Buletinul Ştiinţific al Universităţii "Politehnica" din Timişoara

Seria ELECTRONICĂ şi TELECOMUNICAŢII
TRANSACTIONS on ELECTRONICS and COMMUNICATIONS

Tom 51(65), Fascicola 1-2, 2006

A Very General Family of Turbo-Codes: The Multi-Non-Binary Turbo-Codes

Horia Balta[1], Maria Kovaci[1], Alexandre de Baynast[2], Calin Vladeanu[3], Radu Lucaciu[1]

Abstract – This paper presents a new family of turbo codes whose the constituent codes have $R \geq 1$ non-binary inputs and $R+1$ outputs. We refer this family as the multi input non-binary turbo codes (MNBTC), which is very general. More specifically, we show that this family includes the multi-binary turbo-codes (MBTCs) that themselves include the classical binary turbo-codes (BTCs). Moreover, it also includes the turbo-codes with Reed Solomon codes as constituent codes. In this paper, we fully describe the encoding process and the extension of the Maximum A Posteriori (MAP) decoding algorithm, especially the trellis closing issues for these codes. Additionally, we show by simulations the benefit of using this family of Turbo-codes.

Keywords: turbo-code, MAP algorithm, multi non-binary convolutional code

I. INTRODUCTION

The discovery of the turbo codes (TCs) [1] represents a major breakthrough in the coding theory since the asymptotic performance of the TCs close the gap to the Shannon limit within tenths of decibels. A sensitive component of the TCs is the interleaver. Several interleavers have been recently proposed: the S-interleaver [2], Takeshita-Costello interleaver [3] and ENST interleaver [4] just to name a few. Moreover, different designs and decoding algorithms for TCs have been investigated: the technique of the circular codes [5], the serial concatenation [6], the decoding algorithms: SOVA [7], MaxLogMAP and LogMAP [8].

The recent introduction of MBTCs in [9] is a further step to close the gap to the Shannon limit. Indeed, the MBTCs offer more advantages than BTC such as lower error floor for moderate codeword size and faster convergence [10]. These advantages are crucial in the current and future wireless systems as IEEE 802.11n and IEEE 802.16.

In this paper, we extend the MBTC concept to the non-binary case: Whereas the constituent codes of the MBTC have R binary inputs, we consider TC with constituent codes with R non-binary inputs. We refer this new family as multiple input non-binary turbo-codes (MNBTC). Formally, the code has the same structure as the multi binary code but the arithmetic operations are now performed in GF(2^Q), Galois field of order Q. In this paper, we propose to analyze the MNBTC and to compare them to the MBTC from the viewpoints of decoding algorithms and performance.

The rest of the paper has the following structure. In the next section the construction of the constitutant codes of the MNBTC is presented. Section III is dedicated to the trellis closing problems of the MNB code. In section IV we present several variations of extended versions of MAP decoding algorithms for the MNB codes. In Section V, we show that under some restriction on the polynomials, a Reed-Solomon (RS) code is a particular case of MNB, so the MNBTC family includes an interesting new class of TC, the RS-TC. Finally, some experimental results and concluding remarks are presented in Section V and VI, respectively.

II. MULTI NON BINARY CONVOLUTIONAL ENCODER AND TRANSMISSION CHANNEL

In this section, we describe the encoding scheme of the constituent codes of the MNBTC. Each constituent encoder has R non-binary inputs and is referred as multi non-binary code (MNB). In Fig.1 we present the general scheme of a MNB convolutional encoder, with rate $Rc=R/(R+1)$. Throughout the paper, we focus on recursive and systematic codes due to their superior performance. Each register in Fig.1 stores a vector of Q bits at the time. All the links are supposed to have a width equal to Q in order to carry a vector of Q bits. Each block $g_{r,m}$, with $r=1,\dots,R$, $m=0,\dots,M-1$, represents a multiplier in $GF(2^Q)$ whereas the adders perform the sum in $GF(2^Q)$. At time n, the encoder has R inputs, u_1^n, u_2^n,..., u_R^n and

[1] Facultatea de Electronică şi Telecomunicaţii, Departamentul Comunicaţii, Bd. V. Pârvan Nr. 2, 300223 Timişoara, e-mail horia.balta@etc.upt.ro, maria.kovaci@etc.upt.ro, radu.lucaciu@etc.upt.ro

[2] Department of Electrical and Computer Eingineering, Rice University, MS-366 – 6100 Main Street, Huston, Texas 77005, e-mail debaynas@rice.edu

[3] Universitatea Politehnica din Bucuresti, Facultatea de Electronică şi Telecomunicaţii si Teoria Informatie, 1-3 Iuliu Maniu, Bucuresti, e-mail cvladeanu@yahoo.com

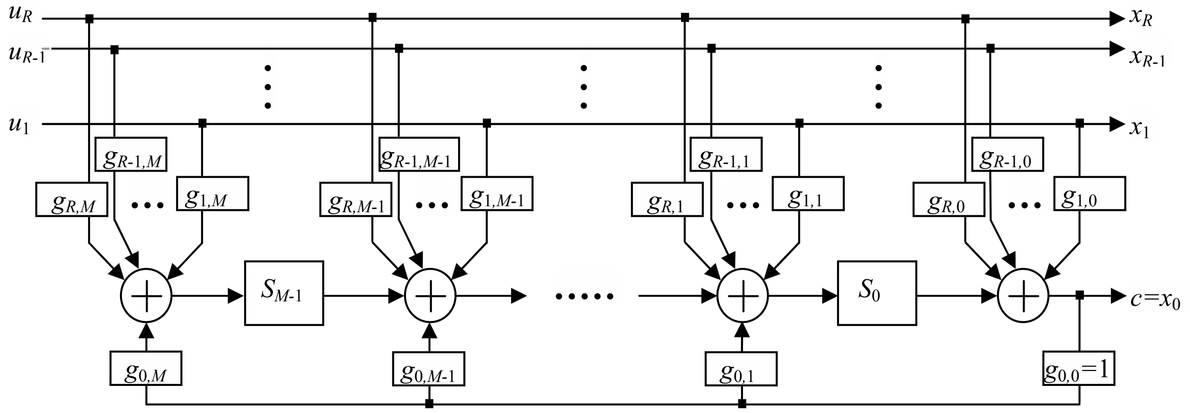

Fig.1 Multi-Non-Binary Convolutional Encoder – general scheme.

R+1 outputs $x_1^n, x_2^n, \ldots, x_R^n$ corresponding to the R inputs and one redundant bit x_0^n also referred as c^n. The current encoder state is given by the outputs of the M shift registers S_0^n, $S_1^n, \ldots, S_{M-1}^n$.We adopt the following compact notations: $s^n = \left[S_{M-1}^n \quad S_{M-2}^n \quad \mathrm{K} \quad S_0^n\right]^T$,

$u^n = \left[u_R^n \quad u_{R-1}^n \quad \mathrm{K} \quad u_1^n\right]^T$, $0 \le n \le N$, for the encoder state vector and the ``input word'', respectively.

The input/current state and output/current state relations of the encoder at the time n can be respectively expressed in the compact form:

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

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

where: $G_T = G_F \cdot G_L + G_0$ and $W = [0\ 0\ \ldots\ 0\ 1]_{1\times M}$.

$G_0 = [g_{r,m}]_{M\times R}$ denotes the partial generator matrix, $1 \le m \le M$, $1 \le r \le R$, which excludes the feedback coefficients and the generator coefficients for the redundant symbol. The vector $G_F = [g_{0,M}\ g_{0,M-1}\ \ldots\ g_{0,1}]^T$ contains the coefficients of the feedback loop. and $G_L = [g_{R,0}\ g_{R-1,0}\ \ldots\ g_{1,0}]$. The matrix T equals:

$$T = \begin{bmatrix} 0 & 0 & 0 & \ldots & 0 & g_{0,M} \\ 1 & 0 & 0 & \ldots & 0 & g_{0,M-1} \\ 0 & 1 & 0 & \ldots & 0 & g_{0,M-2,} \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & \ldots & 1 & g_{0,1} \end{bmatrix} = \begin{bmatrix} 0_{1\times M-1} & G_F \\ I_{M-1} & \end{bmatrix}.$$

Additionally, we define the full generator matrix G such as: $G = [g_{r,m}]_{(M+1)\times(R+1)} = [g_R\ g_{R-1}\ \ldots\ g_1\ g_0]_{10}$, $0 \le m \le M$, $0 \le r \le R$.

The necessary and sufficient condition to have a decodable code is such that the matrix G_T is full rank.

Applying the „D" transform ($X(D) = \sum_{k=-\infty}^{+\infty} x^k \cdot D^k$) to the equations (1) and (2) we obtain:

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

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

After some basic manipulations, it can be shown that:

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

where $g_r(D) = \sum_{m=0}^{M} g_{r,m} \cdot D^m$.

In order to understand better the encoding procedure, we give an example in $GF(4)$. We recall the addition and multiplication operations on $GF(4)$ in Fig.2., [11].

+	0	1	2	3
0	0	1	2	3
1	1	0	3	2
2	2	3	0	1
3	3	2	1	0

*	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	2	3	1
3	0	3	1	2

Fig. 2 The addition and the multiplication in $GF(4)$.

Consider the double-non-binary convolutional code defined by the following generator matrix:

$$G = \begin{bmatrix} 2 & 1 & 3 \\ 0 & 3 & 1 \\ 3 & 1 & 1 \end{bmatrix}. \quad (6)$$

By construction, from G, we have: G_L=[3 1], G_F=[3 1]^T, G_T=[0 2;3 2] and T= =[0 3;1 1]. Consider two input sequences u_1 and u_2 starting as follows: u_1 = [1 2 3 3 1 3 2 …] and u_2 = [0 2 1 1 3 0 3 ...]. The values of the state vector of the encoder are determined according to (1) with initial state s^0=[0 0]^T. After some basic calculations based on the operation tables given in Figure 2, it is easy to show that s^1=[0 3]^T and c^1=3 where c^1 is determined by (2), s^2=[1 1]^T and c^2=0, s^3=[1 0]^T and c^3=2, s^4=[2 1]^T and c^4=3, s^5=[2 1]^T and c^5=1 and s^6=[3 1]^T and c^6=3, etc.

After the description of the encoding process for each constituent code, we describe next the full encoding method for the MNBTC.

Fig.3 shows the scheme for the MNBTC. The input sequence $(u^n)_{0\le n<N}$ = $[u^0\ u^1\ \ \ldots\ u^{N-1}]$ represents a sequence of N+1 consecutive input words. Every word of size $R\times 1$ is formed by stacking R consecutive symbols, i.e., u^n = $[u_{R+1}^n\ u_R^n\ \ldots\ u_2^n]^T$. In turn, each symbol results from a mapping of Q bits such as:

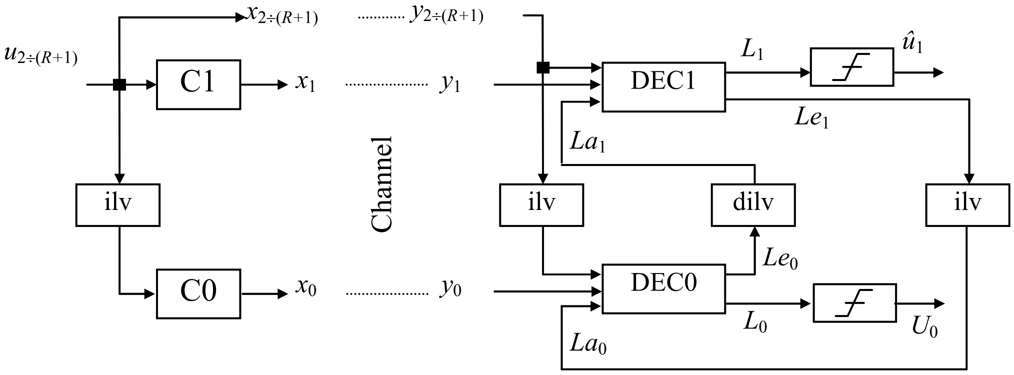

Fig. 3 Principle of the encoder and decoder for the Multiple inputs Non Binary Turbo-Code.

$u_r^n = [u_{r,Q-1}^n \; u_{r,Q-2}^n \; \ldots \; u_{r,0}^n]^{\mathrm{T}}$, $2 \le r \le R+1$, with: $u_{r,q}^n \in \{0, 1\} = GF(2)$, $0 \le q < Q-1$.

At the output of the encoders C1 and C0, we obtain the codeword $(x^n)_{0 \le n < N} = [x^0 \; x^1 \; \ldots \; x^{N-1}]$, where each word $x^n = \left[x_{R+1}^n \; x_R^n \; \mathrm{K} \; x_2^n \; x_1^n \; x_0^n\right]^T$ is composed by R+2 symbols: the first R symbols correspond to the input symbols whereas the last two symbols correspond to the two redundant symbols. As for the input sequence case, each symbol of the codeword corresponds to Q bits $x_r^n = \left[x_{r,Q-1}^n \; x_{r,Q-2}^n \; \mathrm{K} \; x_{r,0}^n\right]^T$ .The codeword of length N that corresponds to $N\times(R+2)\times Q$ bits is then modulated (throughout the paper, for sake of simplicity we consider BPSK or QPSK signaling) and then transmitted through the channel to its destination. The destination received a noisy version $(y^n)_{0 \le n \le N}$ of the transmitted signal. By using the same formulation as we did for the transmitted sequence $(x^n)_{0 \le n < N}$, the received sequence can be expressed as: $y^n = \left[y_{R+1}^n \; y_R^n \; \mathrm{K} \; y_2^n \; y_1^n \; y_0^n\right]^T$.

The vector y^n has R+2 components where each component can be represented itself by a vector of Q values $y_r^n = \left[y_{r,Q-1}^n \; y_{r,Q-2}^n \; \mathrm{K} \; y_{r,0}^n\right]^T$. Each component $y_{r,q}^n$ can be modeled for a transmission over additive white Gaussian channel as: $y_{r,q}^n = x_{r,q}^n + w_{r,q}^n$, $0 \le q < Q$, where $w_{r,q}^n$ represents the receiver noise. We model $w_{r,q}^n$ as zero-mean mutually independent Gaussian random sequences with variance σ^2.

III. THE MULTI NON BINARY DECODING AND THE TRELLIS CLOSING

In this section, we describe the modifications of the MAP decoding algorithm that are needed in order to decode the MNBTC codes.

Considering the two decoders DEC1 and DEC0 in Fig.3. For any variation of the iterative MAP decoding algorithm, e.g., MAP, LogMAP, MaxLogMAP or SOVA, the decoding of the MNBTCs can be done in different manners: per word, per symbol or per bit. The three methods calculate differently the a priori probabilities, the extrinsic messages, and the a posteriori probabilities (APP). We successively describe the calculation for the three cases:

Word-wise decoding supposes that all probabilities, i.e. the a priori, extrinsic and a posteriori probabilities correspond to one word among the $N_W = (2^Q)^R$ possible words. Using the same notations as in Fig.3, we can express the APPs $L_j^{n,i}(d)$ and the extrinsic probabilities $Le_1^{n,i}(d)$ for each decoder j, j = 0 or 1, at iteration i as:

$$\begin{aligned} L_1^{n,i}(d) &= La_1^{n,i}(d) + Y_1^n + Le_1^{n,i}(d), \\ L_0^{n,i}(d) &= La_0^{n,i}(d) + Y_0^n + Le_0^{n,i}(d), \end{aligned} \qquad (7)$$

where $d \in \Delta = \{0, 1 \ldots N_W-1\}$ denotes the index of the candidate word $u^n(d)$ among the N_W possible words of the code. Y_1^n and Y_0^n correspond to the bit-wise received sequence and are determined as follows:

$$Y_j^n = \frac{1}{\sigma^2} \cdot \sum_{r=2}^{R+1} \sum_{q=0}^{Q-1} x_{r,q}^n \cdot y_{r,q}^n + \frac{1}{\sigma^2} \cdot \sum_{q=0}^{Q-1} x_{j,q}^n \cdot y_{j,q}^n \; , \quad (9)$$

with j = 0 or 1 and and noise dispersion σ^2.

Thus, the a priori probabilities are expressed as:

$$\begin{aligned} La_1^{n,i}(d) &= \pi^{-1}(Le_0^{n,(i-1)}(d)), \\ La_0^{n,i}(d) &= \pi(Le_1^{n,(i-1)}(d)). \end{aligned} \qquad (8)$$

The operations π ( ) andπ^{-1}( ) denote the interleaving and the de-interleaving operations, respectively (ilv" and „dilv" respectively in Fig.3).

Note that the number of components for the extrinsic probabilities is equal to the number of outgoing vertices at any node of the treillis since each vertice corresponds to one possible value of the information word.

After several iterations, the estimated word $\hat{u}$ of the transmitted sequence can be determined for each n by searching the largest value of the APP given by one of the decoders:

$$\hat{u}^n = \max_{d \in \Delta} L_j^{n,i}(d). \qquad (10)$$

For the *Symbol-wise decoding* algorithm, the APPs and the extrinsic probabilities are computed per each symbol u_r^n, $1 \leq r \leq R$, at time n. Since a symbol corresponds to Q bits, there are 2^Q possible values for the estimate of u_r^n. Thus, at iteration i, both decoders compute $R{\cdot}2^Q$ values for the APPs and extrinsic probabilities. This decoding strategy is similar to the approach proposed in [12] for the MBTC codes.

For the *Bit-wise decoding* algorithm, either the APPs, or the log likelihood ratios (LLRs) can be used. In the first variant, we compute two values which correspond to the binary values 0 or 1 for every bit $u_{r,q}^n$ of each symbol u_r^n of the every word u^n, and that from all N words of the original sequence u. Thus, both decoder, at iteration i, calculate $2{\cdot}R{\cdot}Q$ values for APP and the extrinsic probabilities. Alternatively, the LLRs can be computed for the $R{\cdot}Q$ bits of u^n as in [8]. The decoding algorithms correspond to the ones that are used for the BTC.

The word-wise decoding approach has the largest computation complexity compare to the symbol-wise and bitwise approach since the computational complexity is exponential with respect to the Galois field order Q and the number of inputs R whereas the computational complexity of the word-wise and the bit-wise approaches is linear in the number of inputs R and exponential (resp. linear) with respect to the number of inputs Q for the symbol-wise (resp. bit-wise). However, the word-wise decoding provides better performance as it is demonstrated in Section V.

After describing the update of the probabilities at the decoder, we detail the trellis closure. The trellis closure of the MNBC is more complicated than from the NBC because the trellis itself is more complicated. In this paragraph, we propose a new termination scheme for the MNBTC.

For a given input sequence $[u^0\ u^1\ \ldots\ u^{N-1}]$, the final state of the trellis represented by s^N is:

$$s^N = \sum_{j=0}^{N-1} T^{N-j-1} \cdot G_T \cdot u^j + T^N \cdot s^0. \qquad (11)$$

The trellis closure can be performed in two different ways: i) by making the trellis circular; ii) by zero-padding as in [5,12].

The trellis is *circular* if $s^N = s^0$. Replacing this equality in (11), we obtain:

$$(I_M + T^N) \cdot s^0 = \sum_{j=0}^{N-1} T^{N-j-1} \cdot G_T \cdot u^j = s^x. \qquad (12)$$

As soon as s^x has been estimated, the corresponding initial state can be determined as:

$$s^0 = (I_M + T^N)^{-1} \cdot s^x, \qquad (13)$$

subject to the constraint that N is not a multiple of p, with p period of T defined as the smallest integer such that $T^p = I_M$. Indeed, if $T^p = I_M$, the right term in (13) is always equal to 0 for any s^x, so (13) cannot be not satisfied for $s^x \neq 0$.

By closing the trellis with *zero padding,* we have $s^N = 0$ and (11) becomes:

$$\sum_{j=0}^{N-1} T^{N-j-1} \cdot G_T \cdot u^j = 0_{M\times 1}. \qquad (14)$$

We first have to determine the number of unknowns $u^j \in GF(2^q)$ in (14) that are required to close the trellis. We show next that it is necessary and sufficient to have M unknowns in order to close the trellis. Decompose M as:

$$M = a \cdot R + b, \quad a, b \text{ integers}, \quad 0 \leq b < R, \qquad (15)$$

Then, (14) becomes:

$$\begin{bmatrix} G_T & \mathrm{K} & T^{a-1} \cdot G_T & T^a \cdot G_T \end{bmatrix} \cdot \begin{bmatrix} u^{N-1} \\ \ldots \\ u^{N-a} \\ u^{N-a-1} \end{bmatrix} = \qquad (16)$$

$$s^{N-a-1} = \sum_{j=0}^{N-a-2} T^{N-j-1} \cdot G_T \cdot u^j.$$

Since the matrices $T^k \cdot G_T$, $0 \leq k < p$, are supposed to be full column rank or full row rank, we select a set of indices $\{i_1, i_2, \ldots i_b\} \subset \{0, 1, \ldots R\}$ such that the corresponding truncated matrix $(T^a \cdot G_T)^b$ formed by the columns $\{i_1, i_2, \ldots i_b\}$ of the matrix $T^a \cdot G_T$ is full rank. Taking into account that the matrix T is periodic of period $p > a$, the matrix $A = [G_T \ldots T^{a-1} \cdot G_T\ \ (T^a \cdot G_T)^b]$ is invertible. Therefore, we have to introduce $a+1$ redundant symbols u^j in order to close the trellis, i.e.:

$$\begin{bmatrix} u^{N-1} \\ \ldots \\ u^{N-a} \\ \left(u^{N-a-1}\right)^b \end{bmatrix} = A^{-1} \cdot s^{N-a-1} + (T^a \cdot G_T)^{R-b} \cdot (u^{N-a-1})^{R-b}, \qquad (17)$$

where the partial word $(u^{N-a-1})^b$ corresponds to the components $\{i_1, i_2, \ldots i_b\}$ of the word u^{N-a-1}. $(u^{N-a-1})^{R-b}$ is the partial word of u^{N-a-1} built from the $(R-b)$ remaining components. $(u^{N-a-1})^{R-b}$ can be set to zero for simplicity or can be dedicated to some information data, for a slightly higher encoding rate.

In this section, we investigated three different manners to perform the decoding of a codeword. We also show that the trellis of the MNBTC can be terminated in two different ways (circular closure and zero padding) as for the classical TC. We determine for both cases, the conditions on the codeword that ensure the trellis closure. The circular closure (tail biting) is more efficient since it provides a higher coding rate than the closure by zero padding. Nevertheless, the closure with zero padding offers better performance, as we will show in section V.

In the next section, we show that the MNBTCs include a new family of TC with Reed-Solomon (RS) as component codes with very interesting properties.

IV. REED SOLOMON TURBO-CODE

We showed in the previous section that the MNBTC include the NBTC which include themselves the classical TC. In this section, we show that this new family also includes a very interesting family that we call RS-TC since the constituent codes are RS codes.

Proposition 1: A MNB code is equivalent to a RS code if the following constraints hold:

1) the polynomial $g_0(D)$ corresponds to the generator polynomial of the targeted RS code [11]. Implicitly, it results M=grad(g_0);

2) The encoder has a single input: $R = 1$ and $g_1(D) = 1$. Thus, each word corresponds to a single symbol;

3) the trellis is closed at zero. The redundancy is determined only by the symbols that help to close the trellis in (17) [11].

4) the length of the sequence of symbols which is equal to $N = 2^q$-1 should match the RS word length.

The turbo encoder will generate $2{\cdot}M$ redundant words (M from each constituent encoder) from the k=N-M input symbols. If the four conditions above hold, the RS turbo codes can be viewed as a special case of the MNBTC codes.

V. SIMULATIONS

Bit error rate and frame error rate performance of the MNBTC of memory 3 with two inputs defined by generator matrix G=[2 1 2; 3 2 2; 0 0 2; 3 1 1] are presented with respect to the signal-to-noise ratio (SNR) in Fig.4. The data block is composed of N=376 words, each word of two symbols and each symbol of two bits. The decoding algorithm that we used is the approximation MaxLogMAP of the word-wise MAP algorithm described in Section III.

The intersymbol interleaving is realized with an S interleaver, with S=21 [13]. A second interleaver is used to permute the symbols between words with an even index.

We consider an AWGN transmission channel with BPSK signaling. The trellis termination is realized with zero padding as discussed in Section III. A stopping criterion based on a threshold for the APP values is applied. The maximum number of decoding iterations is set to 15.

The performance presented in Fig.4 is slightly worse than to the 4-memory MBTC proposed in [12]. It may come from the fact that we optimized neither the component codes nor the interleavers in this study. However, the MNBTC have several intrinsic advantages: deeper slop (we conjecture that MNBTC have a larger minimum distance than the NBTC), very low error-floor even for moderate codeword length (for a FER lower than 10^{-4}) and faster convergence speed. Indeed, At equivalent SNR, the average number of iteration is smaller for the MNBTC than for the BTC proposed in [11] with also memory of 4).

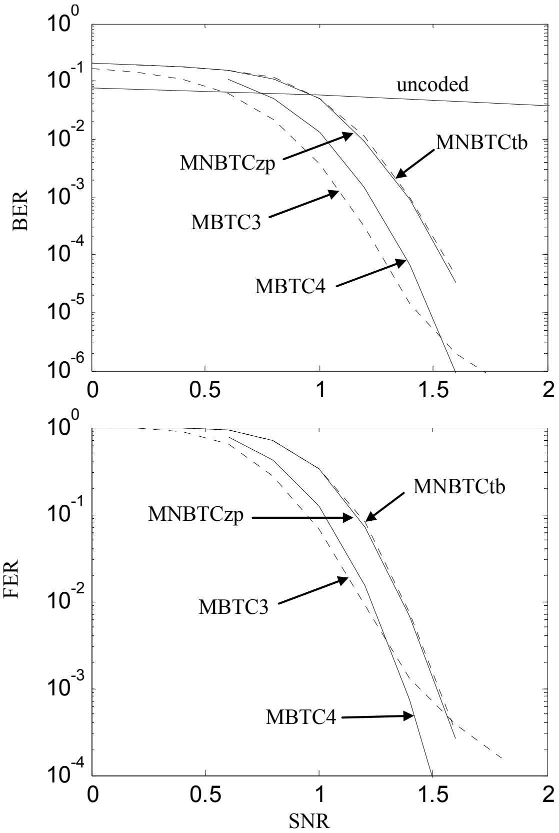

Fig 4. Comparison between the Multi Non Binary Turbo Code with encoding matrix G=[2 1 2; 3 2 2; 0 0 2; 3 1 1] (tb: tail biting, zp: zero padding) and the Multi Binary Turbo Codes (3 and 4 memory) proposed in [12] over AWGN transmission channel: Bit Error Rate (BER) and Frame Error Rate (FER) are plotted as functions of the signal-to-noise ratio (SNR).

VI. CONCLUSIONS

We introduced in this paper a new class of Turbo-codes referred as the Multiple Input Non Binary Turbo Codes where each constituent code is a non binary code with multiple inputs. We showed that this family is very general and includes all existing TCs. Particularly, it includes the classical TC, the recently proposed NBTC [12] but also a new class of turbo-codes with Reed Solomon code as constituent code. Although we did not optimize the component codes, our codes have similar performance than the codes presented in [12]. We expect significant gain by optimizing the interleaver and the components codes.

REFERENCES

[1] C. Berrou, A. Glavieux, P. Thitimajshima, „Near Shannon Limit Error –Correcting Coding and Decoding: Turbo –Codes", Proc. of ICC, Geneve, may 1993, pp. 1064-1070.

[2] S. Dolinar, D. Divsalar, "Weight Distributions for Turbo Codes Using Random and Nonrandom Permutations", TDA Progress Report 42-122, August15, 1995.

[3] Oscar Y. Takeshita, and Daniel J. Costello. Jr., "New Classes of Algebraic Interleavers for Turbo-Codes", ISIT 1998, Cambridge, MA, USA, August 16-21.

[4] C. Berrou, Y. Saouter, C. Douillard, S. Kerouédan and M. Jézéquel, "Designing good permutations for turbo codes: towards a single model", International Conference on Communications, ICC'2004, Paris, France, June 2004, Vol. 1, pp.341-345.

[5] C. Weiss, C. Bettstetter, S. Riedel, and D.J. Costello, "Turbo decoding with tailbiting trellises", in Proc. IEEE Int. Symp.Signals, Syst., Electron., Pisa, Italy, Oct. 1998, pp. 343-348.

[6] [DIP] D.Divsalar, F.Pollara, "Serial and Hybrid Concatenated Codes with Applications", International Symposiom on Turbo Codes – Brest – France, 1997, pag. 80-87.

[7] J. Hagenauer, P. Hoeher, "A Viterbi algorithm with soft-decision outputs and its applications" ,Proc. Of GLOBECOM'89, Dallas, Texas, 47.1.1-47.1.7, 1989.

[8] P. Robertson, P. Hoeher and E. Villebrun, "Optimal and suboptimal maximum a posteriori algorithms suitable for turbo decoding", European Trans. Telecommun., vol. 8, pp. 119-125, Mar-Apr. 1997.

[9] C. Berrou and M. Jezequel, „Nonbinary convolutional codes for turbo coding", Electron. Lett., vol. 35, no. 1, pp. 39-40, Jan. 1999.

[10] C. Berrou, M. Jezequel, C. Douillard, and S. Kerouedan, „The advantages of nonbinary turbo codes", in Proc. Inf. Theory Workshop, Caims, Australia, Sept. 2001, pp.61-63.

[11] I. Reed, and G. Solomon, "Polynomial Codes over Certain Finite Fields" Journal of the Society for Industrial and Applied Mathematics, Vol.8, No.2, Jun. 1960, pp.300-304

[12] C. Douillard, and C. Berrou, "Turbo Codes With Rate-m/(m+1) Constituent Convolutional Codes, IEEE Trans. Comm., vol.53, No. 10, October, 2005, pp.1630-1638.

[13] S. Dolinar, and D. Divsalar, „Weight Distributions for Turbo Codes Using Random and Nonrandom Permutations", TDA Progress Report 42-122, August 15, 1995