A Study on Non-Binary Turbo Codes

Horia BALTA, Maria KOVACI
University "Polytechnic" of Timişoara, Faculty of Electronics and Telecommunications,
Postal Address, 30223 Timişoara, ROMANIA, E-Mail: horia.balta@etc.utt.ro,
maria.kovaci@etc.utt.ro

Abstract: - In this paper a study of unpunctured non-binary turbo codes, NBTCs from point of view of trellis closing is made. Non-binary turbo codes, rates R=n/(n+2), n>1 (where n is the inputs number) are built from two Recursive Systematic Convolutional (RSC) component codes. The trellis closing methods of the convolutional codes, with more inputs, were studied and new closing methods were proposed. The simulations were made for constraint lengths K=3, K=4 and for n=2 (duo-binary) and n=3 inputs number, and for block lengths around 1800 bits. The AWGN and the BPSK modulation were employed. The turbo decoder was used 15 iterations for each block. A Log Likelihood Ratio (LLR) stop criterion was selected. The BER performances of the non-binary turbo codes cases with: both unclosed trellises, the trellis of the first component code is closed and the trellis of the second component code is unclosed, both trellises are closed and the circular trellises cases are compared.
Keywords: - non-binary turbo codes, trellis

I. Introduction

The non-binary or multi-binary turbo codes have more than one input, [1]. Thus, the (convolutinal) codes present the same inputs number. The general scheme of the multi-inputs convolutional coder (multi-binary named in the following) is shown in the Fig. 1. The coder has r binary inputs, noted u_r, u_{r-1}, … u_1 and $r+1$ outputs, constructed by he same r bits plus one control bit, noted u_0. The coder relations are:

$$\begin{aligned}
s_{m-1}^{t+1} &= g_{r,m}\cdot u_r^t + g_{r-1,m}\cdot u_{r-1}^t + \ldots + g_{1,m}\cdot u_1^t + g_{0,m}\cdot c^t \\
s_{m-2}^{t+1} &= g_{r,m-1}\cdot u_r^t + g_{r-1,m-1}\cdot u_{r-1}^t + \ldots + g_{1,m-1}\cdot u_1^t + g_{0,m-1}\cdot c^t + s_{m-1}^t \\
&\quad\cdots\cdots \\
s_0^{t+1} &= g_{r,1}\cdot u_r^t + g_{r-1,1}\cdot u_{r-1}^t + \ldots + g_{1,1}\cdot u_1^t + g_{0,1}\cdot c^t + s_1^t \\
c^t &= g_{r,0}\cdot u_r^t + g_{r-1,0}\cdot u_{r-1}^t + \ldots + g_{1,0}\cdot u_1^t + s_0^t .
\end{aligned} \tag{1}$$

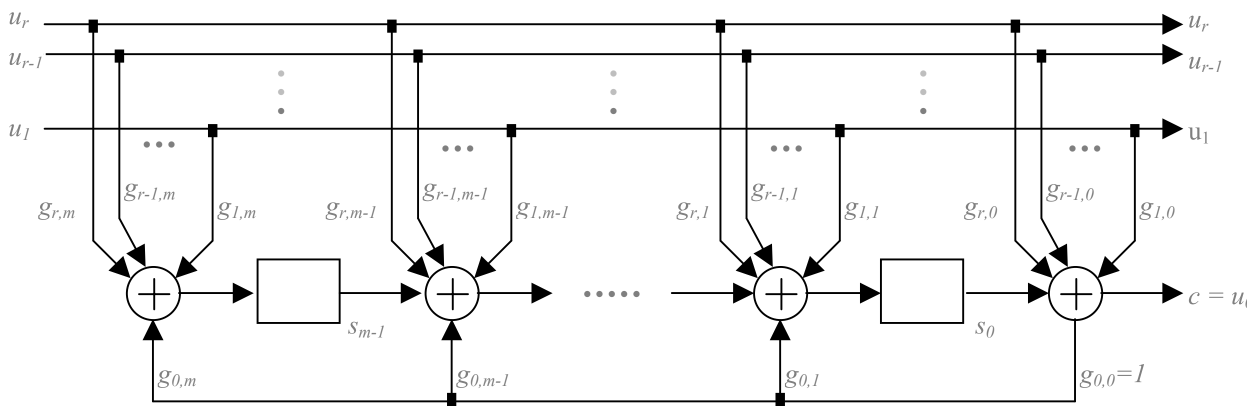


Fig. 1 Multi-inputs convolutional coder

The circuit (coder) functioning analyze of the Fig. 1 can be performed using „D" transform. Thus, transferring in the frequency domain the equations of the relation (1) and multiplying by turns, the first equation with D^m, the second with D^{m-1}, and so on, the last but one equation with D and summing (all of them) we obtain the equation:

$$g_0(D)\cdot c(D) = U_r(D)\cdot g_r(D) + U_{r-1}(D)\cdot g_{r-1}(D) + \ldots + U_1(D)\cdot g_1(D), \qquad (2)$$

with:

$$c(D) = U_r(D)\cdot \frac{g_r(D)}{g_0(D)} + U_{r-1}(D)\cdot \frac{g_{r-1}(D)}{g_0(D)} + \ldots + U_1(D)\cdot \frac{g_1(D)}{g_0(D)}. \qquad (3)$$

The $g_k(D)$ polynomials, with $k = 0 \div r$, have identical coefficients with the generating matrix coefficients, G:

$$g_k(D) = \sum_{j=0}^{m} g_{k,j}\cdot D^j, \qquad 0 \le k \le r. \qquad (4)$$

The $g_k(D)$ polynomials, with $k = 0 \div r$, correspond to the r inputs, and $g_0(D)$ is the reaction polynomial. The polynomial coefficients $U_j(D)$, cu $j = 1 \div r$, constitute the data sequences provided to the multi-binary coder through the r inputs of them:

$$U_j(D) = \sum_{t=0}^{N} u_j^t \cdot D^t, \quad 0 \le j \le r. \qquad (5)$$

II. The state diagram. The trellis diagram.

Starting from the equation system (1) we can construct the state diagram of the multi-binary coder. The diagram will contain 2^m nodes, suitable to the 2^m possible states. From each node will leave 2^r branches, suitable to the 2^r inputs vectors possible. Thus, the diagram will contain 2^{m+r} branches or transitions. Each transition is associated of the one symbol duration. On the symbol duration r input bits are take over and $r+1$ bits are generated at the output. The duo-binary coder suitable to the generating matrix (made by g_{ij} coefficients, from the system equations (1)), is presented on Fig. 2:

$$G = [5\ 3\ 7]_8 = [1\ 0\ 1;\ 0\ 1\ 1;\ 1\ 1\ 1], \qquad (6)$$

and the corresponding state diagram and trellis, [2] are presented on Fig. 3 a) and b).

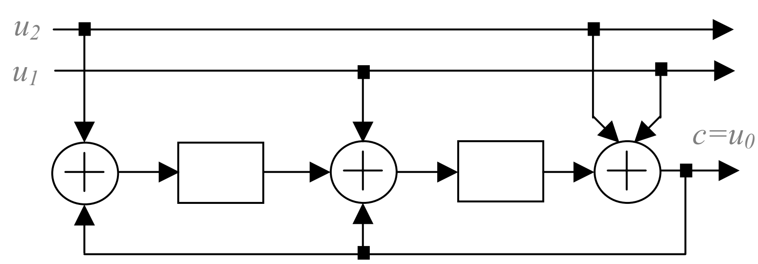


Fig.2 Duo-binary coder associated to the matrix $G = [5\ 3\ 7]_8$;

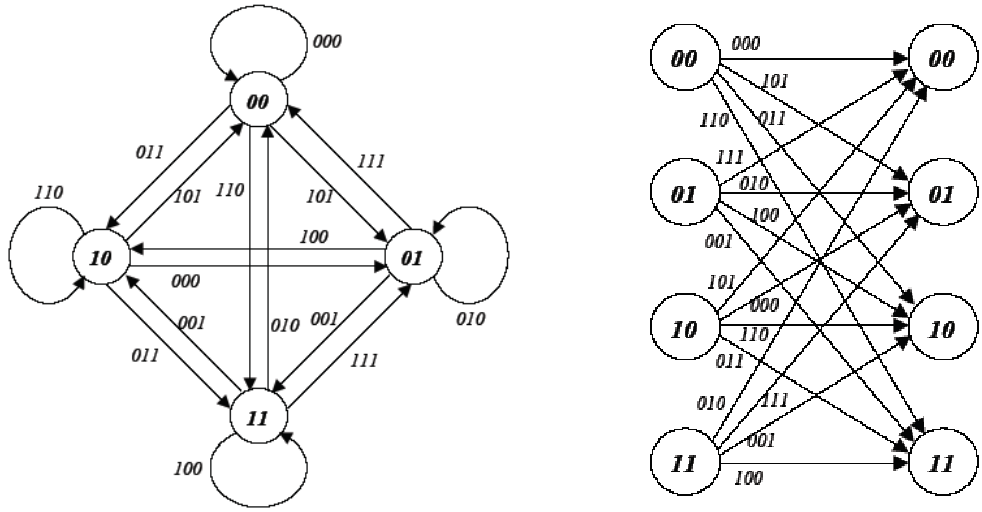


Fig.3 An example of the duo-binary code: a) state diagram, b) trellis diagram.

III. The trellis closing of the multi-binary convolutional codes

As the classic turbo codes, [3], with one input, the multi-binary turbo code segments the date sequences on blocks, which are coded and decoded, the turbo code itself becoming a code block. Thus, it appears the necessity to finalize the coding by the trellis closing. The trellis closing suppose the insertion, on data sequences, of the some redundant bits, with the goal to force on decoder a certain state, at the end of the coding. Except the circular codes, this state will be the null state. In this paragraph the trellis closing methods of the multi-binary convolutional code are described, independently by his posture as component code of the turbo code. Also, the trellis closing problems of the turbo code component codes is described.

A. The circular trellis

Similar to the circular binary codes, the circular multi-binary codes perform the coding of the each dates block starting and ending with the same coder state (different, from a block to the other). To realize this thing, the multi-binary coder performs previously a pre-coding, fact that allow it to find the state from where it must be starting the coding of the respective block.

B. The closing trellis at the null state

The trellis of the multi-binary code can be closed at the null state, $0_{m\times 1}$, with the insertion price of the k redundant bits in dates block. To find these bits, we suppose that between m – the coder states number (the coder memory) and r – the inputs number, there is the relation:

$$r\cdot (q-1) < m \le r\cdot q. \qquad (7)$$

Obviously, no matter in what state will be the coder with q paths before the coding finalizing, there are $2^{r\cdot q-m}$ paths through trellis, which leads, at the final, to the null state (noted with „*00*” on the Fig.3 diagrams). If, $r\cdot q=m$, then there is only one path which leads at the null state. Thus, the trellis closing supposes to add the afferents m bits, of this path, on data sequence. If, $r\cdot q>m$, then from the $r\cdot q$ bits which must to be performed for the respective block, $r\cdot q-m$ bits are allocate to the dates and m bits will specify the path what leads to the null state.

IV. The turbo code closing strategies

The multi-binary turbo code, MBTC, similar with TC classic, contains at least two component codes (multi-binary). Because of the interleaving of the information sequence, provided to the second coder, the trellis closing of this decoder cannot be made by the insertion of the information bits between of the N bits from the interleaved block. The only possibility is that these bits cannot be included between the bits provided to the first coder.

Four possibilities of the MBTC trellis closing and their coding rates are presented on Table 1.

NN. The variant of the unclosed of the both trellises is the most easily to be implemented, and alongside circular variant, it provides the higher coding rate.

CC. The circular variant is attractive from point of view of the coding rate, but it supposes a double volume of compute at coding. The decoder does not know exactly the state from it can start, in the coding process, but it knows that it must start and end with the same state. This fact is used in initialization of the forward and backward recurrences from the decoding algorithm (Maxim A Posteriori).

ZN. The closing of the first trellis is made with the redundancy-increasing price, which implies the coding rate decreasing. The advantage is the firm knowing of the start coefficients in recurrence by the first decoder, but not by the second decoder.

ZZ. Both decoders know the beginning states and how to initialize their forward and backward recurrences coefficients. However, the complications appear in the multiplexing of the added bits by the second coder for the closing of his trellis. Moreover, these bits are not turbo coded and they cannot serve to the turbo decoding process.

V. Experimental results

The MBTC simulation results, for the four trellis closing strategies described in the precedent paragraph, are presented in Fig. 4 a) and b). The implementation is done for two pairs of the RSC components codes. In the first case the memory 2 duo-binary code is used, having the generating matrix *G=[1, 1, 5/7, 3/7],* and in the second, the 3 memory duo-binary code is used, having the generating matrix *G=1, 1, 13/15, 11/15]*.

In all the cases, at decoding, the turbo decoder was used 15 iterations for each block, a LLR stop criterion was selected, [4], and a MAP algorithm was used. The AWGN and the BPSK modulation were employed.

The obtained BER and FER curves were compared with the most performing classic codes, with the same coding rate (approximated at *1/2*) and with the same memory (memory 2 for

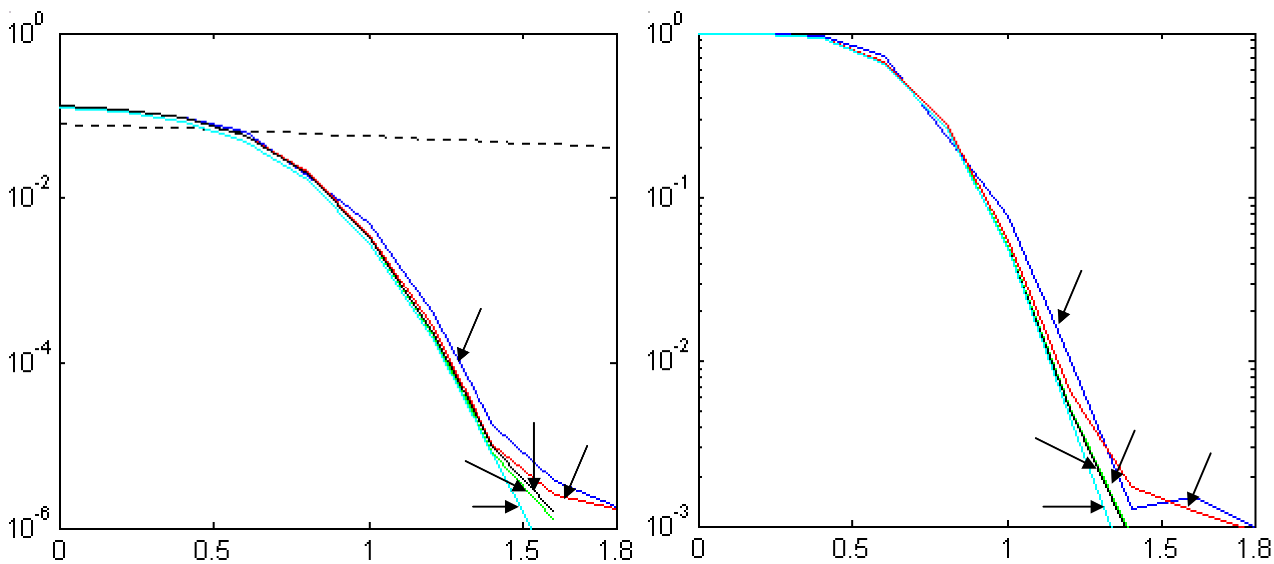
Table 1

Variant	Closing form Code 1	Code 2	Coding rate Rc
NN	unclosed	unclosed	$r/(r+2)$
CC	circular	circular	$r/(r+2)$
ZN	closed	unclosed	$r \cdot (N-m)/((r+2) \cdot N)$
ZZ	closed	closed	$r \cdot (N-m)/((r+2) \cdot (N+m))$

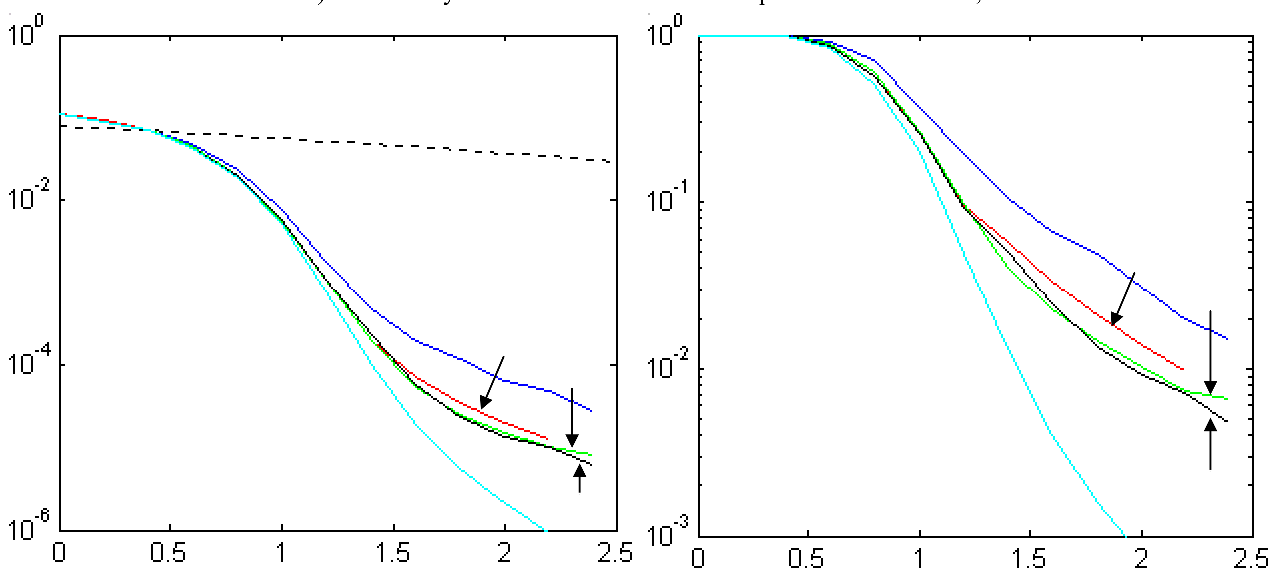
the code with *G=[1, 5/7]* and memory 3 for the code *G=[1, 15/13]*, both punctured with the punctured matrix *P=[10; 01]*).

There is an essential difference between MBTC performances and the punctured one, for the memory 2. Practically, there is not this difference between *15/13* classic-punctured code and *13/11/15* MBTC, for memory 3.

From all the trellis closing variants proposed, as it was expected, the NN variant presents the worst performances. Also, a higher discrepancy between BER and FER performances can be remarked. If at 13/11/15 code the BER performances practically are the same, the FER curves differ essentially. This situation appears because of the "spreading" of the errors through more blocks. The adequate unclosing of the trellis generates these errors.



a) Duo-binary TC 13/11/15 versus classic punctured TC 15/13;



b) Duo-binary TC 5/3/7 versus classic punctured TC 5/7.

Fig. 4 The turbo code performances for the closing strategies presented in Table 1.

VI. Conclusions

In this paper a study on the trellis closing problems of the unpunctured MBTC is approached. The trellis closing possible strategies and their performances (obtained by simulations), for two cases of the duo-binary codes, are presented. Taking in account the implementation complexity, the coding rate and the performances obtained, the most indicated closing variant is ZN (the first trellis closed, the second trellis unclosed). The study must be continued, to verify also the conclusions for the others multi-binary components codes, probably with better performances then the classic TC at the same rate.

Acknowledgement

The authors want to thank their Ph.D. director, Professor Miranda Nafornițā, for the suggestion to write this paper and for her continuous help.

References

[1] C. Berrou, M. Jézéquel, C. Douillard, S. Kerouédan, „The Advantages of Non-Binary Turbo Codes", Information Theory Workshop ITW2001 Cairns, Australia, Sept 2-7, 2001, pp. 61 – 63;

[2] M. Jézéquel, C. Berrou, C. Douillard, „Turbo codes (convolutifs)", seminar, Timisoara, Romania, 15-18 March, 2004, http://hermes.etc.utt.ro/cercetare/carti.html;

[3] C. Berrou, A. Glavieux, P. Thitimajshima - "Near Shannon limit error-correcting coding and decoding: Turbo-codes", Proc.ICC'93, Geneva, Switzerland, May 1993, pp. 1064 – 1070;

[4] L. Trifina, H.G.Balta, A. Ruşinaru, "Decreasing of the Turbo MAP Decoding Time by Using an Iterations Stopping Criterion", paper accepted onn the IEEE conference, Iasi, 14-15 July.