Non-linear Turbo Codes for Interleaver-Division Multiple Access on the OR Channel.

Miguel Griot, Andres I. Vila Casado and Richard Wesel
Dept. of Electrical Engineering, University of California
Los Angeles, CA 90095
{mgriot,avila,wesel}@ee.ucla.edu

***Abstract*— This paper presents an Interleaver-Division Multiple Access (IDMA) based architecture with single-user decoding using parallel concatenated non-linear trellis codes (PC-NLTCs). These PC-NLTCs are designed specifically for the Z-Channel that arises in a multiple-user OR channel when each user treats the other users as noise. Over the OR Multiple Access Channel (OR-MAC) single-user decoding permits operation at about 70% of the full multiple access channel sum capacity. In order to reach the sum capacity of the OR-MAC, these codes employ a ones density of much less than 50%. A union bound technique that predicts the performance of these codes under Maximum-Likelihood (ML) decoding is presented. The uniform interleaver analysis presented in this paper can be applied to any asymmetric channel, as long as an additive distance can be defined. Results for different numbers of users and a sum-rate of 60% are presented.**

I. Introduction

There have been many contributions to the problem of providing multiple access to a same channel. However, the most common forms of multiple acces, such as time-division (TDMA), frequency-division (FDMA), code-division (CDMA) or rate-splitting [1], require considerable coordination. One recent successful approach for uncoordinated multiple-access is Interleaver-Division Multiple-Access (IDMA) [2][3][4], which uses interleaving to distinguish among signals from different users.

Consider the OR channel, or its isomorphic channel, the Binary Multiplier Channel [5], as a target application for IDMA. Completely uncoordinated transmissions using IDMA and simple decoding that treats all signals except the desired signal as noise can theoretically achieve about 70% of the sum capacity over the OR channel. By sacrificing 30% of the sum rate, this IDMA approach provides a significant reduction in complexity over coordinated or joint approaches, making it a practically attractive technique.

This paper presents an uncoordinated multiple access system employing IDMA on the OR-MAC with single-user decoding (SUD), where other users are treated as noise. Since a ones density of much less than 50% is required in this application to achieve the SUD sum capacity, non-linear codes are required. Parallel concatenated non-linear trellis codes (PC-NLTCs) are proposed. These codes provide a wide range of ones densities and potentially approach the approximately 70% SUD sum capacity.

This work was supported by the Defence Advanced Research Project Agency under SPAWAR Systems Center San Diego Grant N66001-02-1-8938.

This paper is organized as follows. Section II reviews uncoordinated multiple access in the OR channel. Section III presents the design of the PC-NLTCs for this application. Section IV presents a BER bound for PC-NLTCs under ML decoding, operating on the Z-Channel. Section V presents an analysis on the number of users that can be supported by this approach, for a fixed complexity, without a major degradation in performance. Section VI presents performance results and Section VII concludes the paper.

II. Uncoordinated Multiple Access in the OR Channel

In the OR-MAC, if all users transmit a zero, then the channel output is a zero. However, if even one user transmits a one, then the channel output is a one. This channel is isomorphic, interchanging ones and zeros at both the transmitters and the receiver's side, to the Binary Multiplier Channel. The information-theoretic capacity region of this channel is the section of the positive orthant bounded by the unit n_u-simplex, where n_u is the number of users. In other words, it is the region where all the rates are non-negative and the sum of all rates is less than or equal to 1.

This capacity may be achieved with time-division multiple access, joint decoding of all the transmitted sequences, or sequential decoding if the transmitted ones densities and rates are carefully controlled [1]. All of these solutions require either coordination of all users or a very complex decoder, especially for a large number of users.

We propose a less complex alternative to joint decoding and successive decoding, where each decoder treats all signals except the desired signal as noise. This transforms the OR channel into the Z-Channel shown in Fig. 1. Assuming that all users have the same transmitted ones density p_1, the zero-to-one transition probability, denoted as α, is the probability that any of the other users transmits a 1:

$$\alpha = 1 - (1 - p_1)^{n-1}, \tag{1}$$

which is a function of that ones density employed by the desired user, and the number of users.

The maximum theoretical sum-rate with single-user decoding decreases as the number of users increases, but it converges monotonically to $\ln 2 \simeq 0.6931$. This is a relatively small loss in rate for the substantial reduction in complexity. In order to be able to achieve this maximum theoretical sum-rate, the

1-4244-0357-X/06/$20.00 ©2006 IEEE
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2006 proceedings.

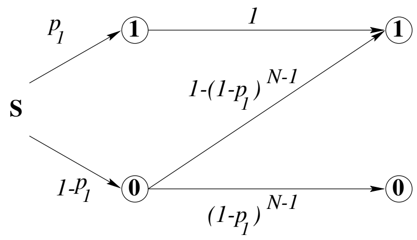

Fig. 1. Z-channel resulting from the OR-MAC channel when other users are treated as noise, and all users employ ones density p_1.

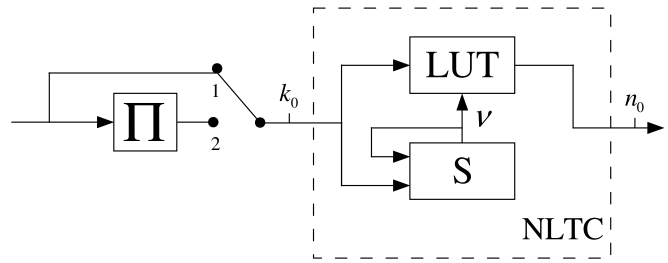

Fig. 2. PC-NLTC structure.

optimal ones density of each individual user decreases as the number of users increase. For example, the optimal density of ones is $p_1 \simeq 0.2864$ for 2 equal-rate users, $p_1 \simeq 0.1080$ for 6 equal-rate users, and $p_1 \simeq 0.0558$ for 12 equal-rate users.

On the other hand, when maintaining equally likely ones and zeros ($p_1 = 0.5$) the maximum theoretical sum-rate rapidly decreases to zero with the number of users.

One successful approach for uncoordinated multiple-access is IDMA. With IDMA, every user has the same channel code, but each user's code bits are permuted using an interleaver drawn at random, unique with probability close to 1. The receiver is assumed to know the interleaver of the desired user. Since the interleavers are independently and randomly picked by each user, the resulting distributions of ones and zeros at each time are IID. Hence, with IDMA in the OR-MAC, a receiver should see the desired signal corrupted by a memoryless Z-channel. We compared the performance of non-linear trellis codes under two channels: 1) a 6-user OR-MAC channel using IDMA and 2) the equivalent Z-channel that the receiver would see if the errors were not generated by codewords but by random errors. The performance was the same, which corroborates the theory. Thus, in the context of IDMA, the remaining challenge is the design of a good code with the desired ones density.

III. PARALLEL CONCATENATED NON-LINEAR TRELLIS CODES WITH CONTROLLED ONES DENSITY

There have been several papers that have addressed the problem of designing codes with $p_1 = 0.5$ for the Z-channel. See [6] for a unified account on such codes and [7] for the most recent advances in this field known to the authors. Only recently there has been work on LDPC codes with an arbitrary density of ones, see [8] and [9]. A design of non-linear trellis codes with a low ones density for the Z-Channel appears in [10]. This manuscript is the first to our knowledge to address the design of parallel concatenated non-linear trellis codes (PC-NLTC) with a controlled ones density for the Z-Channel.

The structure of the PC-NLTC encoder is shown in Fig. 2. It is in essence the well-known turbo-code structure first proposed in [11], for systematic linear encoders. However, non-linear and non-systematic codes are necessary to provide with the low ones densities required by the application considered in this work. The encoder consists of two constituent (n_0, k_0) non-linear trellis encoders (block NLTC) linked by an interleaver (block Π), where k_0 and n_0 are the number of input and output bits per trellis section respectively. The rate of the code is then $k_0/(2n_0)$. The NLTC is composed by a 2^ν-state trellis structure (block S), and a look-up table (block LUT). The block S stores the current trellis state, while the look-up table stores an output for each branch of the trellis. These trellis codes are non-systematic, and provide, by choosing the weights of the outputs of each branch, a controlled ones density.

This code must satisfy the optimal ones density $p_1(n_u)$ given by the number of users n_u. When treating other users as noise, $p_1(n_u) \to 1 - (1/2)^{1/n_u}$ when $n_u \to \infty$. Actually, even for a relatively small number of users one can consider

$$p_1(n_u) \simeq 1 - (1/2)^{1/n_u}. \tag{2}$$

Another design parameter is the desired target sum-rate, which will be denoted as R^+. Theoretically, error-free transmission can be achieved if $R^+ \leq \ln 2$. We set the target sum-rate to $R^+ = 0.6$ in this work, since an excess mutual information of 0.1 bits is typical of AWGN turbo codes with similar blocklengths operating at similar spectral efficiencies.

Given the design parameters $p_1(n_u)$ and R^+, the following parameters for the constituent codes need to be chosen:

- The number of trellis states. Typically 2^ν, where $\nu = 3, 4$.
- The number of bits per output branch n_0. This value has to be chosen so that the sum-rate is as close as possible, if not equal, to the target sum-rate:

$$n_u \cdot \big(k_0/(2 \cdot n_0)\big) \simeq R^+. \tag{3}$$

- The Hamming weight of the output of each trellis branch. The average Hamming weight of the output $\hat{w}_b$ must satisfy:

$$\hat{w}_b \simeq p_1(n_u) \cdot n_0. \tag{4}$$

  Denote W_b the total number of ones in all the $2^{k_0+\nu}$ branches. Then

$$W_b \simeq p_1(n_u) \cdot n_0 \cdot 2^{k_0+\nu}. \tag{5}$$

For example, using a parallel concatenation of two 8-state NLTCs ($\nu = 3$), for $n_u = 6$, the average number of ones per output trellis branch is

$$\hat{w}_b \simeq p_1(n_u) \cdot n_0 = \frac{p_1(n_u) \cdot n_u}{2 \cdot R^+} \cdot k_0 \simeq 0.54 \cdot k_0. \tag{6}$$

If single-input encoders are used ($k_0 = 1$), at least 46% of the branches should have all-zero outputs. This is the case for any number of users. Hence, single-input encoders would have a very low minimum distance in this application, resulting in a poor performance. Therefore, constituent trellis codes with $k_0 \geq 2$ are required. Multiple-input convolutional codes for turbo coding have been studied in [12][13][14] among other papers.

Using a trellis structure with $k_0 = 2$, for $n_u = 6$ users and a target sum-rate of $R^+ = 0.6$, then $n_0 = 10$, and $\hat{w}_b \simeq 1.08$.

The design of the PC-NLTC consists of choosing the trellis branch-structure and the output values of the branches that satisfy the required $\hat{w}_b$.

IV. Upper Bound to Bit Error Probability of Parallel Concatenated NLTCs

A method to evaluate to the bit error probability of a parallel concatenated coding scheme averaged over all interleavers of a certain length, has been proposed in [15]. This upper bound is known as the uniform interleaver bound, and assumes the use of a Maximum-Likelihood (ML) decoder. However, this bound cannot be applied to PC-NLTCs because it assumes a parallel concatenation of **linear codes**. Hence, an upper bound to the BER is found assuming the all-zero word is transmitted. As mentioned earlier, the OR-MAC requires a low ones density, for which non-linear codes are necessary. In this case, all data-words need to be considered when finding the upper-bound.

Also, the analysis in [15] assumes a parallel concatenation of **systematic codes**. In our case, because the transmitted ones density must be controlled and needs to be low, non-systematic codes are used.

Thus, an extension of the bounding technique proposed in [15] for a parallel concatenation of non-linear codes on the Z-Channel, is required. The technique presented in this work is valid for any asymmetric channel, provided an additive distance can be defined for that channel.

A. Some definitions

Consider any two words of length n, $X^n = \{x_1, \cdots, x_n\}$ and $\tilde{X}^n = \{\tilde{x}_1, \cdots, \tilde{x}_n\}$. Define the *Directional Hamming Distance* for the Z-Channel $d_D(X^n, \tilde{X}^n)$ as the number of positions where $x_i = 0$ and $\tilde{x}_i = 1$, with $i = 1, \cdots, n$. Note that $d_D(X^n, \tilde{X}^n)$ is not necessarily equal to $d_D(\tilde{X}^n, X^n)$.

Let $Y^n = \{y_1, \cdots, y_n\}$ the received word. It is clear that given Y^n, any possible transmitted codeword X^n must satisfy $d_D(Y^n, X^n) = 0$, since there cannot be any one-to-zero transitions on the Z-Channel. The most likely transmitted codeword $\hat{X}^n$, is the codeword X^n satisfying $d_D(Y^n, X^n) = 0$, that minimizes the number of zero-to-one transitions. Hence, the maximum likelihood decoder for the Z-Channel chooses the codeword $\hat{X}^n$ as:

$$\hat{X}^n = \text{argmin}_{X^n \in \mathcal{N}} \Big[d_D(X^n, Y^n) \Big], \tag{7}$$

where $\mathcal{N}$ is the set of codewords that satisfy $d_D(Y^n, X^n) = 0$.

Let α be the probability of a zero-to-one transition in the Z-Channel. Using Eq. (7), it can be derived that the probability of transmitting X^n and decoding $\tilde{X}^n$ under ML decoding is:

$$P_e(X^n \to \tilde{X}^n) = \tag{8}$$

$$\begin{cases} \frac{1}{2} \cdot \alpha^{d_D(X^n, \tilde{X}^n)} & , W_H(X^n) = W_H(\tilde{X}^n) \\ \alpha^{d_D(X^n, \tilde{X}^n)} & , W_H(X^n) < W_H(\tilde{X}^n) \\ 0 & , W_H(X^n) > W_H(\tilde{X}^n). \end{cases}$$

where $W_H(\cdot)$ denotes the Hamming weight.

Define the *Weight-Distance Enumerating Function* (WDEF) of a given (n, k) code C as

$$A^C(W, H, I, D) = \sum_{w,h,i,d} A^C_{w,h,i,d} D^d I^i H^h W^w, \tag{9}$$

where $A^C_{w,h,i,d}$ is the number of data-word pairs $(U^k, \hat{U}^k)$ that satisfy the following conditions:
$W_H(U^k) = w$, $W_H(\hat{U}^k) - W_H(U^k) = h$, their Hamming distance $d_H(U^k, \hat{U}^k) = i$, and the directional distance between the corresponding codewords $d_D(X^n, \tilde{X}^n) = d$. W, H, I and D are placeholders.

Also define the *Conditional Directional Distance Enumerating Function* (CDDEF) as:

$$A^C_{w,h,i}(D) = \sum_d A^C_{w,h,i,d} D^d. \tag{10}$$

Inserting Eq. (10) in Eq. (9), the expression for the WDEF can be rewritten as:

$$A^C(W, H, I, D) = \sum_{w,h,i} A^C_{w,h,i}(D) I^i H^h W^w. \tag{11}$$

Consider the sequence pair $(X^n, \tilde{X}^n)$. It can be derived from Eq. (8) that:

$$\begin{aligned} & P_e(X^n \to \tilde{X}^n) + P_e(\tilde{X}^n \to X^n) \\ & = \alpha^{\max(d_D(X^n, \tilde{X}^n), d_D(\tilde{X}^n, X^n))} \\ & \leq \tfrac{1}{2} [\alpha^{d_D(X^n, \tilde{X}^n)} + \alpha^{d_D(\tilde{X}^n, X^n)}] \end{aligned} \tag{12}$$

Therefore, if $P_e(X^n \to \tilde{X}^n)$ is replaced (not always upper-bounded) by $\frac{1}{2} \alpha^{d_D(X^n, \tilde{X}^n)}$ for all the codewords X^n and $\tilde{X}^n$, and assuming equally likely input words, an upper bound to the BER can be written as:

$$\text{BER} \leq \frac{1}{2k} \cdot (1/2)^k \cdot \frac{\partial A^C(W, H, I, D)}{\partial I} \Bigg|_{D=\alpha, H=I=W=1}. \tag{13}$$

Finally, a *Uniform Interleaver* of length k is defined in [15] as a probabilistic device which maps a given input word of weight w into all distinct $\binom{k}{w}$ permutations of it with equal probability $1/\binom{k}{w}$.

B. Parallel concatenation of block codes

Denote C_P as the (n_1+n_2, k) block code resulting from the parallel concatenation of two codes, an (n_1, k) block code C_1 and an (n_2, k) block code C_2. We will assume an interleaver of length k, equal to the input word length, in order to simplify the analysis (An extension easily can be made for the case when l consecutive codewords of the constituent codes are used as a basic codeword for interleaving, as explained in [15]).

As explained in Sec. IV all 2^k codewords need to be considered. Consider an input word pair $(U^k, \hat{U}^k)$ and the corresponding interleaved input word pair $(\Pi(U^k), \Pi(\hat{U}^k))$. The interleaver preserves their Hamming weight, and the Hamming distance between each other. Namely,

$$W_H(U^k) = W_H(\Pi(U^k)) = w, \tag{14}$$

$$W_H(\hat{U}^k) - W_H(U^k) = W_H(\Pi(\hat{U}^k)) - W_H(\Pi(U^k)) = h, \tag{15}$$

$$d_H(U^k, \hat{U}^k) = d_H(\Pi(U^k), \Pi(\hat{U}^k)) = i. \tag{16}$$

Now, the number of pairs with those same w, h and i are:

$$\binom{k}{w} \cdot \binom{k-w}{(i+h)/2} \cdot \binom{w}{(i-h)/2}. \tag{17}$$

Using the uniform interleaver, the resulting interleaved pair $(\Pi(U^k), \Pi(\hat{U}^k))$ can be any of those with equal probability. Furthermore, the directional distance is additive, so the directional distance of the concatenated codeword is the sum of the directional distances between the corresponding constituent codewords.

Hence, the WDEF of C_P can be expressed as:

$$A^{C_P}_{w,h,i}(D) = \frac{A^{C_1}_{w,h,i}(D) \cdot A^{C_2}_{w,h,i}(D)}{\binom{k}{w} \cdot \binom{k-w}{(i+h)/2} \cdot \binom{w}{(i-h)/2}}. \tag{18}$$

C. Parallel concatenated non-linear trellis codes

Biglieri et al. presented a union bound in [16][17] for general trellis codes, using a $2^{2\nu}$-state trellis diagram. This can be applied to non-linear trellis codes over the Z-channel with modifications on the pairwise error probability measure, as shown in [10]. This same concept can be used to find $A^{C_P}(W, H, I, D)$ for the case of parallel concatenated non-linear trellis codes.

As in [16], the product state diagram consists of state pairs, (s_e, s_r), where s_e is the encoder state and s_r the receiver state. Following Biglieri's notation, the product states can be divided into two sets, the good states denoted by S_G and the bad states denoted by S_B defined as

$$S_G = \{(s_e, s_r) \mid s_e = s_r\}, \; S_B = \{(s_e, s_r) \mid s_e \neq s_r\}. \tag{19}$$

By suitably renumbering the product states, we get the transition matrix

$$S(W,H,I,D) = \left[\begin{array}{c|c} S_{GG}(W,H,I,D) & S_{GB}(W,H,I,D) \\ \hline S_{BG}(W,H,I,D) & S_{BB}(W,H,I,D) \end{array}\right], \tag{20}$$

where the $N \times N$ matrix $S_{GG}(W, H, I, D)$ accounts for the transitions between good product states, the $N \times (N^2 - N)$ matrix $S_{GB}(W, H, I, D)$ accounts for the transition from good product states to bad product states, and so forth. N is the number of encoder states 2^ν. For each transition in the product state diagram from product state S_1 to S_2, the branch label is:

$$W^{W_H(u_e)} H^{W_H(u_r) - W_H(u_e)} I^{d_H(u_e, u_r)} D^{d_D(x_e, x_r)}, \tag{21}$$

where u_e and x_e denote the input and output word for the encoder states respectively, and u_r and x_r denote the input and output word for the receiver.

Note that for an (n_0, k_0) non-linear trellis code, k/k_0 trellis sections are traversed with k input bits. Define the *Weight-Distance Enumerating Matrix* as

$$R(W,H,I,D) = \Big[S(W,H,I,D)\Big]^{k/k_0}, \tag{22}$$

as the matrix representing all possible WDEFs when starting from any initial product-state $S_i = (s_{e,i}, s_{r,i})$ (row) and ending in any final product-state $S_f = (s_{e,f}, s_{r,f})$ (column) after (k/k_0) trellis sections.

For the case where zero-termination is used, every encoding process starts and ends in the all-zero state. Thus:

$$A^{C_x}(W,H,I,D) = \{R(W,H,I,D)\}_{(S_i=(0,0)) \times (S_f=(0,0))}, \tag{23}$$

where $C_x = C_1, C_2$ denotes the constituent code, and $\{M\}_{(S_i=(s_{e,i},s_{r,i})) \times (S_f=(s_{e,f},s_{r,f}))}$ denotes the value of matrix M in row $S_i = (s_{e,i}, s_{r,i})$ and column $S_f = (s_{e,f}, s_{r,f})$. For a generic case, every possible combination of initial and final product-state for each of the constituent encoders has to be considered.

However, even for the zero-termination case, the computation of $A^{C_x}(W, H, I, D)$ becomes very complex in terms of number of operations.

In order to reduce complexity, two approximations can be made: (1) Use the same idea presented in [15]: every path in the trellis representation starts and ends in the same state. Any possible incorrect word departs from a good state to a bad state at some trellis section a certain number of times m, and returns to a good state the same number of times m. (2) In the encoding process, at any trellis section, the encoder state can be any of the possible $N = 2^v$ states with equal probability.

Define the approximated single-error event function as:

$$E(\hat{W}, H, I, D, L) = p_s\{S_{GB}(I - S_{BB})^{-1} S_{BG}\}\mathbf{1}, \tag{24}$$

where $p_s = [\frac{1}{N} \frac{1}{N} \cdots \frac{1}{N}]$ is the probability distribution of the encoder states and $\mathbf{1} = [11 \cdots 1]^T$. Note that a new placeholder L has been added. This placeholder counts the length, in trellis sections, of an error event.

Then, $E(\hat{W}, H, I, D, L)$ can be written as:

$$E(\hat{W}, H, I, D, L) = \sum_{\hat{w},h,i,d,l} e_{\hat{w},h,i,d,l} L^l D^d I^i H^h \hat{W}^{\hat{w}}, \tag{25}$$

where $\hat{w}$ is the accumulated Hamming weight of the correct word in the single-error event. Note that the Hamming weight of the correct word is not given by $\hat{w}$.

Now define:

$$E_j(\hat{W}, H, I, D, L) = \Big[E(\hat{W}, H, I, D, L)\Big]^j \qquad (26)$$
$$= \sum_{\hat{w},h,i,d,l} e_{\hat{w},h,i,d,l,j} L^l D^d I^i H^h \hat{W}^{\hat{w}},$$

which counts every concatenation of j single-error events, without leaving any trellis section between them, using approximation (2).

Every error event can be represented as a concatenation of single-error events. Using approximation (2), a concatenation of j single-error events, with a total length l can be positioned in $K[l, j] = \binom{k/k_0 - l + j}{j}$ ways in the trellis. The error event shows the difference in Hamming weight h between the word pair, and their Hamming distance between each other i. However, the accumulated Hamming weight of the correct word during this error event $\hat{w}$, is not necessarily the Hamming weight of the input word. In fact, given a certain position of the error event in the trellis, there are $\binom{k - k_0 \cdot l}{w - \hat{w}}$ pairs that traverse that same error event, where the correct word has $W_H = w$ and the incorrect word has $W_H = w + h$.

Hence, for each constituent code, $A^{C_i}_{w,h,i,d}$ can be expressed as:

$$A^{C_i}_{w,h,i,d} = \sum_{j,l,\hat{w} \le w} \binom{k/k_0 - l + j}{j} \cdot \binom{k - k_0 \cdot l}{w - \hat{w}} e^{C_i}_{\hat{w},h,i,d,l,j}. \qquad (27)$$

V. Limitation on the number of users

As mentioned in Sec. II, a sum-rate of less than or equal to $\ln 2 \simeq 70\%$ can be theoretically achieved for any number of users in the OR-MAC, when each user treats the others as noise. However, for a fixed number of input-bits per trellis section k_0, number of states ν and target sum-rate R^+, and maximum tolerable BER, there may be a limitation on number of users n_u.

Given a certain number of users n_u, and using (2-5), the total number of ones in all the $2^{k_0+\nu}$ branches can be rewritten as:

$$W_b(n_u) = \Big(\frac{k_0 \cdot 2^{k_0+\nu}}{2 \cdot R^+}\Big) \cdot \Big(n_u \cdot (1 - (1/2)^{1/n_u})\Big). \qquad (28)$$

Now,

$$\lim_{n_u \to \infty} n_u \cdot (1 - (1/2)^{1/n_u}) = \ln 2, \qquad (29)$$

and is upper-bounded by that number. It actually converges very fast to that value. Thus,

$$W_b(n_u) \to \ln 2 \cdot \Big(\frac{k_0 \cdot 2^{k_0+\nu}}{R^+}\Big), \qquad (30)$$

for a large enough number of users. For example, in the results shown in Sec. VI, $R^+ = 0.6$ and $k_0 = 2$, so W_b converges to 36.97. Fig. 3 shows the number of output bits per trellis section n_0 and the total number of ones in all the branches

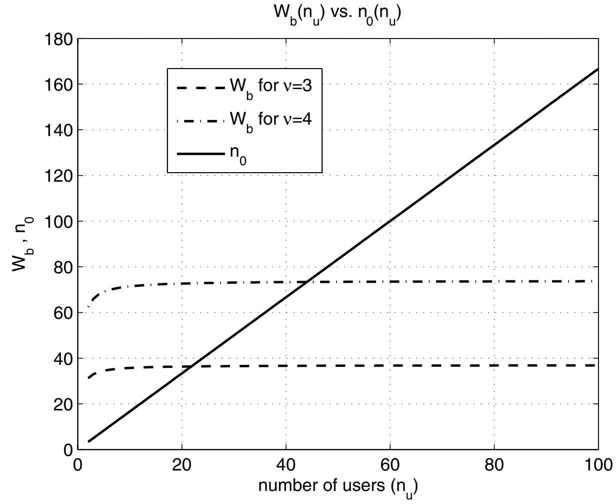

Fig. 3. Total number of ones in the output of all branches (W_b) and number of output bits per branch (n_0) vs. number of users (n_u).

W_b vs. the number of users, for a concatenation of 8-state ($\nu = 3$) and 16-state ($\nu = 4$) trellis codes. It can be seen that for the 8-state encoder case, n_0 is greater than the total number of ones in all branches for 22 users or more. In this case each of the W_b ones can be placed in a different position among the possible n_0 output bits. As the number of users increases, the number of output bits n_0 increases linearly, but the total number of ones remains the same. Thus, the best code for 22 users is the best code for any number of users greater than 22. The only thing that can be done is add zeros to the output. However, while the code strength cannot be improved, the crossover probability

$$\alpha(n_u) = 1 - (1 - p_1(n_u))^{n_u - 1} = 1 - (1/2)^{(n_u-1)/n_u}, \quad (31)$$

increases with the number of users. Hence, the performance of the code will degrade as the number of users increases above 22. As shown in Sec.VI, the performance for the 48-user case is very poor using 8-state encoders. A concatenation of 16-state trellis codes is required for the 48-user case, increasing the complexity.

VI. Performance Results

As a first example, we designed a PC-NLTC for the 6-user case ($n_u = 6$), using $k_0 = 0$ and $n_0 = 10$, which results on a sum-rate $R^+ = 0.6$. The trellis structure is the same as the one proposed on [11][14] for an 8-state ($\nu = 3$) turbo code. An interleaver length and input word length of 8192 was used. The optimal average number of ones per output branch is $\hat{w}_b \simeq 1.08$, which provides a ones density $p_1 = 0.108$. Exactly one 1 per output branch of ten bits was used. The resulting ones density is $p_1 = 1/10 = 0.1$, which corresponds to a crossover probability $\alpha = 0.40951$.

Fig. 4 shows the BER and FER in terms of the crossover probability α, and the uniform interleaver BER upper bound for ML decoding. The vertical line shows the crossover probability $\alpha = 0.40951$, corresponding to the 6-user OR-MAC with single-user decoding. The FER for the 6-user OR-

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2006 proceedings.

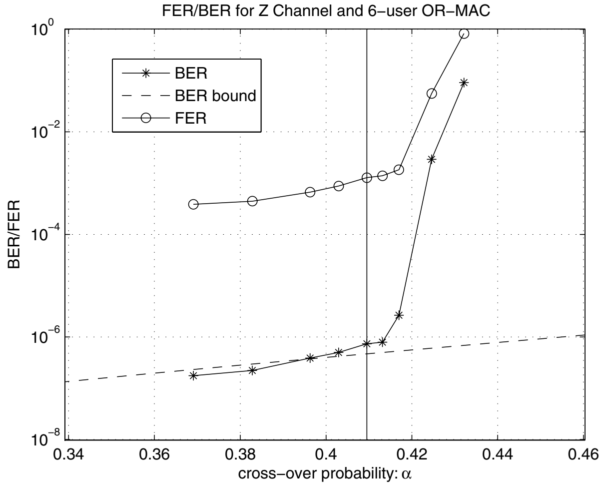

Fig. 4. BER, FER and BER bound vs. crossover probability α. Vertical line shows the crossover probability corresponding to the 6-user OR-MAC with single-user decoding

TABLE I

FER/BER FOR OR-MAC, FOR LARGE NUMBER OF USERS n_u

n_u	p_1	α	FER	BER
24	2.8125×10^{-2}	0.48115	6.34×10^{-4}	4.37×10^{-7}
30	2.25×10^{-2}	0.48312	1.01×10^{-3}	1.88×10^{-5}
48	1.4062×10^{-2}	0.48605	0.006125	2.58×10^{-4}
60	1.125×10^{-2}	0.48702	0.0150	6.05×10^{-4}
72	9.375×10^{-3}	0.48766	0.0260	1.13×10^{-3}
96	7.0312×10^{-3}	0.48846	0.0531	2.98×10^{-3}

MAC is 1.28×10^{-3}, and the BER is 7.34×10^{-7}. It can be seen that for a low α the BER bound for ML decoding and uniform interleaver is close to the actual BER on the simulations. Although for large crossover probabilities the iterative message passing algorithm diverges from the ML decoding bound, the bound predicts with accuracy the actual BER at the point of interest for the 6-user OR-MAC, $\alpha = 0.40951$.

In order to show quantitatively the limitation in the number of users for a fixed number of states, we designed a code for the 24-user case, for a target-rate of $R^+ = 0.6$ and $\nu = 3$. The total number of ones in all the branches is fixed to 36 for more than 22 users in order to satisfy the optimal ones density. Simulations were performed for 24, 30, 48, 60, 72 and 96 users. In all those cases $n_0 = n_u/0.6$ is an integer, and the sum-rate is 0.6. The ones density for each n_u is $p_1(n_u) = (36\cdot p_1)/(32\cdot n_u)$ and $\alpha = 1-(1-p_1)^{n_u-1}$. The best double-input 8-state trellis code concatenation for 24 users is the best code for 30, 48, 60, 72 and 96 users (with added zeros to the output). The only thing that changes is α, thus degrading the performance as α increases. Table I shows the FER and BER for each case. It can be observed that for 24 users, the performance is similar to the performance of the code designed for 6 users. However, as the number of users increases, α increases, and the performance is significantly degraded.

VII. CONCLUSIONS

This paper addresses the problem of designing parallel concatenated non-linear trellis codes for the Z-channel, along with an IDMA-based architecture that allows uncoordinated multiple access in the OR-MAC.

Moreover, a BER upper bound analysis for PC-NLTCs under ML decoding has been presented. This bounding technique, can be applied to any asymmetric channel, as long as an additive distance is defined. Simulation results show its accuracy on the regions where iterative message-passing decoding approaches ML decoding.

Also, an analysis on the limitation on the number of users, for a certain complexity, sum-rate and BER has been shown.

Simulation results for 6 users and 24 users, with a sum rate of 0.6 (slightly less that 0.1 bits below the theoretical SUD sum rate) show a BER below 10^{-6}.

REFERENCES

[1] A. Grant, B. Rimoldi, R. Urbanke, and P. Whiting. Rate-Splitting Multiple Access for Discrete Memoryless Channels. *IEEE Trans. Info. Theo.*, 47(3):873–890, Mar. 2001.

[2] Li Ping, K. Y. Wu, and L. Liu. A Simple, Unified Approach to Nearly Optimal Multiuser Detection and Space-time Coding. In *ITW 2002*, India, October 2002.

[3] Li Ping, Lihai Liu, and W. K. Leung. A Simple Approach to Near-Optimal Multiuser Detection: Interleave-Division Multiple-Access. In *IEEE Wireless Comm. and Networking Conference*, pages 391–396, 2003.

[4] Li Ping, Lihai Liu, Keying Wu, and W.K. Leung. Interleave-Division Multiple-Access. In *IEEE Trans. Wireless Comm.*, volume 5, pages 938–947, Apr 2006.

[5] T.M. Cover and Joy A. Thomas. Elements of Information Theory. In *Wiley Series in Telecommunications*, 1991.

[6] T. Kløve. Error correcting codes for the asymmetric channel. In *Dept. Mathematics, Univ. Bergen, Bergen, Norway, Tech. Rep.1809-0781*, 1995.

[7] Fang-Wei Fu, San Ling, and Chaoping Xing. New Lower Bounds and Constructions for Binary Codes Correcting Asymmetric Errors. In *IEEE Trans. Info. Theo.*, volume 49, pages 3294–3299, March 2003.

[8] Amir Bennatan and David Burshtein. On the Application of LDPC Codes to Arbitrary Discrete-Memoryless Channels. In *IEEE Trans. Info. Theo.*, volume 50, pages 417–438, March 2004.

[9] Edward A. Ratzer and David J.C. MacKay. Sparse Low-Density Parity-Check Codes for Channels with Cross-Talk. In *Proc. ITW 2003*, Paris, France, April 2003.

[10] M. Griot, A. Vila, W-Y. Weng, H. Chan, J. Basak, E. Yablanovitch, I. Verbauwhede, B. Jalali, and R. Wesel. Trellis Codes with Low Ones Density for the OR Multiple Access Channel. In *IEEE ISIT 2006*, July 2006.

[11] C. Berrou and A. Glavieux and P. Thitimajshima. Near shannon limit error-correcting coding and decoding: turbo-codes. In *Proc. ICC'93*, pages 873–890, May 1993.

[12] C. Berrou and . Jezequel. Non-binary convolutional codes for turbo coding. In *Electronics Letters*, volume 35, Jan 1999.

[13] C. Berrou, M. Jezequel, C. Douillard, and S. Kerouedan. The advantages of non-binary turbo codes. In *ITW Proc. 2001 IEEE*, pages 61–63, Sept 2001.

[14] C. Douillard and C. Berrou. Turbo codes with rate-m/(m+1) constituent convolutional codes. In *IEEE Trans. on Comm.*, volume 53, pages 1630–1638, Oct 2005.

[15] S. Benedetto and G. Montorsi. Unveiling turbo codes: Some results on parallel concatenated coding schemes. In *IEEE Trans. Info. Theo.*, volume 42, Mar 1996.

[16] E. Biglieri. High-Level Modulation and Coding for Nonlinear Satellite Channels. In *IEEE Trans. on Comm.*, volume 32, May 1984.

[17] Y. J. Liu and I. Oka and E. Biglieri. Error Probability for Digital Transmission Over non-Linear Channels with Application to TCM. In *IEEE Trans. Info. Theo.*, volume 36, Sep 1990.

1-4244-0357-X/06/$20.00 ©2006 IEEE

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2006 proceedings.