Nonlinear Turbo Codes for the AWGN Channel with Higher-Order Modulations and the OR Multiple Access Channel

Miguel Griot *Student Member, IEEE*, Andres I. Vila Casado *Student Member, IEEE* and Richard D. Wesel *Senior Member, IEEE.*

Abstract

This paper addresses the problem of designing parallel concatenated nonlinear trellis codes (PC-NLTCs). These codes have advantages over binary linear turbo codes for higher-order modulations or in applications where a non-uniform distribution of ones and zeros in the output is optimal. Two applications are considered in this work. First, this paper shows that for higher-order modulations using binary linear codes is too restrictive. Simulation results are shown for a 2 bits/s/Hz 16-state nonlinear turbo code with 8PSK. This code is less than 0.5 dB away from capacity at a BER $= 10^{-5}$ with an interleaver length of 10000 bits, and outperforms previous published linear turbo codes by around 0.2 dB. Simulation results are shown for each application. Second, this paper presents the use of PC-NLTC codes in an Interleaver-Division Multiple Access (IDMA)-based architecture with single-user decoding over the OR multiple access channel (OR-MAC). These PC-NLTCs are designed specifically for the Z-Channel that arises in an OR-MAC channel when each user treats the other users as noise. Over the 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 with single-user decoding, these codes employ a ones density of much less than 50%. To facilitate analysis of the new codes, an extension of Benedetto's uniform interleaver analysis to handle nonlinear constituent codes is presented.

Keywords

Channel coding, information rates, turbo codes, nonlinear codes, IDMA, interleaver division multiple access, interleave division multiple access, PSK, trellis codes, parallel concatenated trellis codes

I. Introduction

THIS paper demonstrates the benefits of using nonlinear constituent codes in parallel concatenated trellis codes for certain applications. In one application, when using higher-order modulations, there is an improvement in performance possible when the binary parallel concatenated trellis codes are permitted to be nonlinear. In another application, they provide a nonuniform distribution of transmitted ones and zeros, which cannot be provided by linear codes.

Trellis-Coded Modulation (TCM) was proposed by Ungerboeck in 1982 [1]. Ungerboeck presented a set-partitioning design technique which directly assigns constellation points to the branches of the trellis. However, it has been shown that the codes Ungerboeck designed

This work was supported by the Defence Advanced Research Project Agency under SPAWAR Systems Center San Diego Grant N66001-02-1-8938.
M. Griot, A. I. Vila Casado and R. D. Wesel are with the Electrical Engineering Department, University of California, Los Angeles, CA 90095 USA (e-mail:[mgriot,avila,wesel]@ee.ucla.edu).

may be represented by a linear convolutional code with a mapper that assigns a series of coded bits to a constellation point. This notion has remained with the appearance of turbo codes. Therefore, parallel concatenated trellis coded modulation (PC-TCM) has been traditionally designed using parallel concatenated convolutional codes with a bits-to-symbol mapper [2][3][4].

This paper shows that for higher-order modulations the use of parallel concatenated linear convolutional codes and a mapper constrains performance. Parallel concatenated *nonlinear* trellis coded modulation (PC-NLTCM), which directly assigns constellation points to the output branches of the constituent codes, can improve the performance. As an example, simulation results are shown for a 2 bits/s/Hz 16-state nonlinear turbo code with 8PSK. This code is less than 0.5 dB away from capacity at a BER $= 10^{-5}$ with an interleaver length of 10000 bits, and outperforms the best previous published linear turbo code by around 0.2 dB in the waterfall region. Moreover, this improvement in performance comes at no cost in increased implementation complexity since the decoding algorithm for linear or nonlinear constituent trellis codes is the same.

As a second application, consider the OR channel, or its isomorphic channel, the Binary Multiplier Channel [5], as a target application for Interleaver-Division Multiple Access (IDMA). There have been many contributions to the problem of providing multiple access. However, many common forms of multiple access, such as time-division (TDMA), frequency-division (FDMA), code-division (CDMA) or rate-splitting [6], require considerable coordination. The common approaches that do not require coordination, such as Aloha or CSMA, require re-transmission which increases the maximum delay. One recent successful approach for low-delay uncoordinated multiple-access is Interleaver-Division Multiple-Access (IDMA) [7][8][9], which uses interleaving to distinguish among signals from different users. Completely uncoordinated transmissions using IDMA and 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, while also providing low-delay transmission, 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. Nonlinear (and nonsystematic) codes are required to provide a ones density of much less than 50%, which is necessary to achieve the SUD sum capacity. Ratzer *et al.* have addressed the problem of designing codes with nonuniform distribution in the output, proposing sparse LDPC codes over large finite fields, *i.e.* using symbols from $GF(q)$ in the parity-check matrix [10]. However, this solution requires a much more complex decoder than binary LDPC codes, especially in the application considered in this work where the required low ones densities would lead to large values of q. We propose the use of parallel concatenated non-linear trellis codes (PC-NLTCs) which have the same decoding complexity as linear turbo codes, provide a wide range of ones densities and approach the approximately 70% SUD sum capacity.

To facilitate analysis of the new codes, an extension of Benedetto's uniform interleaver for nonlinear constituent codes is presented. It is shown that the same design criteria for linear turbo codes can be applied to nonlinear turbo codes. Namely, we generalize the notion of *effective free distance* for nonlinear codes, and show that this is an important metric to maximize when designing constituent codes for a PC-NLTC.

This paper is organized as follows. Section II shows the structure of the parallel concatenated nonlinear trellis structure. Section III shows an extension of Benedetto's uniform interleaver analysis to bound the bit-error rate of parallel concatenated nonlinear codes. Section IV proposes the use of PC-NLTCM with a higher-order modulation over the AWGN channel. Section V introduces the use of PC-NLTCs over the OR-MAC using IDMA with single-user decoding. Section VI delivers the conclusions.

II. Parallel concatenated nonlinear codes

The structure of the PC-NLTC encoder was introduced in [14]. It is in essence the well-known turbo-code structure first proposed in [11] for systematic linear encoders, except that the output label is assigned directly to each branch of the trellis by a look-up table rather than a linear function of the state and the input bits. A similar replacement of a linear operation with a look-up table has been successfully proposed for a decision feedback equalizer to equalize channels with trailing nonlinear inter-symbol interference [12]. Looking at Fig. 1, the encoder consists of two constituent nonlinear trellis encoders (labeled NLTC) linked by an interleaver (labeled Π). Each trellis encoder uses k_b input bits per trellis section. The NLTC includes 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 the output for each branch of the trellis. In the case of higher-order modulation applications, each output consists of n_0 constellation points, resulting in a total rate of $k_b/(2n_0)$ bits/symbol. In the OR-MAC application, each output consists of n_0 bits, resulting in an overall rate of $k_b/(2n_0)$. The look-up table is built so that the optimal ones distribution is transmitted. For the two applications considered in this work the trellis codes are non-systematic. Also, in case of $k_b > 1$, symbol interleaving [4] will be assumed. We will denote the input block length in bits as K_b, and the interleaver length in symbols as $K_s = K_b/k_b$. Also, we will denote the input symbol alphabet size as $q = 2^{k_b}$.

III. Error Rate Bound of Parallel Concatenated Nonlinear Codes

Benedetto and Montorsi proposed a method to evaluate the bit error probability of a parallel concatenated coding scheme averaged over all interleavers of a certain length in [13]. 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. For nonlinear codes all data-words need to be considered when finding the upper-bound. Thus, an extension of the bounding technique proposed in [13] for a parallel concatenation of **nonlinear** codes is required. In order to do that, a new probabilistic interleaver will be defined, which produces similar equations to the linear case.

Also, the analysis in [13] assumes a parallel concatenation of systematic codes. Since the codes used in this work are non-systematic, the new error upper-bounding technique will be derived assuming non-systematic constituent encoders. Nevertheless, it should be clear how to modify the equations in the case of systematic nonlinear codes. Finally, the analysis contemplates constituent encoders with more than one input bit per trellis section, *i.e.* $k_b \geq 1$. In that case, a symbol-interleaver [4] is assumed, and the symbol error rate (SER), i.e. the average number of k_b-bit symbols that are in error, will be computed. Note that for $k_b = 1$ the symbol error rate is the bit error rate.

A. Uniform symbol-interleaver with re-mapping for nonlinear codes

In this section we extend the uniform interleaver bounding technique in [13] to nonlinear constituent codes. The main difference is that for nonlinear codes we can no longer assume that the all-zero codeword is transmitted. We propose a new definition of uniform interleaver that extends the results, conclusions and design criteria drawn in [13] to nonlinear constituent codes.

Definition 1: A *Uniform Symbol-Interleaver with Re-mapping (USIR)* of length K_s (the number of input symbols) for nonlinear codes is a probabilistic device defined as follows: There are two operations considered in the interleaver. First, the uniform interleaver selects any of the $K_s!$ possible permutations of the symbol positions with equal probability. Second, for each position, the value of the symbol can be re-mapped to any of the $q = 2^{K_b}$ possible values with equal probability. The re-mapping can be different for different positions, but in a fixed position it must be an invertible function over the q-ary symbols, *i.e.* no two different symbols can be re-mapped to a same symbol.

The reason for this extension is that for nonlinear codes we need to consider all the possible input pairs. The uniform interleaver as defined in [13] would maintain the Hamming weight of both input words and their Hamming distance, which would make the equations more complicated and would make it harder to draw conclusions from them (see [14]). With this new definition, any input word can be mapped to any other input word, no matter their Hamming weight. Thus, the only value preserved after the interleaver is the symbol-wise Hamming distance between any two input pairs. This is a generalization of the analysis for linear codes, since the Hamming weight of the erroneous input word, which is the value preserved in [13] is the Hamming distance between the correct input word (the all-zero word) and the erroneous word.

Any pair of input words U_m and U_n such that $d_H(U_m, U_n) = i$, can be mapped by the USIR to any other pair of input words satisfying $d_H(\Pi(U_m), \Pi(U_n)) = i$ with probability:

$$P\Big((U_m, U_n) \to (\Pi(U_m), \Pi(U_n))\Big| d_H(U_m, U_n) = d_H(\Pi(U_m), \Pi(U_n)) = i\Big) = \frac{1}{q^{K_s} \cdot (q-1)^i \cdot \binom{K_s}{i}}. \tag{1}$$

Consider any two output codewords X_m and X_n. The technique presented in this work is valid for any channel for which an additive distance can be defined and for which the pair-wise error probability can be upper-bounded by:

$$P(X_m \to X_n) + P(X_n \to X_m) \leq \nu(\lambda^{d(X_m,X_n)} + \lambda^{d(X_n,X_m)}) \tag{2}$$

where the directional distance metric $d(X_m, X_n)$, and the parameters ν and λ depend on the channel. Note that $d(X_m, X_n)$ and $d(X_n, X_m)$ may not be equal in asymmetric channels. As we will see in Sections IV and V, both the pair-wise probability of error of the AWGN and the Z-Channel can be upper-bounded by an expression of the form shown in (2). Considering the sum of both directional pair-wise error probabilities in (2) will be helpful when finding the error bound over the Z-Channel, and is generally useful as shown in [15]. Define the *Input-Output Distance Enumerating Function* (IODEF) of an (n, K_s) code C as

$$A^C(I, D) = \sum_{i,d} A^C_{i,d} I^i D^d, \tag{3}$$

where $A^C_{i,d}$ is the number data-word pairs $(U, \hat{U})$ that satisfy $d_H(U, \hat{U}) = i$, and the directional distance between the corresponding codewords $d(X_m, X_n) = d$. I and D are placeholders.

Also define the *Conditional IODEF* (CIODEF) as:

$$A^C_i(D) = \sum_d A^C_{i,d} D^d. \tag{4}$$

Inserting Eq. (4) in Eq. (3), the expression for the IODEF can be rewritten as:

$$A^C(I, D) = \sum_i A^C_i(D) I^i. \tag{5}$$

Using (2) and (3) the symbol error rate (SER) or bit error rate (BER) in case $k_b = 1$ can be upper bounded by (see Appendix I):

$$\text{SER} \leq \frac{\nu}{K_s} \cdot (1/q)^{K_s} \left. \frac{\partial A^C(I, D)}{\partial I} \right|_{D=\lambda, I=1}. \tag{6}$$

B. Parallel concatenation of block codes

Denote C_P as the $(n_1 + n_2, K_s)$ block code resulting from the parallel concatenation of two codes, an (n_1, K_s) block code C_1 and an (n_2, K_s) block code C_2. We will assume an interleaver of length K_s symbols, equal to the input word length, in order to simplify the analysis (An extension can easily be made for the case when l consecutive codewords of the constituent codes are used for one operation of the interleaver, as explained in [13]). 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 conditional IODEF of C_P can be expressed (using (1)) as:

$$A_i^{C_P}(D) = \frac{A_i^{C_1}(D) \cdot A_i^{C_2}(D)}{q^{K_s} \cdot (q-1)^i \cdot \binom{K_s}{i}}. \tag{7}$$

Notice that the USIR as defined in Sec. III-A can map any input word to any other input word. Now, using (7) and (3) in (6), it can be observed that there are two terms of the form $(1/q)^{K_s}$, corresponding to the probability of the correct input word and the probability of that input word being mapped to any other word after the interleaver. Define the *Normalized Input-Output Distance Enumerating Function* (NIODEF) of a given (n, K_s) code C as

$$\tilde{A}^C(I, D) = \sum_{i,d} \tilde{A}_{i,d}^C I^i D^d, \tag{8}$$

where $\tilde{A}_{i,d}^C = A_{i,d}^C / q^{K_s}$. Hence, the symbol error probability can be upper bounded by:

$$\text{SER} \leq \frac{\nu}{K_s} \frac{\partial \tilde{A}^C(I, D)}{\partial I}\bigg|_{D=\lambda, I=1}. \tag{9}$$

Now, using (4) and (8):

$$\tilde{A}_i^{C_P}(D) = \frac{\tilde{A}_i^{C_1}(D) \cdot \tilde{A}_i^{C_2}(D)}{(q-1)^i \cdot \binom{K_s}{i}}. \tag{10}$$

Note that except for the term $1/q^{K_s}$ in $\tilde{A}_{i,d}$, and the term $1/(q-1)^i$ the equations (8)-(10) for a parallel concatenation of nonlinear codes are the same as for the linear case [13]. As it turns out, all the conclusions and design criteria derived in [13] apply to nonlinear constituent codes. See Appendix II for a thorough derivation. In particular, it is shown that feed-forward encoders are not suitable for parallel concatenation, and that recursive convolutional codes are required. Moreover, just as in [13] an important parameter to maximize is the effective free distance, which we generalize for nonlinear codes as:

Definition 2: *Effective free distance* of a constituent code is the minimum distance $d(X_m, X_n)$ between the two outputs corresponding to any two possible input words U_m and U_n with input Hamming distance $d_H(U_m, U_n) = 2$.

IV. Nonlinear Turbo Codes for Higher-Order Modulations over AWGN

As expressed in Section I, PC-TCM has been traditionally designed using parallel concatenated convolutional codes with a bits-to-symbol mapper. However, using linear codes turns out to be too restrictive for higher-order modulations. In this section we show with an example that nonlinear codes improve performance for PC-TCM over AWGN using 8PSK.

For the AWGN case, the pairwise probability of error can be upper bounded by:

$$P_e(X_m \rightarrow X_n) = Q\left(\sqrt{d_E^2(X_m, X_n)\frac{E_s}{2N_0}}\right) \leq \frac{1}{2}e^{-\frac{E_s}{4N_0}d_E^2(X_m,X_n)}, \tag{11}$$

where E_s/N_0 is the signal-to-noise ratio and $d_E^2(X_m, X_n)$ is the squared Euclidean distance assuming unity power transmission. Thus we obtain (2) with $d(X_m, X_n) = d_E^2(X_m, X_n)$, $\nu = 1/2$ and $\lambda = e^{-\frac{E_s}{4N_0}}$. In the following example we will try to maximize the effective squared Euclidean distance, and show it can be increased using PC-NTCM.

A. Design Example, 2-bits/s/Hz 16-state PC-NLTCM with 8PSK

In this section we will show that directly assigning constellation points to the trellis branches of each constituent code can produce codes that outperform linear codes with mapping. As an example, we will design a 2-bits/s/Hz 16-state PC-NLTCM with 8PSK and compare its performance against the 16-state turbo code presented in [4]. In order to make a fair comparison, we will use the same spread-interleaver technique used in that work and the same interleaver length $K_b = 10000$ bits, and therefore $K_s = 2500$ symbols with $k_b = 4$. Each output branch of each constituent encoder consists of one 8PSK constellation point ($n_0 = 1$), which produces a code rate of 2 bits/s/Hz. This is an interesting comparison since there hasn't been any published work that shows a turbo code with symbol interleaving that outperforms the code presented in [4] under same conditions.

The code in [4] has an effective free distance of $d_{\text{eff,free}} = 1.171573$. We present a 16-state PC-NLTM that has $d_{\text{eff,free}} = 2$. For the design, we make the following observations. Since $k_b = 4$, there are 16 branches leaving each state with each of the 16 possible inputs. It is clear that parallel branches should be avoided, so the trellis structure is fully connected, *i.e.* there is one (and only one) branch connecting each of the 16 origin states with each of the 16 destination states. The design consists of assigning each branch and input symbol and an 8PSK constellation point. These assignments are constrained by the following conditions:

- Branches starting at a same state cannot be produced by the same input symbol.
- Branches merging to a same state cannot be produced by the same input symbol. This constraint avoids error events with input Hamming distance equal to 1 and can be satisfied by using recursive encoders.

Note that since the trellis is fully connected, any two branches leaving a same state at a certain trellis section will produce 16 error events with input Hamming distance equal to 2 in the following trellis section. In other words, there are 16 length-two error events starting at each of the 16 states, which have an input Hamming distance of 2. Thus the effective free distance of the code is upper bounded by these length-two error events. A first step in the design is to assign output labels to each branch so that the minimum distance produced by a length-two error event is maximized. Given the constraints stated above, there is no need to consider the input symbols at this stage. Table I shows the output branch label assignment. The constellation labeling for 8PSK used in this work is shown in Fig. 2. Each row represents the starting state (S_s) of the branch, and each column represents the ending state (S_e). This output labeling produces a minimum length-two error-event distance of 2, assuming a unit-norm constellation.

The next step is to search over all the possible input symbol assignments in order to avoid error events of length three or more that have input Hamming distance of two, and output distance of less than 2. From the conclusions drawn from the USIR analysis (Section III-B and Appendix I), this search must be constrained to recursive trellis structures. We searched over trellis structures of the form:

$$S = A \cdot S + B \cdot u \mod 2, \tag{12}$$

where $S = [s_1, s_2, s_3, s_4]^T$ represents the state, and $u = [u_1, u_2, u_3, u_4]$ represents the input symbol. The trellis structure selected is given by:

$$A = \begin{bmatrix} 0\,0\,0\,1 \\ 0\,1\,1\,1 \\ 1\,0\,1\,1 \\ 1\,0\,0\,0 \end{bmatrix}, B = \begin{bmatrix} 1\,0\,0\,0 \\ 0\,1\,0\,0 \\ 0\,0\,1\,0 \\ 0\,0\,0\,1 \end{bmatrix} \tag{13}$$

This trellis structure together with the output labeling of Table I produces a $d_{\text{eff,free}} = 2$.

Fig. 3 shows a performance comparison between the 2 bits/s/Hz 16-state turbo code for proposed in [4], and the PC-NLTCM presented in this work. The same symbol interleaver has been used for both codes. The interleaver length is $K_b = 10000$ bits, or $K_s = 2500$ symbols from the symbol interleaving perspective. It can be observed that the nonlinear code outperforms the linear code by a little less than 0.2 dB. At $BER = 10^{-5}$, the PC-NLTCM is within 0.5 dB from the constrained capacity 2.8 dB.

Fig. 3 also shows the uniform-interleaver BER bounds for each code. The reason why the

BER bound is not tight in the error floor is that the interleaver design plays an important role in these high-rate applications, as shown in [4], and therefore an average interleaver would perform much worse than the carefully designed one used here. However, at the constituent code design stage, it gives a good prediction of which constituent code would perform better than the other.

It is worth mentioning that this is merely one example where constraining the design to a linear code with a mapper could be too restrictive, and directly assigning constellation points to each branch could produce a larger effective free distance and a better parallel concatenated code. General nonlinear turbo code design is a rich area for continued research.

V. Uncoordinated Multiple Access in the OR Channel : the Z-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 transmitter and the receiver side, to the Binary Multiplier Channel [5]. 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 succesive decoding if the transmitted ones densities and rates are carefully controlled [6]. All of these solutions require either coordination of all users or a very complex decoder, especially for a large number of users.

As in [16] we consider 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. 4. 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:

$$\alpha = 1 - (1 - p_1)^{n_u - 1}, \tag{14}$$

which is a function of the number of users and the ones density employed by the users.

The maximum theoretical sum-rate with single-user decoding decreases as the number of users increases, but it converges monotonically and rapidly 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 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 nonlinear parallel concatenated 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 for the Z-Channel with the desired transmitted 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

$$p_1(n_u) \simeq 1 - (1/2)^{1/n_u}. \tag{15}$$

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 $R^+ = 0.6$, since an excess mutual information requirement 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 to the target sum-rate:

$$n_u \cdot \big(k_b/(2 \cdot n_0)\big) \simeq R^+. \tag{16}$$

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

$$\hat{w}_b \simeq p_1(n_u) \cdot n_0. \tag{17}$$

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

$$\hat{w}_b \simeq p_1(n_u) \cdot n_0 = \frac{p_1(n_u) \cdot n_u}{2 \cdot R^+} \cdot k_b \simeq 0.54 \cdot k_b. \tag{18}$$

If single-input encoders are used ($k_b = 1$), 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 $k_b \geq 2$ are required. Multiple-input convolutional codes for turbo coding have been studied in [17][18][19] among other papers.

Using a trellis structure with $k_b = 2$, 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 $\hat{w}_b$.

A. Pairwise error probability for the Z-Channel

Let X_m and X_n be any two possible codewords of length N_b bits. The *Directional Hamming Distance* for the Z-Channel $d_D(X_m, X_n)$ is the number of positions where $X_m(i) = 0$ and $X_n(i) = 1$, with $i = 1, \cdots, N_b$. Note that $d_D(X_m, X_n)$ is not necessarily equal to $d_D(X_n, X_m)$.

Let $Y = \{Y(1), \cdots, Y(N_b)\}$ be the received word. Given Y, any possible transmitted codeword X must satisfy $d_D(Y, X) = 0$, since there cannot be any one-to-zero transitions on the Z-Channel. The most likely transmitted codeword $\hat{X}$, is the codeword X satisfying $d_D(Y, X) = 0$, that minimizes the number of zero-to-one transitions. Hence, the maximum likelihood decoder for the Z-Channel chooses the codeword $\hat{X}$ as:

$$\hat{X} = \text{argmin}_{X \in \mathcal{N}} \Big[d_D(X, Y) \Big], \tag{19}$$

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

With α as the probability of a zero-to-one transition in the Z-Channel (see Fig. 4), Eq. (19) can be used to derive the probability of transmitting X_m and decoding X_n under ML decoding to be:

$$P_e(X_m \to X_n) = \begin{cases} \frac{1}{2} \cdot \alpha^{d_D(X_m,X_n)} & , W_H(X_m) = W_H(X_n) \\ \alpha^{d_D(X_m,X_n)} & , W_H(X_m) < W_H(X_n) \\ 0 & , W_H(X_m) > W_H(X_n). \end{cases}$$

where $W_H(\cdot)$ denotes the Hamming weight. Therefore, considering the sum of the two directional pair-wise error probabilities:

$$P_e(X_m \to X_n) + P_e(X_n \to X_m) = \alpha^{\max(d_D(X_m,X_n),d_D(X_n,X_m))} \leq \frac{1}{2}[\alpha^{d_D(X_m,X_n)} + \alpha^{d_D(X_n,X_m)}], \tag{20}$$

which is of the form of (2), with $\nu = 1/2$ and $\lambda = \alpha$, and with the directional distance $d_D(X_n, X_m)$ as the distance metric.

B. Performance Results for the OR-MAC

As a first example, we designed a PC-NLTC for the 6-user case ($n_u = 6$), using $k_b = 2$ and $n_0 = 10$, which results on a sum-rate $R^+ = 0.6$. The trellis structure is the same as the one proposed on [19] for an 8-state ($\nu = 3$) turbo code. An interleaver 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. 5 shows the BER and FER in terms of the crossover probability α, and the USIR BER upper bound for ML decoding. The diamond shapes show the crossover probability $\alpha = 0.40951$ corresponding to the 6-user OR-MAC with single-user decoding. The FER for the 6-user OR-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 is close to the actual BER in the simulations. For large crossover probabilities the iterative message passing algorithm diverges from the ML decoding bound, as is the case for standard turbo codes in the AWGN channel at low SNR. 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 plot the BER bound and not the SER bound, we assumed that any symbol error is equally likely, the symbol in error is equally likely to have any of the $2^{k_b} - 1$ possible values (leaving out the correct symbol), and therefore we used a correction factor on the error bound of

$$BER^{\text{bound}} \approx \frac{k_b \cdot 2^{k_b-1}}{2^{k_b}-1} SER^{\text{bound}}. \tag{21}$$

C. Limitation on the number of users

As mentioned in Sec. V, 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_b, fixed number of states ν, fixed target sum-rate R^+, and a fixed maximum tolerable BER, there may be a limitation on number of users n_u.

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

$$W_b \simeq p_1(n_u) \cdot n_0 \cdot 2^{k_b+\nu}. \tag{22}$$

Given a certain number of users n_u, and using (15-22), the total number of ones in all the $2^{k_b+\nu}$ branches can be rewritten as:

$$W_b(n_u) \simeq \Big(\frac{k_b \cdot 2^{k_b+\nu}}{2 \cdot R^+}\Big) \cdot \Big(n_u \cdot (1-(1/2)^{1/n_u})\Big). \tag{23}$$

Now,

$$\lim_{n_u \to \infty} n_u \cdot (1-(1/2)^{1/n_u}) = \ln 2, \tag{24}$$

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

$$W_b(n_u) \to \ln 2 \cdot \Big(\frac{k_b \cdot 2^{k_b+\nu}}{R^+}\Big), \tag{25}$$

for a large enough number of users. For example, in the results shown in Sec. V-B, $R^+ = 0.6$ and $k_b = 2$, so W_b converges to 36.97. Fig. 6 shows the number of output bits per trellis section n_0 and the total number of ones in all the branches 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 essentially the best code for any number of users greater than 22. The only difference additional zeros concatenated to the output. However, while the code strength cannot be improved, the crossover probability

$$\alpha(n_u) = 1 - (1 - p_1(n_u))^{n_u - 1} = 1 - (1/2)^{(n_u-1)/n_u}, \tag{26}$$

increases with the number of users. Hence, the performance of the code will degrade as the number of users increases above 22.

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 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 II 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. Hence, for more than 24 users, a 16 state PC-NLTC should be used.

Fig. 7 shows the performance of a 16-state PC-NLTC designed for a 44-user OR-MAC. The ones density of the code is $p_1 = 1/74 \approx 0.0135$, a little less than the optimal density $p_1 = 1 - (1/2)^{1/44} \approx 0.0156$. The sum-rate is $R^+ = 44/74 \approx 0.595$, and the interleaver-length is $K_b = 8192$. Again, the diamond shapes show the crossover probability $\alpha = 0.442918$ corresponding to the 44-user OR-MAC with single-user decoding. The FER is still around 10^{-3}, but the BER is above 10^{-4}. This is due to the fact that the crossover probability corresponding to the 44-user OR-MAC falls in the waterfall region of the performance of the code, as opposed to the 6-user OR-MAC case where it falls in the error floor region. For more than 44 users, either the number of states or k_b should be increased to maintain the performance, increasing the decoding complexity as well. Fig. 7 also shows the performance of the same code for 43 users. In that case the sum-rate is $R^+ = 43/74 \approx 0.581$ and the BER is around 10^{-7}.

VI. Conclusions

This paper addressed the problem of designing parallel concatenated nonlinear trellis codes for two applications. The first application is the transmission of 8PSK over the AWGN channel where PC-NLTC improves performance over parallel concatenated binary linear trellis codes with bits-to-constellation-point mapping. The second application is the uncoordinated multiple access to the OR channel with single user decoding where a non-uniform distribution of ones and zeros in the transmission is required.

Parallel concatenated nonlinear trellis codes can be beneficial for higher-order modulations. Although trellis coded modulation can achieve optimal performance using convolutional codes with a proper labeling, we showed with an example that for parallel concatenated trellis coded modulation using convolutional codes with labeling may be suboptimal under certain scenarios. As an example, we have designed a rate 2 bits/s/Hz 16-state parallel concatenated nonlinear trellis code for 8PSK, which outperforms the best previously reported linear turbo code with labeling by 0.2 dB over AWGN under same conditions. This code is within 0.5 dB away from capacity at a $BER = 10^{-5}$. Moreover, this improvement comes with the same decoding complexity as with convolutional codes as constituent codes.

For the second application, parallel concatenated nonlinear trellis codes have been designed for the Z-channel, along with an IDMA-based architecture that allows uncoordinated multiple access in the OR-MAC. With single-user decoding and no coordination a maximum sum-rate of around 0.7 can be achieved. Simulation results using an 8-state PC-NLTC 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) and an interleaver length of 8192 bits show a BER below 10^{-6}. Also, an analysis on the limitation on the number of users, for a certain complexity, sum-rate and BER has been shown. For a number of users between 24 and 44, a 16-state PC-NLTC must be used to maintain a good performance. For more than 44 users, a PC-NLTC with more states or more input bits per trellis sections would be required.

To facilitate analysis of the new codes, an extension of Benedetto's uniform interleaver analysis for nonlinear constituent codes was derived. It was shown that the design criteria for linear codes can be generalized to nonlinear codes. In particular, we generalize the notion of effective free distance for nonlinear constituent codes, and conclude that this is an important parameter to maximize at the design stage.

References

[1] Gottfried Ungerboeck. Channel coding with multilevel/phase signal. *IEEE Trans. on Info. Theory*, pages 55–67, Jan 1982.

[2] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara. Parallel Concatenated Trellis Coded Modulation. In *ICC '96*, June 1996.

[3] D. Divsalar and F. Pollara. Turbo trellis coded modulation with iterative decoding for mobile satellite communications. In *IMSC '97*, June 1996.

[4] C. Fragouli and R.D. Wesel. Turbo-encoder design for symbol-interleaved parallel concatenated trellis-coded modulation. *Trans. on Info. Theo.*, 47:873–890, Mar 2001.

[5] T.M. Cover and Joy A. Thomas. Elements of Information Theory, Chapter 14, Example 14.3.2. *Wiley Series in Telecommunications*, 1991.

[6] A.J. Grant, B. Rimoldi, R.L. Urbanke, and P.A. Whiting. Rate-splitting multiple access for discrete memoryless channels. *IEEE Trans. on Info. Theory*, 47:873–890, Mar 2001.

[7] 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, Oct 2002.

[8] 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.

[9] Li Ping, Lihai Liu, Keying Wu, and W.K. Leung. Interleave division multiple-access. *IEEE Transactions on Wireless Communications*, 5:938–947, April 2006.

[10] E.A. Ratzer and D.J.C. MacKay. Sparse low-density parity-check codes for channels with cross-talk. *Info. Theo. Workshop*, pages 127–130, April 2003.

[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] K.D. Fisher, J.M. Cioffi, W.L. Abbott, P.S. Bednarz, and C.M. Melas. An adaptive RAM-DFE for storage channels. *IEEE Trans. on Comm.*, 39:1559–1568, Nov 1991.

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

[14] M. Griot, A.I. Vila Casado, and R.D. Wesel. Non-linear turbo codes for interleaver-division multiple access on the OR channel. In *GLOBECOM '06. IEEE Global Telecomm. Conf.*, 27 Nov. - 1 Dec. 2006.

[15] Miguel Griot, Wen-Yen Weng, and Richard D. Wesel. A tighter bhattacharyya bound for decoding error probability. *IEEE Communications Letters*, 11, April 2007.

[16] 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.

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

[18] 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.

[19] C. Douillard and C. Berrou. Turbo codes with rate-m/(m+1) constituent convolutional codes. *IEEE Trans. on Comm.*, 53(10):1630–1638, Oct 2005.

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

[21] Y. J. Liu, I. Oka, and E. Biglieri. Error probability for digital transmission over non-linear channels with application to TCM. *IEEE Trans. on Info. Theory*, 36, Sep 1990.

Appendix

I. SER bound expressed in terms of the IODEF

Denote as $\mathcal{U}$ all the possible q^{K_s} input words. Then the symbol error rate (SER) can be upper bounded by the union bound:

$$SER \leq \frac{1}{K_s} \sum_{U_i \neq U_j \in \mathcal{U}} d_H(U_i, U_j) P(X_i \to X_j) = \frac{1}{2K_s} \sum_{U_i \neq U_j \in \mathcal{U}} d_H(U_i, U_j) \big[P(X_i \to X_j) + P(X_j \to X_i) \big]. \tag{27}$$

Using (2) then

$$SER \leq \frac{\nu}{2K_s} \sum_{U_i \neq U_j \in \mathcal{U}} d_H(U_i, U_j) \big[\lambda^{d(X_i \to X_j)} + \lambda^{d(X_j \to X_i)} \big] = \frac{\nu}{K_s} \sum_{U_i \neq U_j \in \mathcal{U}} d_H(U_i, U_j) \lambda^{d(X_i \to X_j)}. \tag{28}$$

Using the definition of IODEF in (4) then:

$$SER \leq \frac{\nu}{K_s} (1/q)^{K_s} \sum_{i,d} i A^C_{i,d} \lambda^d = \frac{\nu}{K_s} (1/q)^{K_s} \frac{\partial A^C(I, D)}{\partial I} \bigg|_{D=\lambda, I=1}. \tag{29}$$

II. Computing the SER Bound for Constituent Nonlinear Trellis Codes

Biglieri et al. presented a union bound in [20][21] for general trellis codes, using a $2^{2\nu}$-state trellis diagram. This concept can be used to find $A^{C_P}(I, D)$ for the case of parallel concatenated nonlinear trellis codes.

As in [20], 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

$$S_G = \{(s_e, s_r) \mid s_e = s_r\},\ S_B = \{(s_e, s_r) \mid s_e \neq s_r\}. \tag{30}$$

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

$$S(I, D) = \left[\begin{array}{c|c} S_{GG}(I, D) & S_{GB}(I, D) \\ \hline S_{BG}(I, D) & S_{BB}(I, D) \end{array}\right], \tag{31}$$

where the $N \times N$ matrix $S_{GG}(I, D)$ accounts for the transitions between good product states, the $N \times (N^2 - N)$ matrix $S_{GB}(I, D)$ 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:

$$(1/q) I^{d_H(u_e, u_r)} D^{d(x_e, x_r)}, \tag{32}$$

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 since, there are $q = 2^{k_b}$ possible inputs per trellis branch, $(1/q)$ is the the probability of each branch transition given a certain current state.

Although $\tilde{A}^C(I, D)$ can be computed using $S(I, D)$, it becomes very complex. To reduce complexity, two approximations can be made. *Approximation 1*: Use the same idea presented in [13]: 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. *Approximation 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:

$$E(I, D) = p_s\{S_{GB}(I - S_{BB})^{-1} S_{BG}\}\mathbf{1}, \tag{33}$$

where $p_s = [\frac{1}{N} \frac{1}{N} \cdots \frac{1}{N}]$ is the probability distribution of the encoder states and $\mathbf{1} = [\mathbf{1}\mathbf{1} \cdots \mathbf{1}]^\mathbf{T}$. Then, $E(I, D)$ can be written as:

$$E(I, D) = \sum_{i,d} e_{i,d} D^d I^i. \tag{34}$$

Now define:

$$E_j(I,D) = \Big[E(I,D)\Big]^j = \sum_{i,d} e_{i,d,j} I^i D^d, \tag{35}$$

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

$$K[l,j] \leq \binom{K_s - l + j}{j} \approx \frac{K_s^j}{j!}, \tag{36}$$

ways in the trellis. Note that the two sides of the inequality in (36) are not exactly equal, since the error events start at a particular state, and there might be positions where the concatenation of two error events is not possible. However, for K_s large the upper bound becomes very tight. Also, the symbols of the rest of the $K_s - l$ positions of both input words are equal and could be almost any of the possible $q^{K_s - l}$ combinations, which divided by the term q^{K_s} appearing in $\tilde{A_{i,d}}$ gives q^{-l} which is already counted by the terms $(1/q)$ appearing in the branch labels in $S(I,D)$ (see (32)). The approximation in (36) follows from the fact that $K_s >> l$, $K_s >> j$ and the Stirling approximation $\binom{k}{i} \approx k^i/i!$ for $k >> i$. Therefore, for each constituent code,

$$\tilde{A^C} \approx \sum_j \frac{K_s^j}{j!} E_j(I,D). \tag{37}$$

Using (9), (10) and (37), and using again Stirling for $K_s >> j$, we get:

$$SER \approx \sum_{i,j_1,j_2,d_1,d_2} \nu \frac{ii!}{j_1! j_2!} \frac{K_s^{(j_1+j_2-i-1)}}{(q-1)^i} e^{C_1}_{i,d_1,j_1} e^{C_1}_{i,d_2,j_2} \lambda^{d_1+d_2}. \tag{38}$$

Therefore, as K_s increases, the performance of the code will be driven by the terms with the largest possible value of $(j_1 + j_2 - i - 1)$. For recursive encoders, that happens for a concatenation of error events with $i = 2$. Therefore, an important parameter to maximize is the effective free distance as defined in Sec. III-B. As for linear encoders, feed-forward encoders lead to poor performance since i can be equal to 1 in which case $j_1 + j_2 - i - 1 = 0$.

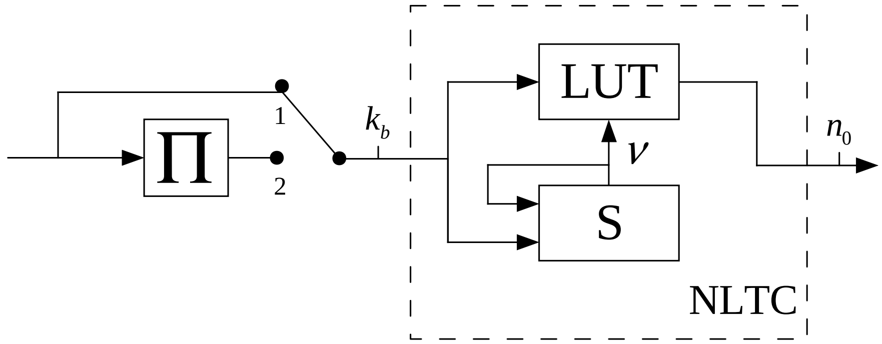

Fig. 1. PC-NLTC structure.

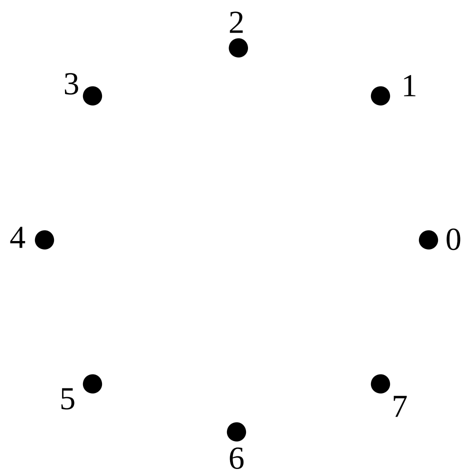

Fig. 2. Labeling for 8PSK.

TABLE I

Output labels for 8PSK. The rows indicate the starting states S_s, and the columns the ending states S_e. s_1/s_2 indicates that the output label is the same for both ending states.

$S_s : S_e$	0/8	1/9	2/10	3/11	4/12	5/13	6/14	7/15
0	0	1	2	3	4	5	6	7
1	4	5	6	7	0	1	2	3
2	0	3	2	5	4	7	6	1
3	4	7	6	1	0	3	2	5
4	0	1	2	3	4	5	6	7
5	4	5	6	7	0	1	2	3
6	0	3	2	5	4	7	6	1
7	4	7	6	1	0	3	2	5
8	2	3	4	5	6	7	0	1
9	6	7	0	1	2	3	4	5
10	2	1	4	3	6	5	0	7
11	6	5	0	7	2	1	4	3
12	2	3	4	5	6	7	0	1
13	6	7	0	1	2	3	4	5
14	2	1	4	3	6	5	0	7
15	6	5	0	7	2	1	4	3

TABLE II

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}

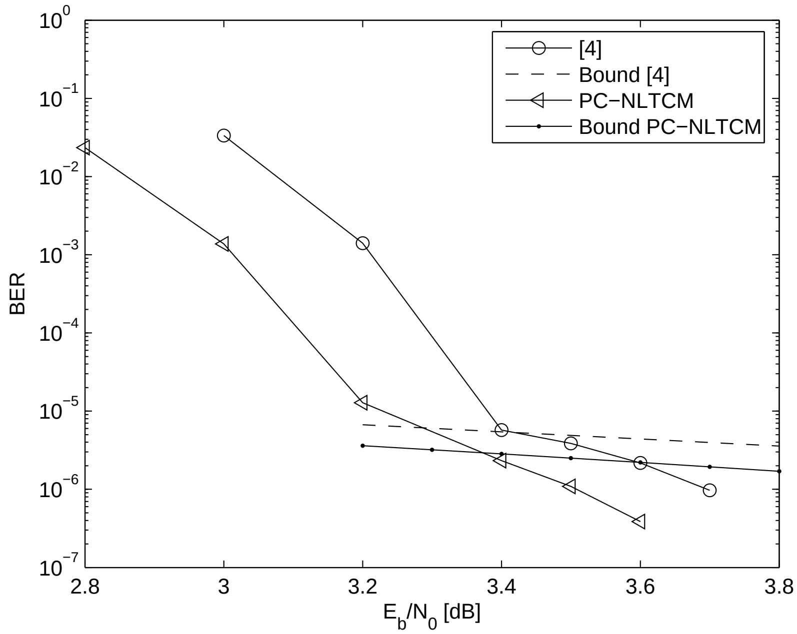

Fig. 3. BER vs. E_b/N_0 comparison, for 2 bits/s/Hz 16-state parallel concatenated codes with 8PSK over AWGN. Interleaver length $K_b = 10000$ bits.

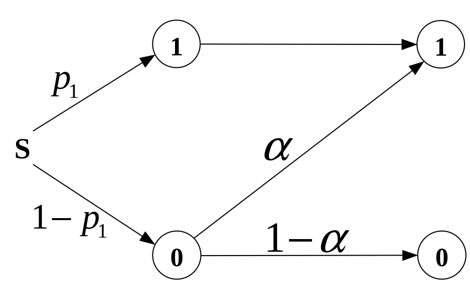

Fig. 4. Z-channel resulting from the OR-MAC channel when other users are treated as noise, and all users employ ones density p_1.

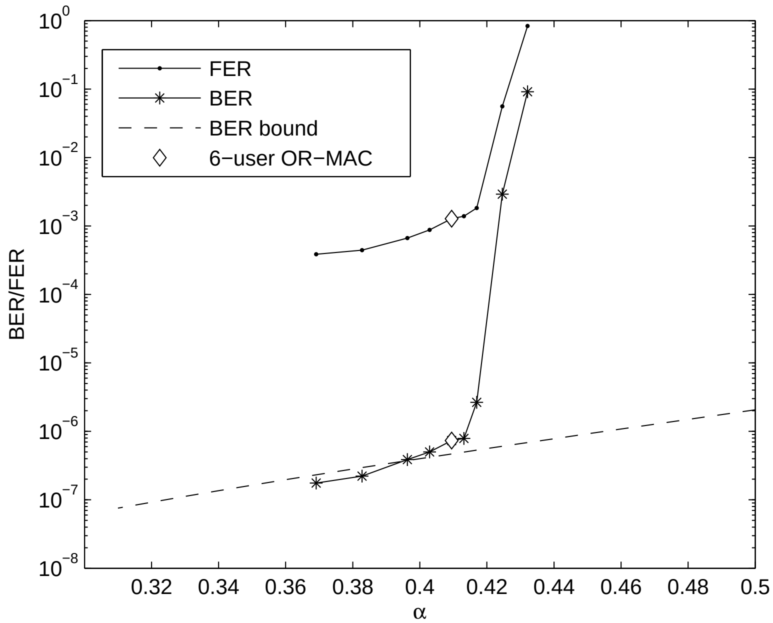

Fig. 5. 8-state PC-NLTC for 6-user OR-MAC. Interleaver length = 8192. BER, FER and BER bound vs. crossover probability α. Diamond shape indicates the BER/FER for the α corresponding to a 6-user OR-MAC

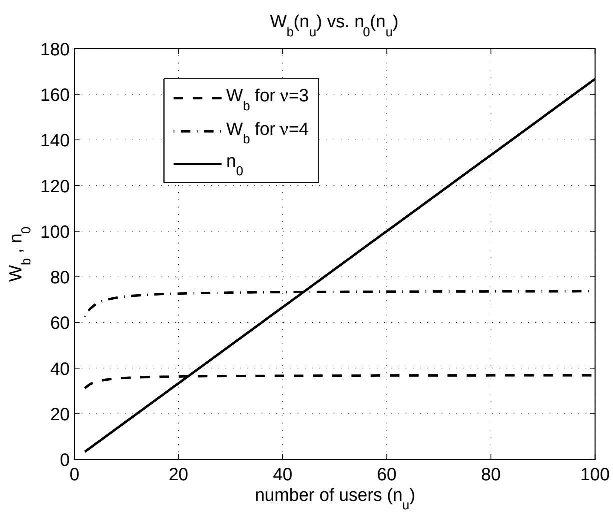

Fig. 6. Total number of ones in the output of all branches (W_b) and number of output bits per branch (n_0) vs. number of users (n_u).

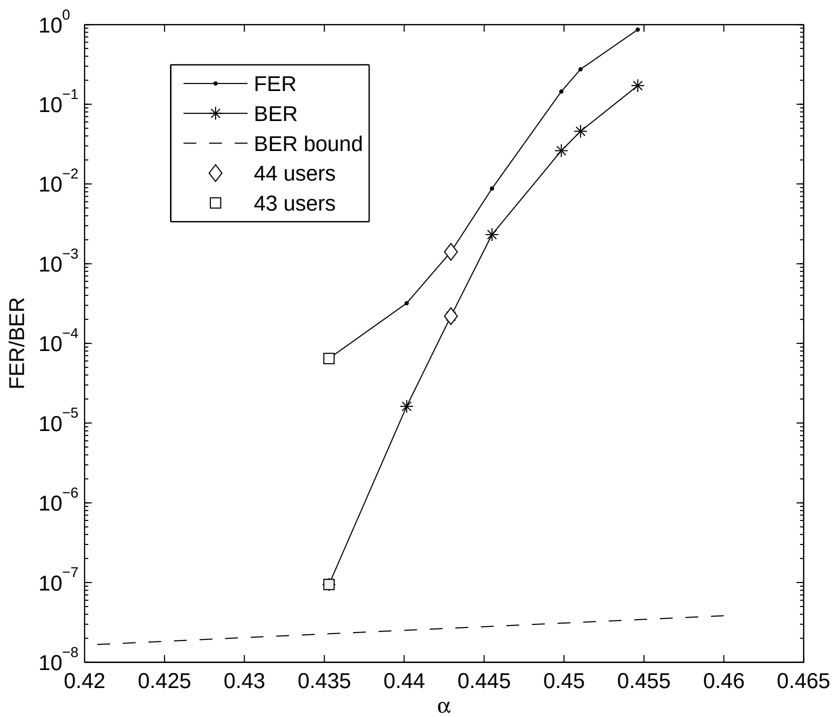

Fig. 7. 16-state PC-NLTC for 44-user and 43-user OR-MAC. Interleaver length = 8192. BER, FER and BER bound vs. crossover probability α. Diamond and square shapes indicate the BER/FER for the α corresponding to a 44-user and 43-user OR-MAC respectively.