

The Bahl-Cocke-Jelinek-Raviv Algorithm Applied to the Two-User Binary Adder Channel

Maria de Lourdes M. G. Alcoforado, Valdemar C. da Rocha Jr.

Abstract—The Bahl-Cocke-Jelinek-Raviv (BCJR) algorithm is a well known maximum *a posteriori* probability decoding algorithm which has been proposed earlier for point to point communication applications, employing block codes or convolutional codes, and turbo codes. This paper describes an application of the BCJR algorithm for decoding the output of a multiple access channel called the noisy two-user binary adder channel, in the presence of additive white Gaussian noise.

Index Terms—Error control coding, BCJR algorithm, multiple access channel, turbo code, two-user binary adder channel.

I. INTRODUCTION

THE Bahl-Cocke-Jelinek-Raviv (BCJR) algorithm [1] is a well known maximum *a posteriori* probability decoding algorithm which has been proposed for point to point communication applications, employing block codes or convolutional codes [1], and turbo codes [2]. In this paper we apply the BCJR decoding algorithm to the two-user communication system illustrated in Figure 1, where the channel is the two-user binary adder channel (2-BAC) [3] with the addition of noise. Convolutional codes are considered in the sequel but the treatment for block codes is similar. In Figure 1 the two sources are assumed to produce equally likely 0's and 1's and the two encoders are binary convolutional encoders. For transmission purposes the mapping $0 \rightarrow 1, 1 \rightarrow -1$ is applied to the components in \mathbf{r} and \mathbf{s} to produce the vectors \mathbf{v} and \mathbf{w} , respectively, by means of the expressions $v_t^{(l)} = (1 - 2r_t^{(l)})$ and $w_t^{(l)} = (1 - 2s_t^{(l)})$, for $0 \leq l \leq n-1$, where $v_t^{(l)}, r_t^{(l)}, w_t^{(l)}$ and $s_t^{(l)}$ denote components of the vectors $\mathbf{v}, \mathbf{r}, \mathbf{w}$ and \mathbf{s} , respectively. In the noiseless case the 2-BAC output y_t , at time t , is just the arithmetic sum of the binary inputs v_t and w_t , i.e., $y_t = v_t + w_t \in \{-2, 0, 2\}$, while in the noisy case $y_t = v_t + w_t + q_t$ and the noisy 2-BAC output behaviour is described by the conditional probability $P(y_t | v_t w_t)$, where q_t denotes a sample of a time-discrete noise process which here we assume to be additive white Gaussian noise (AWGN). In Figure 1, consider for each user a binary rate k/n convolutional encoder, where k and n are positive integers and $k < n$. For simplicity we assume each encoder has overall constraint length $k\nu$ and can be implemented by k shift registers, each of length ν . Let $\mathbf{u} = \mathbf{u}_1^N = \{u_1, u_2, \dots, u_t, \dots, u_N\}$ and $\mathbf{d} = \mathbf{d}_1^N = \{d_1, d_2, \dots, d_t, \dots, d_N\}$ denote input binary data sequences of length N from users 1 and 2, respectively. Let $\mathbf{r} = \mathbf{r}_1^N =$

$\{r_1, r_2, \dots, r_t, \dots, r_N\}$ and $\mathbf{s} = \mathbf{s}_1^N = \{s_1, s_2, \dots, s_t, \dots, s_N\}$ denote binary (0, 1) encoder output sequences for users 1 and 2, respectively, where $\mathbf{r}_t = (r_t^{(0)}, r_t^{(1)}, \dots, r_t^{(n-1)})$ and $\mathbf{s}_t = (s_t^{(0)}, s_t^{(1)}, \dots, s_t^{(n-1)})$ denote the output subblocks associated with information blocks $\mathbf{u}_t = (u_t^{(0)}, u_t^{(1)}, \dots, u_t^{(k-1)})$ and $\mathbf{d}_t = (d_t^{(0)}, d_t^{(1)}, \dots, d_t^{(k-1)})$, respectively.

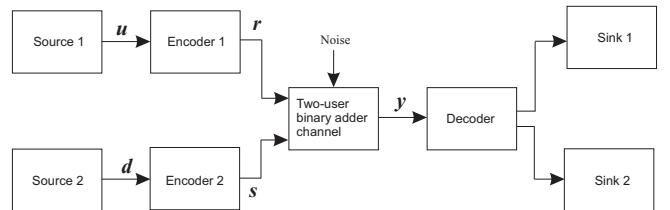


Fig. 1. Two-user communication system, where the channel is a noisy two-user binary adder channel (2-BAC).

In order to use error-correcting codes in a noisy 2-BAC and avoid the well known ambiguity resulting from the input pairs $(u_t = 0, d_t = 1)$ and $(u_t = 1, d_t = 0)$, a construction was proposed in [4] which employs the same error-correcting code in systematic form, for both users in Figure 1, in a serial concatenation with noiseless 2-BAC codes. However, the construction described in [4] still leads to a form of ambiguity expressed as the conditional probability equality $P\{u_t = 0, d_t = 1 | y\} = P\{u_t = 1, d_t = 0 | y\}$, which forbids the decoder of separating the symbols sent by each user in the 2-BAC at time instant t , except for the trivial cases, i.e., where $u_t = d_t$. For this reason henceforth we consider always distinct turbo codes for user 1 and user 2, respectively. It follows that the potential ambiguity resulting from the input pairs $(u_t = 0, d_t = 1)$ and $(u_t = 1, d_t = 0)$ will be resolved by a 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.

II. TWO-USER TRELLIS CONSTRUCTION

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 [5]. 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

M.L.M.G. Alcoforado is with Telecommunication Research Group - NPT, University of Pernambuco (UPE), Recife, 50.750-470, PE, BRAZIL (e-mail: mlmga@poli.br).

V.C.da Rocha Jr. is with Communication Research Group - CODEC, Department of Electronics and Systems, Federal University of Pernambuco (UFPE), PO Box 7800, Recife 50711-970, PE, BRAZIL (e-mail: vcr@ufpe.br).

$$\Lambda_{2i+j,t} = \log \frac{\sum_m \sum_{m'} \mathbf{P}\{u_t = i, d_t = j, \sigma_t = m, \sigma_{t-1} = m', \mathbf{y}_1^{t-1}, \mathbf{y}_t, \mathbf{y}_{t+1}^N\}}{\sum_m \sum_{m'} \mathbf{P}\{u_t = 0, d_t = 0, \sigma_t = m, \sigma_{t-1} = m', \mathbf{y}_1^{t-1}, \mathbf{y}_t, \mathbf{y}_{t+1}^N\}}. \quad (1)$$

$$\Lambda_{2i+j,t} = \log \frac{\sum_m \sum_{m'} \mathbf{P}\{\mathbf{y}_{t+1}^N | \sigma_t = m\} \mathbf{P}\{\sigma_{t-1} = m', \mathbf{y}_1^{t-1}\} \mathbf{P}\{u_t = i, d_t = j, \sigma_t = m, \mathbf{y}_t | \sigma_{t-1} = m'\}}{\sum_m \sum_{m'} \mathbf{P}\{\mathbf{y}_{t+1}^N | \sigma_t = m\} \mathbf{P}\{\sigma_{t-1} = m', \mathbf{y}_1^{t-1}\} \mathbf{P}\{u_t = 0, d_t = 0, \sigma_t = m, \mathbf{y}_t | \sigma_{t-1} = m'\}}. \quad (2)$$

single-user trellis, i.e., the two-user trellis state σ_t at time t is simply the contents of the two encoder shift registers. Representing the state of each encoder as a $k\nu$ -tuple, we have $\sigma_t = (\sigma_t^{(1)}, \sigma_t^{(2)}, \dots, \sigma_t^{(k\nu)}, \sigma_t^{(k\nu+1)}, \dots, \sigma_t^{(2k\nu)})$.

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 $\mathbf{v} + \mathbf{w} = \mathbf{x} = \mathbf{x}_1^N = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_t, \dots, \mathbf{x}_N\}$ denote a ternary $(-2, 0, 2)$ sequence of sub-blocks through the two-user trellis or, equivalently, let \mathbf{x} denote the output of the noiseless 2-BAC, where $\mathbf{x}_t = (x_t^{(0)}, x_t^{(1)}, \dots, x_t^{(n-1)})$ and $x_t^{(l)} = v_t^{(l)} + w_t^{(l)}$, $0 \leq l \leq n-1$. The noisy 2-BAC output is the received vector $\mathbf{y} = \mathbf{y}_1^N = \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_t, \dots, \mathbf{y}_N\}$, where $\mathbf{y}_t = (y_t^{(0)}, y_t^{(1)}, \dots, y_t^{(n-1)})$, $y_t^{(l)} = x_t^{(l)} + q_t^{(l)}$, $0 \leq l \leq n-1$.

Example 1: Suppose that recursive rate 1/2 convolutional codes with encoders 1 and 2 in Figure 1 have, respectively, transfer function matrices $G_1(D) = [1 \quad 1/(1+D)]$ and $G_2(D) = [1 \quad D/(1+D)]$. Each single-user trellis has two states and the two-user trellis has four states as shown in Figure 2. The branch labels in Figure 2 mean values of $u_t, d_t/x_t^{(0)} x_t^{(1)}$, for example, $1, 1/-2+2$ connecting states 01 and 10 means that $(u_2, d_2) = (1, 1)$ and $(x_2^{(0)}, x_2^{(1)}) = (-2, +2)$.

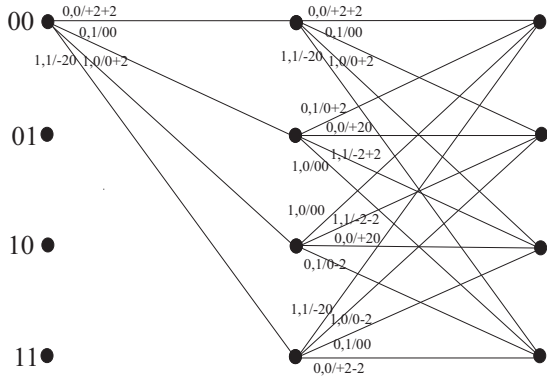


Fig. 2. Two-user trellis using rate 1/2 convolutional codes with transfer function matrices $G_1(D) = [1 \quad 1/(1+D)]$ and $G_2(D) = [1 \quad D/(1+D)]$.

III. BCJR DECODING FOR THE 2-BAC

The decoding problem for the noisy 2-BAC for a long time was left aside and the main goal in code construction for the 2-BAC has been in most cases just to achieve high rates. Perhaps the most serious difficulty for the practical use of the 2-BAC model has been precisely the decoding problem. In the sequel for simplicity we will consider systematic convolutional codes

of rate $1/n$, and will write $P_t(i, j)$ to mean the joint probability $P\{u_t = i, d_t = j\}$ at time slot t , and will write $P_t(i, j|\mathbf{y})$ to mean the *a posteriori* probability $P\{u_t = i, d_t = j|\mathbf{y}\}$ at time slot t given the received vector \mathbf{y} . We employ the two-user trellis [5], as described earlier, and compute the log-likelihood ratios $\Lambda_1(u_t, d_t)$, $\Lambda_2(u_t, d_t)$ and $\Lambda_3(u_t, d_t)$ associated with each decoded information pair (u_t, d_t) , defined as follows.

$$\Lambda_1(u_t, d_t) = \log[P_t(0, 1|\mathbf{y})/P_t(0, 0|\mathbf{y})], \quad (3)$$

$$\Lambda_2(u_t, d_t) = \log[P_t(1, 0|\mathbf{y})/P_t(0, 0|\mathbf{y})], \quad (4)$$

$$\Lambda_3(u_t, d_t) = \log[P_t(1, 1|\mathbf{y})/P_t(0, 0|\mathbf{y})]. \quad (5)$$

For simplifying notation we will denote $\Lambda_1(u_t, d_t)$, $\Lambda_2(u_t, d_t)$ and $\Lambda_3(u_t, d_t)$ by $\Lambda_{1,t}$, $\Lambda_{2,t}$ and $\Lambda_{3,t}$, respectively. For $1 \leq t \leq N$ the decoder makes a decision on (u_t, d_t) by comparing the $P\{u_t = i, d_t = j|\mathbf{y}\}$, $i \in \{0, 1\}$, $j \in \{0, 1\}$, as follows. Make $\hat{u}_t = u$ and $\hat{d}_t = d$, if $P_t(u, d|\mathbf{y}) \geq P_t(i, j|\mathbf{y})$, $i \neq u$, $j \neq d$; where $u \in \{0, 1\}$, $d \in \{0, 1\}$. Let σ_t denote the two-user trellis state at time slot t and let M denote the number of distinct trellis states, i.e., let $\#\{\sigma_t\} = M$. Let $\lambda_t^{i,j}(m) = P\{u_t = i, d_t = j, \sigma_t = m|\mathbf{y}_1^N\}$ or, equivalently, let

$$\lambda_t^{i,j}(m) = \frac{P\{u_t = i, d_t = j, \sigma_t = m, \mathbf{y}_1^N\}}{P\{\mathbf{y}_1^N\}}. \quad (6)$$

The *a posteriori* probability of a decoded data pair (u_t, d_t) , denoted as $P\{u_t = i, d_t = j|\mathbf{y}_1^N\}$, can be expressed in terms of $\lambda_t^{i,j}(m)$ as $P\{u_t = i, d_t = j|\mathbf{y}_1^N\} = \sum_{m=0}^{M-1} \lambda_t^{i,j}(m)$, $i \in \{0, 1\}$, $j \in \{0, 1\}$, and thus equalities (3), (4) and (5) can be written as follows.

$$\Lambda_{2i+j,t} = \log \frac{\sum_{m=0}^{M-1} \lambda_t^{i,j}(m)}{\sum_{m=0}^{M-1} \lambda_t^{0,0}(m)}, \quad (7)$$

where $(i, j) \in \{(0, 1), (1, 0), (1, 1)\}$. From (6) and (7) it follows the equation (1).

By taking into account that events after time t are not influenced by observation \mathbf{y}_1^t and the pair of bits (u_t, d_t) if state σ_t is known, it follows that (1) can be written as the equation (2).

Following [1], let us introduce the probability functions $\alpha_t(m)$, $\beta_t(m)$ and $\gamma_{i,j}(\mathbf{y}_t, m', m)$ as follows.

$$\alpha_t(m) = P\{\sigma_t = m|\mathbf{y}_1^t\}, \quad (8)$$

$$\beta_t(m) = \frac{P\{\mathbf{y}_{t+1}^N | \sigma_t = m\}}{P\{\mathbf{y}_{t+1}^N | \mathbf{y}_1^t\}}, \quad (9)$$

$$\gamma_{i,j}(\mathbf{y}_t, m', m) = P\{u_t = i, d_t = j, \sigma_t = m, \mathbf{y}_t | \sigma_{t-1} = m'\}. \quad (10)$$

Now we can express (2) in terms of $\alpha_t(m)$, $\beta_t(m)$ and

$\gamma_{i,j}(\mathbf{y}_t, m', m)$ as follows.

$$\Lambda_{2i+j,t} = \log \frac{\sum_m \sum_{m'} \gamma_{i,j}(\mathbf{y}_t, m', m) \alpha_{t-1}(m') \beta_t(m)}{\sum_m \sum_{m'} \gamma_{0,0}(\mathbf{y}_t, m', m) \alpha_{t-1}(m') \beta_t(m)}, \quad (11)$$

where $(i, j) \in \{(0, 1), (1, 0), (1, 1)\}$, and $\alpha_t(m)$ can be recursively calculated [2], for $1 \leq t \leq N$, from $\gamma_{i,j}(\mathbf{y}_t, m', m)$ as

$$\alpha_t(m) = \frac{\sum_{m'} \sum_{i=0}^1 \gamma_{i,j}(\mathbf{y}_t, m', m) \alpha_{t-1}(m')}{\sum_m \sum_{m'} \sum_{i=0}^1 \gamma_{i,j}(\mathbf{y}_t, m', m) \alpha_{t-1}(m')}. \quad (12)$$

Considering that for $t = 0$ the initial state for the two-user trellis is $\sigma_0 = \mathbf{0}$, we have the boundary conditions

$$\alpha_0(0) = 1 \text{ and } \alpha_0(m) = 0, \text{ for } m \neq 0. \quad (13)$$

Similarly, $\beta_t(m)$ can be recursively calculated for $t = 1, 2, \dots, N-1$ as

$$\beta_t(m) = \frac{\sum_{m'} \sum_{i=0}^1 \gamma_{i,j}(\mathbf{y}_{t+1}, m', m') \beta_{t+1}(m')}{\sum_m \sum_{m'} \sum_{i=0}^1 \gamma_{i,j}(\mathbf{y}_{t+1}, m', m) \alpha_t(m')}. \quad (14)$$

The appropriate boundary conditions, when we have $\sigma_N = \mathbf{0}$ are

$$\beta_N(0) = 1 \text{ and } \beta_N(m) = 0, \text{ for } m \neq 0. \quad (15)$$

However, since not always the encoder reaches the state $\sigma_N = \mathbf{0}$, the following conditions can be used

$$\beta_N(0) = 1/M \text{ and } \beta_N(m) = 0, \text{ for } m \neq 0. \quad (16)$$

From (10) it follows that $\gamma_{i,j}(\mathbf{y}_t, m', m) = P\{\sigma_t = m | \sigma_{t-1} = m'\} P\{\mathbf{y}_t | u_t = i, d_t = j, \sigma_t = m, \sigma_{t-1} = m'\} P\{u_t = i, d_t = j | \sigma_t = m, \sigma_{t-1} = m'\}$, where $\Pr\{\sigma_t = m | \sigma_{k-1} = m'\}$ denotes the *state transition probabilities of the two-user trellis*, defined by the encoder input statistics. We assume that $P_t(0, 0) = P_t(1, 0) = P_t(0, 1) = P_t(1, 1) = 1/4$, and since there are four possible transitions from each trellis state we assume that $P\{\sigma_t = m | \sigma_{k-1} = m'\} = 1/4$ for each of these transitions; $P\{\mathbf{y}_t | u_t = i, d_t = j, \sigma_t = m, \sigma_{t-1} = m'\}$ denotes the *transition probabilities of the time-discrete Gaussian memoryless 2-BAC* which we write as $P\{\mathbf{y}_t | u_t = i, d_t = j, \sigma_t = m, \sigma_{t-1} = m'\} = P\{\mathbf{y}_t | \mathbf{x}_t\}$, where $P\{\mathbf{y}_t | \mathbf{x}_t\} = \prod_{l=0}^{n-1} P\{y_t^{(l)} | x_t^{(l)}\}$, and $P\{y_t^{(l)} | x_t^{(l)}\} = (\sigma\sqrt{2\pi})^{-1} \exp[-(y_t^{(l)} - x_t^{(l)})^2 / 2\sigma^2]$; $P\{u_t = i, d_t = j | \sigma_t = m, \sigma_{t-1} = m'\}$ is equal to either 0 or 1. It follows that $P\{y_t^{(l)} | x_t^{(l)} = a\} = P\{r_t^{(l)} | u_t^{(l)} = i, d_t^{(l)} = j\} = (\sigma\sqrt{2\pi})^{-1} \exp[-(y_t^{(l)} - a)^2 / 2\sigma^2]$, where $a = 2$ if $(i, j) = (0, 0)$, $a = 0$ if either $(i, j) = (0, 1)$ or $(i, j) = (1, 0)$, and $a = -2$ if $(i, j) = (1, 1)$.

Let $B_t^{i,j}$ denote the set of transitions $\sigma_{t-1} = m' \rightarrow \sigma_t = m$ that are caused by the input pair $u_t = i, d_t = j$. We can further express $\gamma_{i,j}(\mathbf{y}_t, m', m)$ as (17).

where $x_t^{(l)}$ denotes the sub-block through the two-user trellis associated with the transition $\sigma_{t-1} = m' \rightarrow \sigma_t = m$ and inputs $u_t = i$ and $d_t = j$. From (11) and (17) we can rewrite (3), (4) and (5) in a compact form as (18).

We can now outline the operation of the BCJR decoder:

- 1) Initialise $\alpha_0(m)$ and $\beta_N(m)$ according to (13) and (15) or (16);
- 2) As soon as \mathbf{y}_t is received the decoder computes $\gamma_{i,j}(\mathbf{y}_t, m', m)$ using (17) and $\alpha_t(m)$ using (12). The obtained values of $\alpha_t(m)$ are stored for all t and m ;
- 3) After the complete sequence \mathbf{y}_1^N is received, the decoder computes recursively $\beta_t(m)$ using (14). When the $\beta_t(m)$ have been computed they are multiplied by the appropriate $\alpha_t(m)$ and $\gamma_{i,j}(\mathbf{y}_t, m', m)$ to obtain (18) and make decisions on (\hat{u}_t, \hat{d}_t) .

We refer the reader to [6] where some performance curves are presented for the BCJR algorithm applied to a turbo coding system for the noisy 2-BAC.

IV. TURBO DECODING

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

Suppose the encoder for C_1 employs the turbo code construction of Berrou et al. [2]. 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_t , 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_t , 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 $1/n$.

Example 2: Let C_1^- and C_1^+ denote two binary recursive systematic rate $1/2$ convolutional codes with 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 with 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 3 and Figure 4, respectively.

The decoder employed, illustrated in Figure 5, uses iterative turbo decoding [7] to detect the most likely pairs (u_t, d_t) of binary information symbols. The iterative algorithm employed uses the BCJR technique [1], adapted for use in the 2-BAC [8], making use of the 2-user trellis. The log-likelihood ratios $\Lambda_1(u_t, d_t)$, $\Lambda_2(u_t, d_t)$ and $\Lambda_3(u_t, d_t)$ associated with the pair (u_t, d_t) of information symbols from users 1 and 2, respectively, are computed using expressions (1), (2) and (3).

$$\gamma_{i,j}(y_t, m', m) = \begin{cases} P_t(i, j) \exp \left[-\sum_{l=0}^{n-1} (y_t^{(l)} - x_{t,i,j}^{(l)})^2 / 2\sigma^2 \right], & \text{for } (m, m') \in B_t^{i,j}. \\ 0, & \text{otherwise.} \end{cases} \quad (17)$$

$$\Lambda_{2i+j,t} = \log \frac{\sum_m \sum_{m'} P_t(i, j) \exp \left[-\sum_{j=0}^{n-1} (y_t^{(l)} - x_{t,i,j}^{(l)})^2 / 2\sigma^2 \right] \alpha_{t-1}(m') \beta_t(m)}{\sum_m \sum_{m'} P_t(0, 0) \exp \left[-\sum_{j=0}^{n-1} (y_t^{(l)} - x_{k,0,0}^{(l)})^2 / 2\sigma^2 \right] \alpha_{t-1}(m') \beta_t(m)}. \quad (18)$$

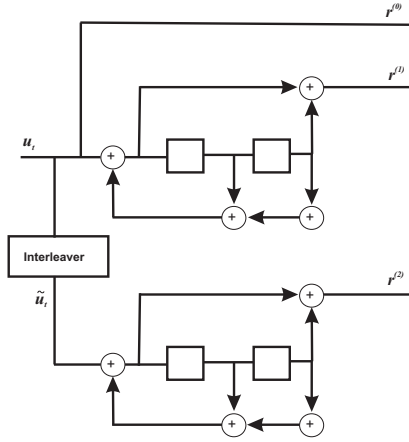


Fig. 3. Encoder for turbo code C_1 , employing two identical polynomial generator matrices, namely $\left[1 \quad \frac{1+D^2}{1+D+D^2} \right]$.

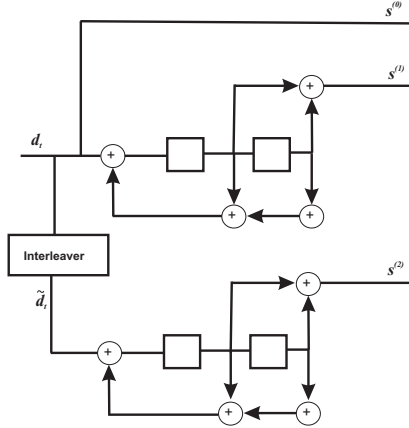


Fig. 4. Encoder for turbo code C_2 , employing two identical polynomial generator matrices, namely $\left[1 \quad \frac{D+D^2}{1+D+D^2} \right]$.

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 $\mathbf{y}^{(0)} = \{y_1^{(0)}, y_2^{(0)}, \dots, y_N^{(0)}\}$ and $\mathbf{y}^{(1)} = \{y_1^{(1)}, y_2^{(1)}, \dots, y_N^{(1)}\}$, where $y_t^{(j)}$ was defined earlier. DEC1 produces the soft outputs $\Lambda_1^1(u_t, d_t)$, $\Lambda_2^1(u_t, d_t)$ and $\Lambda_3^1(u_t, d_t)$, 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 $\Lambda_1^1(u_t, d_t)$, $\Lambda_2^1(u_t, d_t)$, $\Lambda_3^1(u_t, d_t)$ is used to indicate the soft outputs $\Lambda_1(u_t, d_t)$, $\Lambda_2(u_t, d_t)$ and $\Lambda_3(u_t, d_t)$ associated with DEC1, respectively. The values $\hat{\Lambda}_{1e}^1(u_t, d_t)$, $\hat{\Lambda}_{2e}^1(u_t, d_t)$ and $\hat{\Lambda}_{3e}^1(u_t, d_t)$ represent the extrinsic information produced

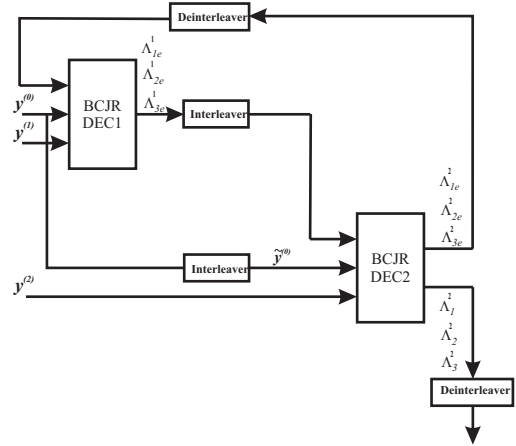


Fig. 5. Iterative turbo decoder, containing two component decoders connected in series.

by decoder DEC1.

The input to decoder DEC2 receives the sequences $\tilde{\mathbf{y}}^{(0)}$ and $\mathbf{y}^{(2)} = \{y_1^{(2)}, y_2^{(2)}, \dots, y_N^{(2)}\}$. The sequence $\tilde{\mathbf{y}}^{(0)}$ corresponds to the sequence $\mathbf{y}^{(0)}$ interleaved. Decoder DEC2 also produces soft outputs $\Lambda_1^2(u_t, d_t)$, $\Lambda_2^2(u_t, d_t)$ and $\Lambda_3^2(u_t, d_t)$, denoted as $\hat{\Lambda}_1^2(u_t, d_t)$, $\hat{\Lambda}_2^2(u_t, d_t)$ and $\hat{\Lambda}_3^2(u_t, d_t)$, 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_t, d_t) input to decoder DEC1. Decoder DEC2 computes the log likelihood ratios $\Lambda_1^2(u_t, d_t)$, $\Lambda_2^2(u_t, d_t)$ and $\Lambda_3^2(u_t, d_t)$. The values $\hat{\Lambda}_{1e}^2(u_t, d_t)$, $\hat{\Lambda}_{2e}^2(u_t, d_t)$ and $\hat{\Lambda}_{3e}^2(u_t, d_t)$ represent the extrinsic information produced by decoder DEC2. Such values depend on the redundant information supplied by the encoders C_1^1 and C_2^1 . The extrinsic information produced by decoder DEC2 is used as an estimate of the *a priori* probabilities to decoder DEC1. The values $\hat{\Lambda}_{1e}^2(u_t, d_t)$, $\hat{\Lambda}_{2e}^2(u_t, d_t)$ and $\hat{\Lambda}_{3e}^2(u_t, d_t)$ correspond, respectively, to the values of $\Lambda_{1e}^2(u_t, d_t)$, $\Lambda_{2e}^2(u_t, d_t)$ and $\Lambda_{3e}^2(u_t, d_t)$ when deinterleaved.

As mentioned in Section I, the use of identical error-correcting codes in systematic form for both users in the 2-BAC suffers with a form of ambiguity expressed as $\mathbf{P}\{u_t = 0, d_t = 1 | \mathbf{y}\} = \mathbf{P}\{u_t = 1, d_t = 0 | \mathbf{y}\}$. In terms of the decoder operation this ambiguity condition is expressed, equivalently, in terms of log likelihood ratios as $\Lambda_1(u_t, d_t) = \Lambda_3(u_t, d_t)$. The occurrence of this log likelihood ratio equality, at any time instant t , forbids any trellis decoder of separating the symbols sent by each user in the 2-BAC at time instant t , except for the trivial cases, i.e., where $u_t = d_t$.

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,

$$\Lambda_1(u_t, d_t) \neq \Lambda_3(u_t, d_t)$$

or, equivalently,

$$P\{u_t = 0, d_t = 1|y\} \neq P\{u_t = 1, d_t = 0|y\},$$

when the turbo codes for user 1 and user 2 are distinct.

V. SIMULATION RESULTS

The standard Berrou and Glavieux [2] 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 y through DEC 1 only once. As expected, when iterative turbo 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.

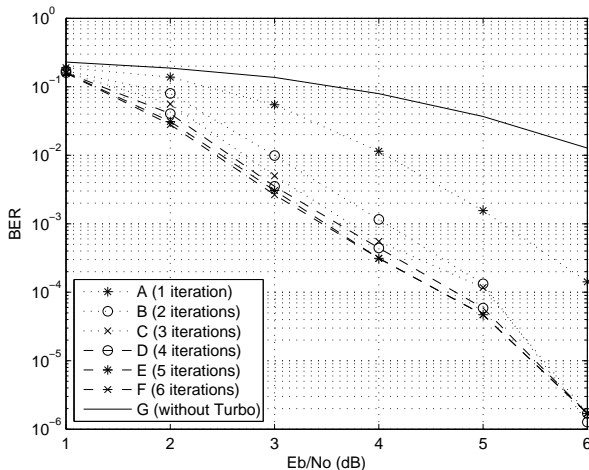


Fig. 6. Bit error rate for user 1, for a rate 1/3 turbo convolutional code with polynomial generator matrix $[1 \ (1 + D^2)/(1 + D + D^2)]$.

VI. CONCLUSIONS

The advantage of this construction, employing distinct turbo codes, one for each user, comes from the possibility of directly separating the binary data for each of the two users at the receiver by using the BCJR [1] iterative decoding algorithm applied to the received sequence y and the corresponding 2-BAC trellis. In [9] the problem of designing good low density parity-check codes (LDPC) is addressed, for the Gaussian multiple access channel. However long codes are not attractive for some practical cases due to loss of synchronization. In

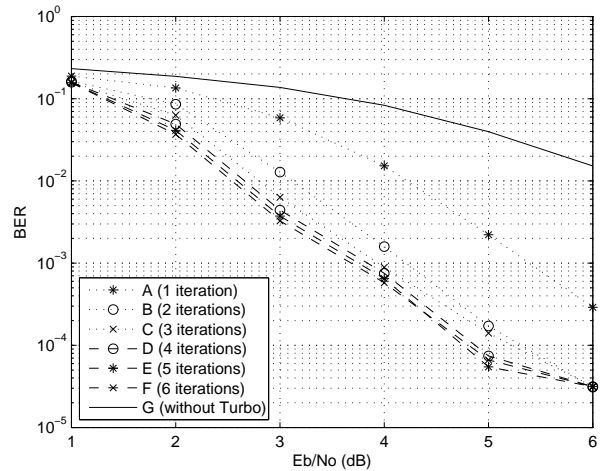


Fig. 7. Bit error rate for user 2, for a rate 1/3 turbo convolutional code with polynomial generator matrix $[1 \ (D + D^2)/(1 + D + D^2)]$.

these cases, short codes are desirable. Recently [10], 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.

The complexity analysis of the BCJR algorithm for the 2-BAC produces essentially similar results to those given in [1], i.e., the algorithm requires large storage and considerable computation, being practical only for short constraint lengths and short subblock lengths. All the values of $\alpha_t(m)$ must be stored, which requires roughly $\tau 2^{2k\nu}$ storage locations. The storage size grows exponentially with constraint length and linearly with block length. The number of computations in determining the $\alpha_t(m)$ (or $\beta_t(m)$) for each t are $M 2^{2k}$ multiplications and M additions of 2^{2k} numbers each. The computation of the $\gamma_t(m', m)$ is quite simple and [1] suggests using a table lookup. Computing $\lambda_t(m)$ requires M multiplications for each t and computing the *a posteriori* probability of the input digits requires $kM/2$ additions.

ACKNOWLEDGEMENT

Valdemar C. da Rocha Jr. acknowledges partial support from the Brazilian National Council for Scientific and Technological Development - CNPq under grant number 306612/2007-0.

REFERENCES

- [1] L.R. Bahl, J. Cocke, F. Jelinek and J. Raviv, “Optimal decoding of linear codes for minimizing symbol error rate”, *IEEE Trans. on Inform. Theory*, vol. IT-20, pp.284-287, March 1974.
- [2] C. Berrou, A. Glavieux and P. Thitimajshima, “Near Shannon limit, error-correcting coding and decoding: turbo codes”, *IEEE Int. Conf. on Comm. (ICC'93)*, vol. 2/3, pp.1064-1071, May 1993.
- [3] T. Kasami and S. Lin, “Coding for a multiple-access channel”, *IEEE Trans. on Inform. Theory*, Vol. IT-22, Number 2, March 1976, pp.129-137.
- [4] 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*, Ambleside, UK, pp. 142-147, 2005.
- [5] 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, January 1979.

- [6] M. L. M. G. Alcoforado, V. C. da Rocha Jr. and G. Markarian, "Further results on turbo convolutional codes for the binary adder channel". *Ninth Intern. Symp. on Comm. Th. and Applications (ISCTA'07)*, 2007, Ambleside, UK, pp. 1-6, 2007.
- [7] J. Hagenauer, "Iterative decoding of binary block and convolutional codes", *IEEE Trans. on Inform. Theory*, vol. 42, no. 2, pp. 429-445, 1996.
- [8] M. L. M. G. Alcoforado, V. C. da Rocha Jr. and G. M. J. de C. Lima, "A Turbo Coding Scheme for the 2-User Gaussian Adder Channel". *International Telecommunications Symposium 2010 (ITS 2010)*, Manaus, Brasil, 2010.
- [9] A. Roumy and D. Declercq, "Characterization and optimization of LDPC codes for the 2-User Gaussian multiple access channel", *EURASIP Journal on Wireless Communications and Networking*, 2007.
- [10] 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.



Maria de Lourdes M. G. Alcoforado was born in Recife, PE, Brazil. She received the Electronic Engineering degree (1995), the M.Sc. degree (2000) and the D.Sc. degree (2005) from the Federal University of Pernambuco. Since 2001 she has been dedicated to researching and teaching in undergraduate and postgraduate courses at University of Pernambuco. Research interests include turbo codes, multiple access channel and cryptography.



Valdemar C. da Rocha Jr. (M'77, SM'04) was born in Jaboatão, Pernambuco, Brazil, on August 27, 1947. He received in 1970 the B.Sc. degree in Electrical/Electronics Engineering from the Escola Politécnica, Recife, Brazil, and in 1976 he received the Ph.D. degree in Electronics from the University of Kent at Canterbury, U.K. He joined the faculty of the Federal University of Pernambuco, Recife, Brazil, in 1976 as an Associate Professor and founded its Electrical Engineering Postgraduate Programme. He served as Department Chair (1992-1996), and in 1993 he became Professor of Telecommunications.

He was editor for Coding Theory and Techniques, Journal of Communication and Information Systems, co-sponsored by the Brazilian Telecommunications Society and the IEEE Communications Society, and has been a reviewer for a number of scientific journals including IET Electronics Letters, IET Communications and IEEE Transactions on Information Theory. He has also been involved in the organization of conferences in Brazil and abroad.

He is a founder (2002) and past President (2002-2004) of the IEEE Information Theory Society Chapter, Brazil Council. He is founder (2003) and Vice-President for two consecutive terms (2003-2011) of the Institute for Advanced Studies in Communications. He is a founding member (1983) of the Brazilian Telecommunications Society, served as Vice-President for two terms (2000-2004) and as President also for two terms (2004-2008). He joined the IEEE Communications Society in 1977 and the IEEE Information Theory Society in 1981. He is a Member (1982) of the Brazilian Society of Applied and Computational Mathematics, and a Fellow (1992) of the Institute of Mathematics and its Applications, UK.

During 1990-1992, he was a Guest Professor at the Swiss Federal Institute of Technology-Zurich, Institute for Signal and Information Processing. In 2005-2006 he was a Visiting Professor at the Institute of Integrated Information Systems, University of Leeds, UK, and in 2007 he was a Visiting Professor at the Department of Communication Systems, Lancaster University, UK.

Prof. da Rocha research interests are in applied digital information theory, including error-correcting codes and cryptography. He has published over 100 engineering and scientific papers, including journal and conference papers, and the book *Communication Systems*, by Springer, 2005, in co-authorship.