSTDOC, contract nr. POSDRU/89/1.5/S/52603, project co-funded from the European Social Fund through the Sectorial Operational Program Human Resources 2007–2013, and was partially supported by a grant of the Romanian Ministry of Education, CNCS – UEFISCDI, project number PN-II-RU-PD-2012-3-0122.

Author details

¹Faculty of Electronics and Telecommunications, University POLITEHNICA of Timisoara, V. Parvan 2, 300223, Timișoara, Romania. ²Faculty of Electrical Engineering, Electronics and Information Technology, Valahia University of Targoviste, 2 Avenue King Carol I, Targoviste 130024, Romania. ³Télécom Bretagne, 292338, Brest Cedex 3, France.

Received: 18 March 2013 Accepted: 19 November 2013

Published: 5 December 2013

References

1. C Berrou, A Glavieux, P Thitimajshima, Near Shannon limit error-correcting coding and decoding: turbo-codes. In Proceedings of ICC, Geneva **23–26**, 1064–1070 (May, 1993)
2. ETSI, 3GPP TS 36.212, *Evolved universal terrestrial radio access (E-UTRA)*, 2010. http://www.etsi.org/deliver/etsi_ts/136200_136299/136212/08.08.00_60/ts_136212v080800p.pdf (January 2010)
3. DVB, European Telecommunications Standards Institute, *DVB Interactive Satellite System, Part 2 (Lower Layers for Satellite standard, DVB document A155-2, 2011)*. http://www.dvb.org/resources/public/standards/a155-2_DVB-RCS2_Lower_Layers.pdf (March 2011)
4. Consultative Committee for Space Data Systems, *Recommendation for space data system standards, synchronization and channel coding*, 2011. <http://public.ccsds.org/publications/archive/131x0b2ec1.pdf> (August 2011)
5. C Douillard, C Berrou, Turbo codes with rate- $m/(m+1)$ constituent convolutional codes. *IEEE Transactions on Communications* **53**(10), 1630–1638 (2005)
6. M Ferrari, S Bellini, Rate variable multi-binary turbo codes with controlled error-floor. *IEEE Transactions on Communications* **57**(5), 1209–1214 (2009)
7. J Berkmann, On turbo decoding of nonbinary codes. *IEEE Communications Letters* **2**(4), 94–96 (1998)
8. AC Reid, TA Gulliver, DP Taylor, Rate-1/2 component codes for nonbinary turbo codes. *IEEE Transactions on Communications* **53**(9), 1417–1422 (2005)
9. G Liva, S Scalise, E Paolini, M Chiani, *Turbo codes based on time-variant memory-1 convolutional codes over F_q* . In *IEEE ICC 2011, Kyoto, 5–9, June 2011*
10. H Balta, M Kovaci, A de Baynast, C Vlădeanu, R Lucaciu, *Multi-Non-Binary Turbo-Codes From Convolutional to Reed-Solomon Codes*, In *Scientific Bulletin of Politehnica University of Timisoara, Transactions on Electronics and Communications, Tom 51–65 Timisoara (Romania, Sept, 2006)*, pp. 113–118
11. DJC Mackay, RM Neal, Good codes based on very sparse matrices, in *Cryptography and Coding, 5th IMA Conference, Lecture Notes in Computer Science number 1025*, ed. by B Colin (Springer, Berlin, Germany, 1995), pp. 100–111
12. MC Davey, DJC Mackay, Low-density parity check codes over GF(q). *IEEE Communications Letters* **2**(6), 165–167 (1998)
13. IB Djordjevic, B Vasic, Nonbinary LDPC codes for optical communication systems. *IEEE Photonics Technology Letters* **17**(10), 2224–2226 (2005)
14. G Bosco, R Garelo, F Mininni, M Baldi, F Chiaraluca, Non-binary low density parity check codes for satellite communications. In Proceedings of the ISCC, Cagliari, Italy **26–29**, 1019–1024 (June, 2006)
15. G Cocco, S Pfletschinger, M Navarro, C Ibars, *Opportunistic adaptive transmission for network coding using nonbinary LDPC codes* (*EURASIP Journal on Wireless Communications and Networking* (2010), 2010). Doi: 10.1155/2010/517921
16. A Mourad, O Picchi, I Gutierrez, M Luise, Low complexity soft demapping for non-binary LDPC codes. *EURASIP Journal on Wireless Communications and Networking* **2012**, 55 (2012)
17. A Marinoni, P Savazzi, P Gamba, Efficient detection and decoding of q-ary LDPC coded signals over partial response channels. *EURASIP Journal on Wireless Communications and Networking* **2013**, 18 (2013)
18. T Kailath, *Linear Systems* (Prentice-Hall, Upper Saddle River, 1980)
19. R Johannesson, KS Zigangirov, *Fundamentals of Convolutional Coding* (IEEE Press, Fundamentals of Convolutional Coding, 1999)
20. C Weiss, C Bettstetter, S Riedel, DJ Costello, *Turbo decoding with tailbiting trellises* In Proc (IEEE Int. Symp. Signals, Syst, Electron, Pisa, Italy, 1998), pp. 343–348
21. P Guinand, J Lodge, *Trellis Termination for Turbo Encoders* In Proceedings of the 17th Biennial Symposium on Communications Queen's University (Kingston, ON, Canada, May 1994), pp. 389–392
22. J Hokfelt, O Edfors, T Maseng, On the theory and performance of trellis termination methods for turbo codes. *IEEE Journal on Selected Areas in Communications* **19**(5), 838–847 (2001)
23. LR Bahl, J Cocke, F Jelinek, J Raviv, Optimal decoding of linear codes for minimising symbol error rate. *IEEE Transactions on Information Theory* **20**(2), 284–287 (1974)
24. P Robertson, P Hoeher, E Villebrun, Optimal and suboptimal maximum a posteriori algorithms suitable for turbo decoding. *European Trans. Telecommun.* **8**(2), 119–125 (1997)
25. W Koch, A Baier, *Optimum and sub-optimum detection of coded data disturbed by time-varying intersymbol interference*, In *GLOBECOM'90* (San Diego, CA, USA, December 1990), pp. 1679–1684
26. J Vogt, A Finger, Improving the max-log-MAP turbo decoder. *Electronics Letters* **36**(23), 1937–1939 (2000)
27. SN Crozie, *New high-spread high-distance interleavers for turbo-codes* (20th Biennial Symp, Commun, Kingston, Canada, 2000), pp. 3–7
28. C Berrou, Y Saouter, C Douillard, S Kerouédan, M Jézéquel, Designing good permutations for turbo codes: towards a single model. In *IEEE International Conference on Communications, Paris, France 1*, 341–345 (2004)
29. QJ Sun, OY Takeshita, Interleavers for turbo codes using permutation polynomials over integer rings. *IEEE Trans. Inform. Theory* **51**(1), 101–119 (2005)
30. A Matache, S Dolinar, F Pollara, *Stopping rules for turbo decoders* (TMO Progress Report 42–142, Jet Propulsion Laboratory, Pasadena, California, 2000)
31. RG Gallager, *Information Theory and Reliable Communication* (Wiley, New York, 1968), pp. 5–6
32. E Boutillon, L Conde-Canencia, Bubble check: a simplified algorithm for elementary check node processing in extended min-sum non-binary LDPC decoders. *Electronic Letter* **46**(9), 633–634 (2010)
33. K Kasai, D Declercq, C Poulliat, K Sakaniwa, Multiplicatively repeated non-binary LDPC codes. *IEEE Transactions on Information Theory* **57**(10), 6788–6795 (2011)
34. L Costantini, B Matuz, G Liva, E Paolini, M Chiani, Non-binary protograph low-density parity-check codes for space communications. *Int. J. Satell. Commun. Network* **30**, 43–51 (2012)
35. L Costantini, B Matuz, G Liva, E Paolini, M Chiani, On the Performance of Moderate-Length Non-Binary LDPC Codes for Space Communications, In the *5th Advanced Satellite Multimedia Systems Conference and the 11th Signal Processing for Space Communications Workshop*. Cagliari **13–15**, 122–126 (Sept. 2010)
36. L Sassatelli, D Declercq, Non-binary Hybrid LDPC Codes. *IEEE Transactions on Information Theory* **56**(10), 5314–5334 (2010)
37. E Boutillon, L Conde-Canencia, AA Ghouwayel, Design of a GF(64)-LDPC Decoder Based on the EMS Algorithm. *IEEE Transactions on Circuits and Systems I* **60**(10), 2644–2656 (2013)
38. L Sassatelli, D Declercq, Non-binary Hybrid LDPC Codes: structure, decoding and optimization. *IEEE Information Theory Workshop, ITW'06 (Chengdu, 2006)*, 71–75 (2006). doi: 10.1109/ITW2.2006.323759
39. H Balta, C Douillard, M Kovaci, *"The Minimum Likelihood APP Based Early Stopping Criterion for Multi-Binary Turbo Codes"*, In *Scientific Bulletin of Politehnica University of Timisoara, Transactions on Electronics and Communications, Tom 51–65 (Timisoara, Romania, Sept. 2006)*, pp. 199–203
40. J Hagenauer, E Offer, L Papke, Iterative decoding of binary block and convolutional codes. *IEEE Transactions on Information Theory* **42**(2), 429–445 (1996)
41. RY Shao, S Lin, MPC Fossorier, Two simple stopping criteria for turbo decoding. *IEEE Transactions on Communications* **47**(8), 1117–1120 (1999)

42. F Zhai, IJ Fair, Techniques for early stopping and error detection in turbo decoding. *IEEE Transactions on Communications* **51**(10), 1617–1623 (2003)
43. L Guerrieri, D Veronesi, P Bisaglia, *Stopping rules for duo-binary turbo codes and application to HomePlug AV* (In IEEE Global Telecommunications Conference, New Orleans, 2008), pp. 1–5. doi: 10.1109/GLOCOM.2008.ECP.558
44. H Van Khuong, T Le-Ngoc, A cooperative turbo coding scheme for wireless fading channels. *IET Communications* **3**(10), 1606–1615 (2009)
45. V Bota, P ZsA, A Silva, S Teodoro, MP Stef, A Moco, A Botos, A Gameiro, Combined distributed turbo coding and space frequency block coding techniques. *EURASIP Journal on Wireless Communications and Networking* (2010). doi: 10.1155/2010/327041
46. F Babich, A Crismani, Cooperative coding schemes: design and performance evaluation. *IEEE Transactions on Wireless Communications* **11**(1), 222–235 (2012)

doi:10.1186/1687-1499-2013-279

Cite this article as: Balta et al.: Multi-non-binary turbo codes. *EURASIP Journal on Wireless Communications and Networking* 2013 **2013**:279.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- ▶ Convenient online submission
- ▶ Rigorous peer review
- ▶ Immediate publication on acceptance
- ▶ Open access: articles freely available online
- ▶ High visibility within the field
- ▶ Retaining the copyright to your article

Submit your next manuscript at ▶ springeropen.com
