A Turbo Coding Scheme for the 2-User Gaussian Adder Channel

M. L. M. G. Alcoforado
Telecommunications Research Center
NPT - POLI
University of Pernambuco-UPE
Recife, PE, Brazil, 50.750-470
e-mail: mlmga@upe.poli.br

V. C. da Rocha Jr. and M. J. de Carvalho Lima
Communications Research Group - CODEC
Department of Electronics and Systems
Federal University of Pernambuco
Recife, PE, Brazil, 50711-970, PO Box 7800
Email: vcr@ufpe.br, marcio.lima@ee.ufpe.br

***Abstract*— This paper discusses a coding scheme for the 2-user binary adder channel (2-BAC) employing a pair of distinct turbo convolutional codes, not necessarily forming a uniquely decodable code pair, in the presence of additive white Gaussian noise. The decoder is able to separate the binary data for each user by exploiting the code structure to combat noise and to resolve ambiguities, i.e., to distinguish between noisy versions of the two-user pairs $(0,1)$ and $(1,0)$. By means of computer simulation the performance of a particular pair of distinct turbo convolutional codes is determined and is presented as an example. The results are considerably better than those obtained by using a serial concatenation of 2-BAC uniquely decodable codes as outer codes and identical turbo convolutional codes as inner codes in the 2-BAC.**

Keywords*— *Turbo codes; iterative decoding, multiple access channel, additive channel.

I. Introduction

The two-user binary adder channel (2-BAC) is one of the simplest models for a multiple access channel [1], [2]. The 2-BAC is a memoryless channel which, at each time interval, accepts two binary inputs, one from each user. In the absence of noise the 2-BAC output is given by the arithmetic sum of its two inputs, i.e., the output symbols belong to the alphabet $\{0,1,2\}$. In the noisy case the 2-BAC output is described by a conditional probability distribution [1].

For many years the research on coding for the 2-BAC consisted of attempting to construct pairs of codes with high sum rates, usually resorting to a computer search. More recently, references [3] and [4] present new uniquely decodable codes for the 2-BAC and for the t-user binary adder channel (t-BAC) for $3 \leq \mathrm{t} \leq 5$, respectively. These codes have the best known rates for the respective adder channels. In general, however, codes obtained in this manner have no structure that could help to simplify their decoding.

In an effort to overcome the lack of structure present in 2-BAC codes generated by computer search, in 2004 a serial concatenation technique was proposed for constructing uniquely decodable trellis codes for the 2-BAC [5], [6], where identical convolutional codes were allocated as inner codes to both users. For each of the two users the encoder consisted of a serial concatenation of a block code as the outer code, with a convolutional code as the inner code. The block code, which is one code of a pair of 2-BAC uniquely decodable binary block codes, acts as a filter to eliminate those paths in the 2-BAC trellis [7] that would otherwise lead to ambiguity at the decoder. Some 2-BAC computer simulation results were presented of a construction [8], [9] where the convolutional code employed in [6] was replaced by a turbo code [10]. Still in 2005, further computer simulation results were presented of a similar 2-BAC code construction employing however turbo block codes as inner codes [11] instead of turbo convolutional codes. It is argued in the sequel that, independent of the strategy adopted for decoding, the use of identical inner codes for both users in a 2-BAC may cause a loss in performance.

Differently of [5], [6], [8], [9] and [11] which uses a serial concatenation to get the unique decodability, our purpose in this paper is to investigate the performance of a coding scheme for the 2-BAC employing just a pair of distinct turbo convolutional codes, not necessarily constituting a uniquely decodable code pair. The efficiency of this construction is illustrated by means of computer simulation results, considering the BCJR [12] algorithm adapted for the 2-BAC case, i.e., a joint BCJR decoder. This remainder of this paper is organized as follows. In Section II we describe in detail the encoder employed, in Section III we review the main aspects of the noisy 2-BAC and a new table description of the 2-BAC trellis is introduced. In Section IV we describe the details of the iterative decoder employed and we close the paper in Section V presenting computer simulation results and some comments.

II. Encoder

Let (C_1, C_2) denote a pair of turbo convolutional codes associated with users 1 and 2 of a 2-BAC, respectively, as illustrated in Figure 1.

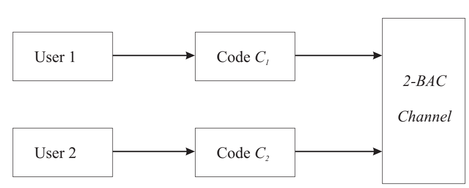

Fig. 1. Model of coding scheme for the 2-BAC.

Suppose the encoder for C_1 employs the turbo code construction of Berrou et al. [10]. It follows that the encoder for C_1 consists of the parallel concatenation of two systematic recursive binary convolutional codes, C_1^{-} and $C_1^{|}$, not necessarily identical. The respective inputs for both component encoders use the same information bits u_k, however in different order due to the use of an interleaver in one of the encoders. Similarly, the encoder for C_2 consists of the parallel concatenation of two systematic recursive binary convolutional codes, C_2^{-} and $C_2^{|}$, not necessarily identical. The inputs for both component encoders use the same information bits d_k, however in different order due to the use of an interleaver in one of the encoders, which must be identical to the interleaver employed for C_1. The transmission rate of C_1 is assumed to be equal to that of C_2. Without loss of essential generality, assume that each systematic recursive encoder has asymptotic rate $1/n$ and M states, for both users. Let the binary sequence of information symbols from user 1 be denoted as

$$\boldsymbol{u} = \boldsymbol{u}_1^N = \{u_1, u_2, \ldots, u_k, \ldots, u_N\},$$

and let the corresponding binary sequence of information symbols for user 2 be denoted as

$$\boldsymbol{d} = \boldsymbol{d}_1^N = \{d_1, d_2, \ldots, d_k, \ldots, d_N\}.$$

Let the sequence of codewords from user 1 be denoted as

$$\boldsymbol{v} = \boldsymbol{v}_1^N = \{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_k, \ldots, \boldsymbol{v}_N\},$$

and let the corresponding sequence of codewords from user 2 be denoted as

$$\boldsymbol{w} = \boldsymbol{w}_1^N = \{\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_k, \ldots, \boldsymbol{w}_N\},$$

where

$$\boldsymbol{v}_k = (v_k^{(0)}, v_k^{(1)}, \ldots, v_k^{(n-1)}) = (u_k, v_k^{(1)}, \ldots, v_k^{(n-1)}),$$

$1 \le k \le N$, denotes the output associated with each information symbol from user 1 and, similarly,

$$\boldsymbol{w}_k = (w_k^{(0)}, w_k^{(1)}, \ldots, w_k^{(n-1)}) = (d_k, w_k^{(1)}, \ldots, w_k^{(n-1)}),$$

$1 \le k \le N$, denotes the output associated with each information symbol from user 2. The symbols $v_k^{(0)}$ and $w_k^{(0)}$ denote systematic encoder outputs for user 1 and user 2, respectively.

Example 1. *Let C_1^{-} and $C_1^{|}$ denote two binary recursive systematic rate $1/2$ convolutional codes and identical polynomial generator matrices*

$$G_1(D) = \left[1 \quad \frac{1+D^2}{1+D+D^2}\right].$$

Similarly, let C_2^{-} and $C_2^{|}$ denote two binary recursive systematic rate $1/2$ convolutional codes and identical polynomial generator matrices

$$G_2(D) = \left[1 \quad \frac{D+D^2}{1+D+D^2}\right].$$

The corresponding encoders for users 1 and 2 are illustrated in Figure 2 and Figure 3, respectively.

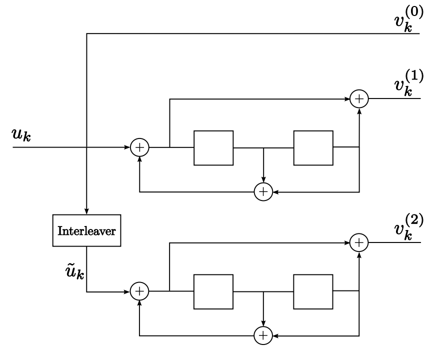

Fig. 2. Encoder for code C_1, employing two identical polynomial generator matrices, namely $\left[1 \quad \frac{1+D^2}{1+D+D^2}\right]$.

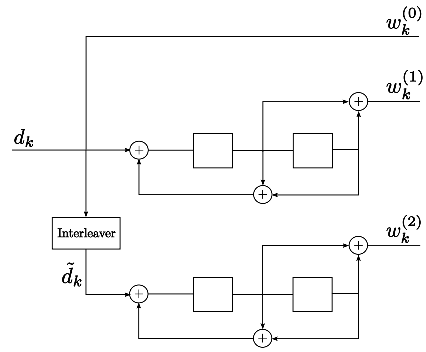

Fig. 3. Encoder for code C_2, employing two identical polynomial generator matrices, namely $\left[1 \quad \frac{D+D^2}{1+D+D^2}\right]$.

III. TWO-USER TRELLIS AND THE NOISY 2-BAC

Because the 2-BAC is defined in terms of input pairs, at any time interval the decoder must consider pairs of paths, one from each single-user trellis. The a posteriori probabilities of single paths are not defined, however, the a posteriori probabilities of path pairs are defined. This leads immediately to the concept of a two-user trellis [7]. The two-user trellis is defined such that, at any given time slot, each distinct pair of paths, one through each single-user trellis, corresponds to a unique path through the two-user trellis, each branch of the two-user trellis corresponds to a pair of branches, one in each single-user trellis, and each state of the two-user trellis corresponds to a pair of states, one in each single-user trellis, i.e., the two-user trellis state at time k is simply the contents of the two encoder shift registers. The decoder task is to discover which path along the two-user trellis is the most likely. If each single-user trellis has L_i states, $i \in \{1,2\}$, then the two-user trellis will have $L_1 L_2$ states.

Let the codeword sequences $\boldsymbol{v}_1^N$ and $\boldsymbol{w}_1^N$ be the two inputs for a memoryless 2-BAC. Assume the noise to be additive white Gaussian noise. The sequence of sub-blocks in the two-user trellis is denoted as

$$\boldsymbol{x} = \boldsymbol{x}_1^N = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_k, \ldots, \boldsymbol{x}_N\},$$

where

$$\boldsymbol{x}_k = (x_k^{(0)}, x_k^{(1)}, \ldots, x_k^{(n-1)}).$$

The random variable $x_k^{(j)},\ j = 0, \ldots, n-1$, at time instant k, is defined as

$$x_k^{(j)} = (1 - 2v_k^{(j)}) + (1 - 2w_k^{(j)}),\ j = 0, \ldots, n-1.$$

The 2-BAC output sequence is denoted as

$$\boldsymbol{r} = \boldsymbol{r}_1^N = \{\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_k, \ldots, \boldsymbol{r}_N\},$$

where $\boldsymbol{r}_k = (r_k^{(0)}, r_k^{(1)}, \ldots, r_k^{(n-1)})$. The random variable $r_k^{(j)},\ j = 0, \ldots, n-1$, at time instant k, is defined as

$$r_k^{(j)} = x_k^{(j)} + q_k^{(j)},\ j = 0, \ldots, n-1,$$

TABLE I
OUTPUT SUB-BLOCKS AND TRELLIS STATES FOR THE 2-USER TRELLIS IN EXAMPLE 2.

	$(u_k, d_k) \rightarrow$			
$S_{k-1} \downarrow$	$(+1+1)$	$(-1+1)$	$(-1-1)$	$(+1-1)$
0000	+2+2, 0000	0 0, 1000	-2 0, 1010	0+2, 0010
0001	+2 0, 0010	0-2, 1010	-2-2, 1000	0 0, 0000
0010	+2 0, 0011	0-2, 1011	-2-2, 1001	0 0, 0001
0011	+2+2, 0001	0 0, 1001	-2 0, 1011	0+2, 0011
0100	+2+2, 1000	0 0, 0000	-2 0, 0010	0+2, 1010
0101	+2 0, 1010	0-2, 0010	-2-2, 0000	0 0, 1000
0110	+2 0, 1011	0-2, 0011	-2-2, 0001	0 0, 1001
0111	+2+2, 1001	0 0, 0001	-2 0, 0011	0+2, 1011
1000	+2 0, 1100	0+2, 0100	-2+2, 0110	0 0, 1110
1001	+2-2, 1110	0 0, 0110	-2 0, 0100	0-2, 1100
1010	+2-2, 1111	0 0, 0111	-2 0, 0101	0-2, 1101
1011	+2 0, 1101	0+2, 0101	-2+2, 0111	0 0, 1111
1100	+2 0, 0100	0+2, 1100	-2+2, 1110	0 0, 0110
1101	+2-2, 0110	0 0, 1110	-2 0, 1100	0-2, 0100
1110	+2-2, 0111	0 0, 1111	-2 0, 1101	0-2, 0101
1111	+2 0, 0101	0+2, 1101	-2+2, 1111	0 0, 0111

where the values $q_k^{(j)}$ represent independent noise samples with identical variance σ^2 and zero mean value.

Example 2. *Consider again the situation presented in Example 1. Figure 4 presents a section of the 2-user trellis, showing two states at time instant $k-1$, namely $S_{k-1} = 0110$ and $S_{k-1} = 1111$, and branches connecting them to their corresponding states at time instant $t = k$ denoted by S_k. In $S_{k-1} = 0110$, for example, the first two bits 01 corresponds to the contents of the C_1^- or $C_1^|$ shift register, and the last two bits 10 corresponds to the contents of the C_2^- or $C_2^|$ shift register. Each branch has a label which indicates the pair (u_k, d_k) of input symbols, followed by the noiseless 2-BAC output $\boldsymbol{x}_k = x_k^0, x_k^1$, i.e., $(u_k, d_k)/x_k^0, x_k^1$. A simple and compact way of representing the full trellis for 2-users is presented in Table I, where the 2-BAC inputs are binary digits from the alphabet $\{-1+1\}$, and the output is ternary from the alphabet $\{-2, 0, +2\}$. Each cell in Table I contains the corresponding noiseless 2-BAC output x_k^0, x_k^1 followed by the current trellis state S_k, when the row label is the trellis state in the previous time instant S_{k-1} and the column label is the 2-user pair of inputs (u_k, d_k). For example, if $S_{k-1} = 0110$ and $(u_k, d_k) = (-1, +1)$, then it follows from Table I that $\boldsymbol{x}_k = 0, -2$ and $S_k = 0011$. If $S_{k-1} = 1111$ and $(u_k, d_k) = (+1, -1)$, then it follows that $\boldsymbol{x}_k = 0, 0$ and $S_k = 0111$.*

IV. THE DECODER

The decoder employed, illustrated in Figure 5, uses iterative decoding [13] to detect the most likely pairs (u_k, d_k) of binary information symbols. The iterative algorithm employed uses the BCJR technique [12], adapted for use in the 2-BAC [8], making use of the 2-user trellis.The log-likelihood ratios $\Lambda_1(u_k, d_k)$, $\Lambda_2(u_k, d_k)$ and $\Lambda_3(u_k, d_k)$ associated with the pair (u_k, d_k) of information symbols from users 1 and 2, respectively, are computed by means of the following expressions.

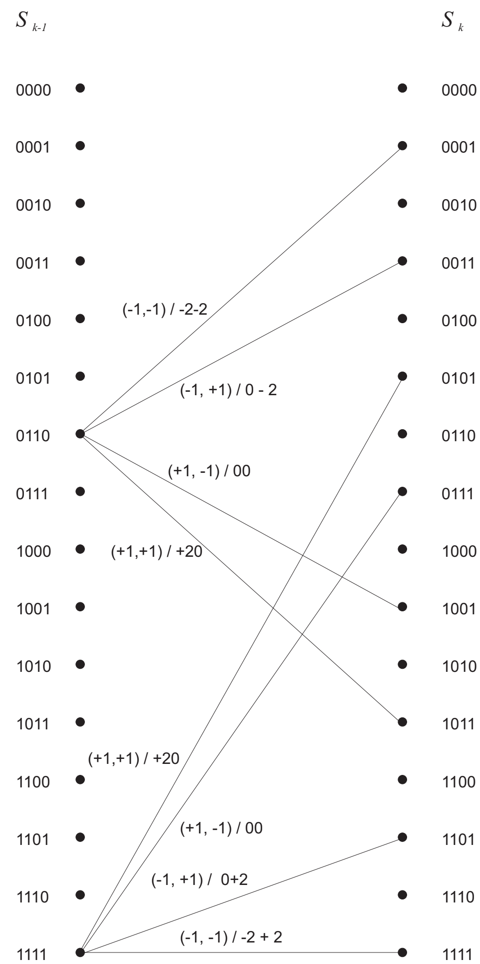

Fig. 4. Branches of the 2-user trellis connecting states $S_{k-1} = 0110$ and $S_{k-1} = 1111$, at time instant $t = k-1$, to the corresponding states S_k at time instant $t = k$. For example, if $S_{k-1} = 0110$ and $S_k = 0011$, then $(u_k, d_k)/\boldsymbol{x}_k = (-1, +1)/0-2$.

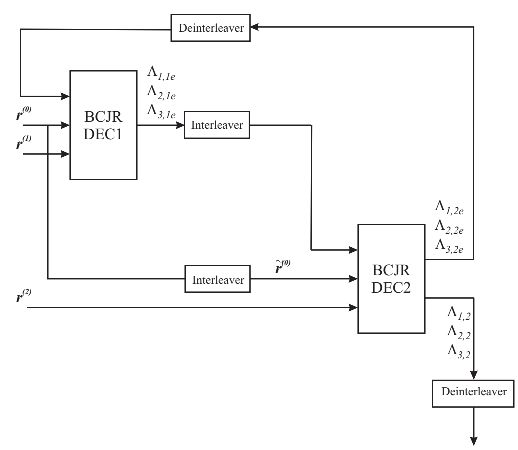

Fig. 5. Iterative decoder, containing two component decoders connected in series.

$$\Lambda_1(u_k, d_k) = \log \frac{P\{u_k = -1, d_k = +1 | \boldsymbol{r}\}}{P\{u_k = +1, d_k = +1 | \boldsymbol{r}\}}, \quad (1)$$

$$\Lambda_2(u_k, d_k) = \log \frac{P\{u_k = -1, d_k = -1 | \boldsymbol{r}\}}{P\{u_k = +1, d_k = +1 | \boldsymbol{r}\}}, \quad (2)$$

$$\Lambda_3(u_k, d_k) = \log \frac{P\{u_k = +1, d_k = -1 | \boldsymbol{r}\}}{P\{u_k = +1, d_k = +1 | \boldsymbol{r}\}}, \quad (3)$$

where $\mathrm{P}\{u_k = i, d_k = s|\boldsymbol{r}\}$, $i \in \{-1,+1\}$, $s \in \{-1,+1\}$ denotes the *a posteriori* probability of the pair (u_k, d_k) of information symbols, given the received 2-BAC sequence $\boldsymbol{r}$.

The decoder operates as follows. The input to the first BCJR decoder, denoted by the block labeled as DEC1 in Figure 5, is fed with the received sequences $\boldsymbol{r}^{(0)} = \{r_1^{(0)}, r_2^{(0)}, \ldots, r_N^{(0)}\}$ and $\boldsymbol{r}^{(1)} = \{r_1^{(1)}, r_2^{(1)}, \ldots, r_N^{(1)}\}$, where $r_k^{(j)}$ was defined earlier. DEC1 produces the soft outputs $\Lambda_{1,1}(u_k,d_k), \Lambda_{2,1}(u_k,d_k)$ and $\Lambda_{3,1}(u_k,d_k)$, which are interleaved and are used to produce estimates of the *a priori* probabilities of pairs of information sequences to be fed as inputs to the second BCJR decoder, denoted by the block labeled as DEC2 in Figure 5. The notation $\Lambda_{1,1}(u_k,d_k)$, $\Lambda_{2,1}(u_k,d_k)$, $\Lambda_{3,1}(u_k,d_k)$ is used to indicate the soft outputs $\Lambda_1(u_k,d_k), \Lambda_2(u_k,d_k)$ and $\Lambda_3(u_k,d_k)$ associated with DEC1, respectively. The values $\Lambda_{1,1e}(u_k,d_k)$, $\Lambda_{2,1e}(u_k,d_k)$ and $\Lambda_{3,1e}(u_k,d_k)$ represent the extrinsic information to decoder DEC1.

The input to decoder DEC2 receives the sequences $\tilde{\boldsymbol{r}}^{(0)}$ and $\boldsymbol{r}^{(2)} = \{r_1^{(2)}, r_2^{(2)}, \ldots, r_N^{(2)}\}$. The sequence $\tilde{\boldsymbol{r}}^{(0)}$ corresponds to the sequence $\boldsymbol{r}^{(0)}$ interleaved. Decoder DEC2 also produces soft outputs $\Lambda_1(u_k,d_k), \Lambda_2(u_k,d_k)$ and $\Lambda_3(u_k,d_k)$, denoted as $\Lambda_{1,2}(u_k,d_k), \Lambda_{2,2}(u_k,d_k)$ and $\Lambda_{3,2}(u_k,d_k)$, to indicate the fact that they are associated with DEC2, respectively. These soft outputs are used to improve the estimates of the *a priori* probabilities of pairs of information bit sequences (u_k,d_k) input to decoder DEC1. Decoder DEC2 estimates the log likelihood ratios $\Lambda_{1,2}(u_k,d_k)$, $\Lambda_{2,2}(u_k,d_k)$ and $\Lambda_{3,2}(u_k,d_k)$. The values $\Lambda_{1,2e}(u_k,d_k)$, $\Lambda_{2,2e}(u_k,d_k)$ and $\Lambda_{3,2e}(u_k,d_k)$ represent the extrinsic information to decoder DEC2. Such values depend on the redundant information supplied by the encoders $C_1^{|}$ and $C_2^{|}$. The extrinsic information for decoder DEC2 is used as an estimate of the *a priori* probabilities to decoder DEC1. The values $\hat{\Lambda}_{1,2e}(u_k,d_k)$, $\hat{\Lambda}_{2,2e}(u_k,d_k)$ and $\hat{\Lambda}_{3,2e}(u_k,d_k)$ correspond, respectively, to the values of $\Lambda_{1,2e}(u_k,d_k)$, $\Lambda_{2,2e}(u_k,d_k)$ and $\Lambda_{3,2e}(u_k,d_k)$ when deinterleaved.

The use of identical error-correcting codes, in a systematic manner for both users in the 2-BAC is possible only when a serial concatenation construction, e.g., as in [8] and [11], is employed. However, it follows for this construction technique that the log likelihood ratios $\Lambda_1(u_k,d_k)$ and $\Lambda_3(u_k,d_k)$ are equal, i.e., $\mathrm{P}\{u_k=+1,d_k=-1|\boldsymbol{r}\} = \mathrm{P}\{u_k=-1,d_k=+1|\boldsymbol{r}\}$. This latter condition forbids any trellis decoder of separating the symbols sent by each user in the 2-BAC, except for the trivial cases, i.e., where $u_k = d_k$. The inner decoder outputs its best estimate of a ternary sequence which is then fed to the outer 2-BAC decoder in the serial concatenation. Since the outer codes are not error-correcting codes, any error not corrected by the inner code probably will not be corrected by the outer decoder. This situation leads to a loss in performance.

We notice that the key point which allows the decoder used in this paper to separate from the received noisy sequence the two binary sequences, one for each of the two users, is the fact that, in general,

$$\Lambda_1(u_k,d_k) \neq \Lambda_3(u_k,d_k)$$

or, equivalently,

$$\mathrm{P}\{u_k=+1,d_k=-1|\boldsymbol{r}\} \neq \mathrm{P}\{u_k=-1,d_k=+1|\boldsymbol{r}\}$$

when the codes for user 1 and user 2 are distinct. It follows that the potential ambiguity resulting from the pairs $(u_k = +1, d_k = -1)$ and $(u_k = -1, d_k = +1)$ will be resolved by this joint decoder most of the time. The use of distinct codes thus allows both the correction of errors due to noise and the correction of errors due to the interference between users.

V. Results and Final Considerations

We considered for the computer simulation the situation where the code for user 1 and the code for user 2 are distinct.The standard Berrou and Glavieux [10] interleaver was employed with blocklength 512. The curves obtained, relating the bit error probability and signal to noise ratio for user 1 and for user 2 are illustrated in Figure 6 and Figure 7, respectively. The encoders for $C_1^- = C_1^{|}$ in Figure 6 have polynomial generator matrices $G(D) = \begin{bmatrix} 1 & \frac{1+D^2}{1+D+D^2} \end{bmatrix}$ and the encoders for $C_2^- = C_2^{|}$ in Figure 7 have polynomial generator matrices $G(D) = \begin{bmatrix} 1 & \frac{D+D^2}{1+D+D^2} \end{bmatrix}$. The two curves labeled “Without Turbo”, one in Figure 6 and one in Figure 7, are obtained when the decoder recovers the binary data for each user by running the received sequence $\boldsymbol{r}$ through DEC 1 only. As expected, when iterative decoding is employed this construction shows a significant performance improvement with respect to the no-iterations (without turbo) case, with a gain of approximately 3dB for a bit error probability of approximately 10^{-2} by using 2 iterations, for both users. For purpose of comparison, observing the curves produced with a serial concatenated coding scheme for the 2-BAC as described in [9], having sum rate $1.29 \times (1/3) = 0.43$, the results presented in Figure 6 and Figure 7 are considerably better, although the turbo convolutional codes employed have the same memory as those employed in [9].

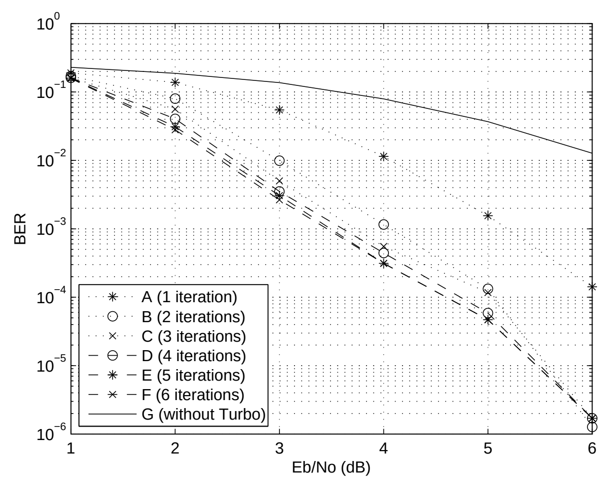

Fig. 6. Bit error rate for user 1, for a rate 1/3 turbo convolutional code with polynomial generator matrix $[1 \quad (1+D^2)/(1+D+D^2)]$.

The advantage of this construction comes from the possibility of directly separating the binary data for each of the two users at the receiver by using the BCJR [12] iterative decoding algorithm applied to the received sequence $\boldsymbol{r}$ and the corresponding 2-BAC trellis. In [14] is adressed the problem of designed good low density parity-check codes (LDPC) for the Gaussian multiple access channel but long codes are not attractive for some practical cases due to synchronism lost. In these cases, short codes are desirable. Recently [15], LDPC codes have been investigated for the 2-BAC and the 3-BAC by means of computer simulation and the results for the 2-BAC are comparable to those presented here.

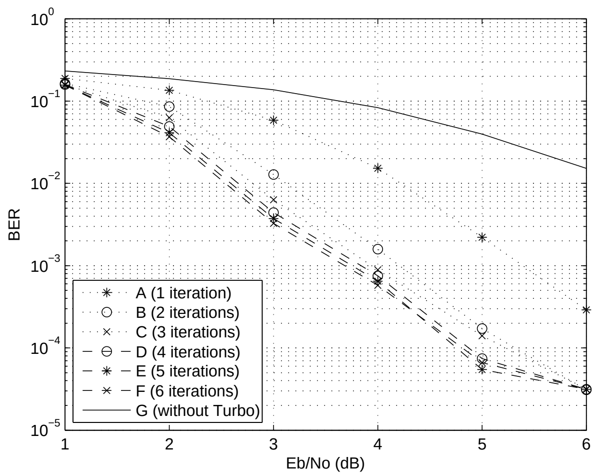

Fig. 7. Bit error rate for user 2, for a rate $1/3$ turbo convolutional code with polynomial generator matrix $[1 \quad (D+D^2)/(1+D+D^2)]$.

Acknowledgements

This work received partial support from the Brazilian Ministry of Education (CAPES) and from the Brazilian Ministry of Science and Technology (CNPq), through projects 306612/2007-0, 555046/2008-7 and 141346/2008-6. The authors are grateful to their colleague Humberto V. Beltrão Neto for helpful technical discussions and suggestions, and for helping with the preparation of this paper.

References

[1] T. Kasami and Shu Lin, "Coding for a multiple-access channel", *IEEE Trans. on Inform. Theory*, vol. 22, Number 2, pp.129-137, March 1976.

[2] R. G. Gallager, "A Perpective on Multiaccess Channels", *IEEE Trans. on Info. Theory*, vol. IT-31, Number 2, pp. 124-142, March 1985.

[3] M. Mattas and P. R. J. Ostergard, "A new bound for the zero-error capacity region of the two-user binary adder channel", *IEEE Trans. on Info. Theory*, vol. 51, pp. 3289-3291, 2005.

[4] L. Kiviluoto and P. R. J. Ostergard, "New uniquely decodable codes for the t-user binary adder channel with 3≤ t ≤ 5", *IEEE Trans. on Info. Theory*, vol. 53, pp. 1219-1220, 2007.

[5] V. C. da Rocha Jr. and M. L. M. G. Alcoforado, "Trellis code construction for the 2-user binary adder channel", Lecture Notes in Computer Science, Springer Verlag, Germany, v. 3124, pp. 122-127, 2004.

[6] V. C. da Rocha Jr. and M. L. M. G. Alcoforado, "Uniquely decodable trellis codes for the two user binary adder channel", *International Symposium on Information Theory and its Applications*, 10-13 October 2004, Parma, Italy.

[7] R. Peterson and D. J. Costello, Jr., "Binary convolutional codes for a multiple-access channel", *IEEE Trans. on Info. Theory*, vol. 25, no. 1, pp. 101-105, 1979.

[8] M. L. M. G. Alcoforado, V. C. da Rocha Jr. and G. Markarian, "Turbo convolutional codes for the binary adder channel" *VIII International Symposium on Communication Theory and Applications*, Julho, Ambleside, UK, pp. 142-147, 2005.

[9] M. L. M. G. Alcoforado, V. C. da Rocha Jr. and G. Markarian "Simulations using trellis codes and iterative decoding for the binary adder channel", Proceedings of the *XXV Brazilian Telecommunications Symposium*, Recife, Brazil, 2007. (*in Portuguese*)

[10] C. Berrou, A. Glavieux and P. Thitimajshima,"Near Shannon limit, error-correcting coding and decoding: turbo codes", *IEEE International Conference on Communications (ICC'93)*, vol. 2/3, pp. 1064-1071, 1993.

[11] M. L. M. G. Alcoforado, V. C. da Rocha Jr. and G. Markarian "Turbo block codes for the binary adder channel" *2005 IEEE International Symposium on Information Theory*, Adelaide, Australia, pp. 1883-1887, 2005.

[12] L. R. Bahl, J. Cocke, F. Jelinek and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate", *IEEE Trans. Inform. Theory*, vol. 20, pp. 284-287, 1974.

[13] J. Hagenauer, "Iterative decoding of binary block and convolutional codes", *IEEE Trans. on Inform. Theory*, vol. 42, no. 2, pp. 429-445, 1996.

[14] A. Roumy and D. Declerq, "Characterization and optimization of LDPC codes for the 2-User Gaussian multiple access channel", *EURASIP Journal on Wireless Communications and Networking*, 2007.

[15] H.V. Beltrão Neto and V. C. da Rocha Jr., "Iterative decoding results for the Gaussian multiuser binary adder channel", *Tenth International Symposium on Communication Theory and Applications, Ambleside, 2009*, Ambleside, UK, pp. 1-6, 2009.