

Nonlinear Turbo Codes For Higher-Order Modulations

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—

In 1982 Ungerboeck presented a set-partitioning design technique for trellis coded modulation (TCM). Although this technique directly assigns constellation points to the branches of the trellis, it has been shown that the same optimal code can be accomplished by a 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. This paper shows that for higher-order modulations using linear codes is too restrictive. Parallel Concatenated Nonlinear Trellis Coded Modulation (PC-NLTCM) which directly assigns constellation points as output-labels to the branches of the trellis can outperform PC-TCM. 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 code by around 0.2 dB. To facilitate analysis of the new codes, an extension of Benedetto's uniform interleaver analysis to handle nonlinear constituent codes is presented.

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 same optimal code can be accomplished 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 and Parallel Concatenated Trellis Coded Modulation (PC-TCM) has been traditionally designed using parallel concatenated convolutional codes with a bits-to-symbol mapper (See among others [2][3][4]).

However, this paper shows that in the case of TCM for higher-order modulations the use of parallel concatenated linear convolutional codes and a mapper is a restriction that may produce suboptimal codes. Using 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 previous published linear turbo code by around 0.2 dB in the waterfall region. Moreover, this improvement

in performance comes at no cost, as the decoding algorithm for linear or nonlinear constituent trellis codes is the same.

To facilitate analysis of the new codes, an extension of Benedetto's uniform interleaver analysis for nonlinear constituent codes is presented. A previous extension for nonlinear codes and bit-interleaving has been previously presented in [5]. This paper presents an extension of symbol-interleaving. It is shown that the same design criteria for linear turbo codes can be applied to nonlinear turbo codes. Namely, we define *effective free distance* for nonlinear codes, and show that this is an important metric to maximize when designing constituent codes for a PC-NLTCM.

This paper is organized as follows. Section I shows the structure of the parallel concatenated nonlinear trellis structure. Section II shows an extension of Benedetto's uniform interleaver analysis to bound the bit-error rate of parallel concatenated nonlinear codes. Section III shows a design example for a 2-bits/s/Hz 16-state PC-NLTCM with 8PSK and shows it outperforms previously proposed PC-TCM under the same conditions. Section IV delivers the conclusions.

I. PARALLEL CONCATENATED NONLINEAR CODES

The structure of the PC-NLTC encoder has been presented in [5] and is shown in Fig. 1. It is in essence the well-known turbo-code structure first proposed in [6] for systematic linear encoders, except that the output is assigned directly to each branch of the trellis by a look-up table rather than a linear function. A similar replacement of a linear operation by a look-up table has been successfully proposed for the decision feedback equalizer in presence for use in channels with trailing nonlinear inter-symbol interference [7]. The encoder consists of two constituent nonlinear trellis encoders (labeled NLTC) linked by an interleaver (labeled Π). Each trellis encoder uses k_0 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. Each output consists of n_0 constellation points, resulting in a total rate of $k_0/(2n_0)$ bits/symbol. Also, for the application considered in this work the trellis codes are non-systematic, and $k_0 > 1$, in which case symbol interleaving [4] will be used. We denote the input block length in bits as K , and the interleaver length in symbols as $k = K/k_0$. Also, we denote the symbol alphabet size as $q = 2^{k_0}$.

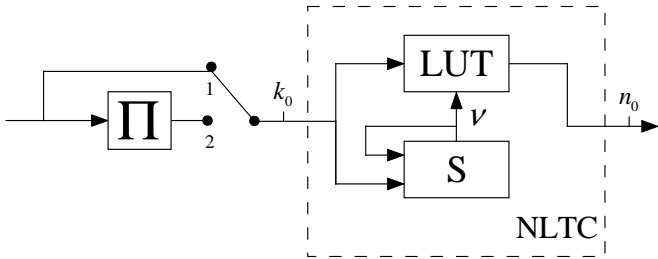


Fig. 1. PC-NLTC structure.

II. 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 [8]. 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 [8] for a parallel concatenation of nonlinear codes is required. In order to do that, a new probabilistic interleaver will be defined as interleaver, which produces similar equations to the linear case.

Since the codes used in this work are non-systematic, the BER 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.* $k_0 \geq 1$. In that case, a symbol-interleaver [4] is assumed, and the symbol error rate (SER), *i.e.* the average number of symbols of k_0 bits that are in error, will be computed. Note that for $k_0 = 1$ the symbol error rate is the bit error rate.

A. Uniform interleaver for nonlinear codes

In this section we extend the uniform interleaver bounding technique in [8] to nonlinear constituent codes and symbol-interleaving. 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 [8] to nonlinear constituent codes.

Definition 1: A *Uniform Interleaver* of length k (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 $k!$ possible permutations of the symbol positions with equal probability. Second, for each position, the value of the q -ary symbol can be changed to any other q -ary symbol with equal probability.

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 [8] would maintain the Hamming weight of both words and their Hamming distance, which would make the equations more complicated and would make it harder to draw conclusions from them (see [5]). With this new definition, any word can be mapped to any other word, no matter their Hamming weight. Thus, the only thing preserved after the interleaver is the symbol-wise Hamming distance between any two input pairs. Therefore, any pair of input words U_1 and \tilde{U}_1 such that $d_H(U, \tilde{U}) = i$, can be mapped by the uniform interleaver to any other pair of input words satisfying $d_H(U_2, \tilde{U}_2) = i$ with probability:

$$P\left(\left.(U_1, U_2\right) \rightarrow \left(\Pi(U_1), \Pi(U_2)\right) \middle| d_H(U_1, U_2) = d_H(\Pi(U_1), \Pi(U_2)) = i\right) = \frac{1}{q^{k \cdot (q-1)^i \cdot \binom{k}{i}}}. \quad (1)$$

Consider any two output codewords of length n , $X = \{x_1, \dots, x_n\}$ and $\tilde{X} = \{\tilde{x}_1, \dots, \tilde{x}_n\}$. For the AWGN case, the pairwise probability of error can be upper bounded by:

$$P_e(X \rightarrow \tilde{X}) = Q\left(\sqrt{d_E^2(X, \tilde{X}) \frac{E_s}{2N_0}}\right) \leq \frac{1}{2} e^{-\frac{E_s}{4N_0} d_E^2(X, \tilde{X})}, \quad (2)$$

where E_s/N_0 is the signal-to-noise ratio and $d_E^2(X, \tilde{X})$ is the squared Euclidean distance assuming unitary power transmission. Although this paper focuses on the AWGN channel the technique presented in this work is valid for any channel, provided an additive distance can be defined for that channel and the pair-wise error probability can be upper-bounded by:

$$P(X \rightarrow \tilde{X}) \leq \nu \lambda^{d(X, \tilde{X})} \quad (3)$$

where the directional distance metric $d(X, \tilde{X})$, and the parameters ν and λ depend on the channel. Note that (2) is equal to (3) when $d(\cdot) = d_E^2(\cdot)$, $\nu = 1/2$ and $\lambda = e^{-\frac{E_s}{4N_0}}$.

Define the *Input-Output Distance Enumerating Function* (IODEF) of a given (n, k) code C as

$$A^C(I, D) = \sum_{i,d} A_{i,d}^C I^i D^d, \quad (4)$$

where $A_{i,d}^C$ 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 $d(X, \tilde{X}) = d$. I and D are placeholders.

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

$$A_i^C(D) = \sum_d A_{i,d}^C D^d. \quad (5)$$

Inserting Eq. (5) in Eq. (4), the expression for the IODEF can be rewritten as:

$$A^C(I, D) = \sum_i A_i^C(D) I^i. \quad (6)$$

Denote as \mathcal{U} all the possible q^k input words. Then the SER can be upper bounded by the union bound:

$$SER \leq \frac{1}{k} \sum_{U_i \neq U_j \in \mathcal{U}} d_H(U_i, U_j) P(X_i \rightarrow X_j) \quad (7)$$

Using (3) then:

$$SER \leq \frac{\nu}{k} \sum_{U_i \neq U_j \in \mathcal{U}} d_H(U_i, U_j) \lambda^{d(X_i \rightarrow X_j)}. \quad (8)$$

Using the definition of IODEF in (5) then:

$$SER \leq \frac{\nu}{k} (1/q)^k \sum_{i,d} i A_{i,d}^C \lambda^d = \frac{\nu}{k} (1/q)^k \frac{\partial A^C(I, D)}{\partial I} \Bigg|_{D=\lambda, I=1} \quad (9)$$

Using (3) and (4) the symbol error rate (SER) or bit error rate in case $k_0 = 1$ can be upper bounded by:

$$SER \leq \frac{\nu}{k} \cdot (1/q)^k \frac{\partial A^C(I, D)}{\partial I} \Bigg|_{D=\lambda, I=1} \quad (10)$$

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 (in q -ary symbols), 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 for one operation of the interleaver, as explained in [8]). 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 (3)) as:

$$A_i^{C_P}(D) = \frac{A_i^{C_1}(D) \cdot A_i^{C_2}(D)}{q^k \cdot (q-1)^i \cdot \binom{k}{i}} \quad (11)$$

Notice that the uniform interleaver as defined in Sec. II-A can map any input word to any other input word. Now, plugging (11) into (10), it can be observed that there are two terms of the form $(1/q)^k$, 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 *Weighted Input-Output Distance Enumerating Function* (WIODEF) of a given (n, k) code C as

$$\tilde{A}^C(I, D) = \sum_{i,d} \tilde{A}_{i,d}^C I^i D^d, \quad (12)$$

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

$$SER \leq \frac{\nu}{k} \frac{\partial \tilde{A}^C(I, D)}{\partial I} \Bigg|_{D=\lambda, I=1} \quad (13)$$

Now, using (5) and (12):

$$\tilde{A}_i^{C_P}(D) = \frac{\tilde{A}_i^{C_1}(D) \cdot \tilde{A}_i^{C_2}(D)}{(q-1)^i \cdot \binom{k}{i}} \quad (14)$$

Note that except for the term $1/q^k$ in $\tilde{A}_{i,d}$, and the term $1/(q-1)^i$ the equations (12)-(14) for a parallel concatenation of nonlinear codes are the same as for the linear case [8]. As it turns out, all the conclusions and design criteria derived in [8] apply to nonlinear constituent codes. See Appendix I 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, an important parameter to maximize is the effective free distance defined as:

Definition 2: *Effective free distance* of a constituent code is the minimum distance in the output between any two possible input words U and \hat{U} with input Hamming distance $d_H(U, \hat{U}) = 2$.

III. NONLINEAR TURBO CODES FOR HIGHER-ORDER MODULATIONS OVER AWGN

In this section we will show that directly assigning constellation points to the trellis branches of each constituent code can potentially 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 $K = 10000$ bits, and therefore $k = 2500$ symbols with $k_0 = 4$. Each output branch of each constituent encoder consists of one 8PSK constellation point, which produces a code rate of 2 bits/s/Hz. This is a valid 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. This code 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 $k_0 = 4$, 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 states with each of the 16 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

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 OUTPUT LABEL IS THE SAME FOR BOTH ENDING STATES.

	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

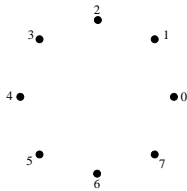


Fig. 2. Labeling for 8PSK.

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 II shows the output 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 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. This search can be constrained to recursive trellis structure of the form:

$$S = A \cdot S + B \cdot u \pmod{2}, \quad (15)$$

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:

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (16)$$

This trellis structure together with the output labeling of Table II 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 $K = 10000$ bits, or $k = 2500$ 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. 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 15 possible values (leaving out the correct symbol), and therefore we used a correction factor on the error bound of

$$BER^{\text{bound}} \approx \frac{k_0 \cdot 2^{k_0-1}}{2^{k_0} - 1} SER^{\text{bound}}. \quad (17)$$

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 design stage, it gives a good prediction of which 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.

IV. CONCLUSIONS

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 sub-optimal 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.

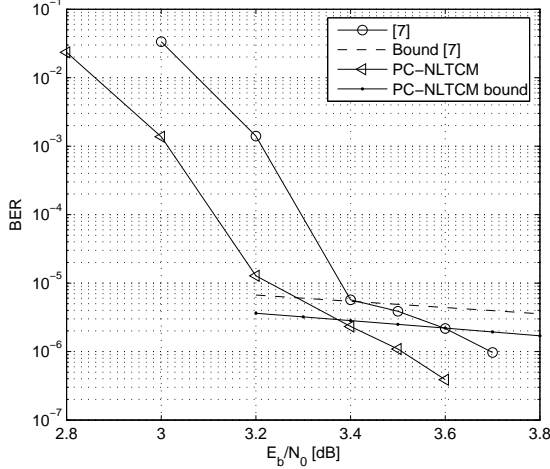


Fig. 3. BER vs. E_b/N_0 comparison, for 2 bits/s/Hz 16-state parallel concatenated codes with 8PSK.

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 define effective free distance for nonlinear constituent codes, and conclude that this is an important parameter to maximize at the design stage.

APPENDIX

I. COMPUTING THE SER BOUND FOR CONSTITUENT NONLINEAR TRELLIS CODES

Biglieri et al. presented a union bound in [9][10] for general trellis codes, using a $2^{2\nu}$ -state trellis diagram. This concept can be used to find $A^{CP}(I, D)$ for the case of parallel concatenated nonlinear trellis codes.

As in [9], 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

$$S_G = \{(s_e, s_r) \mid s_e = s_r\}, \quad S_B = \{(s_e, s_r) \mid s_e \neq s_r\}. \quad (18)$$

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

$$S(I, D) = \begin{bmatrix} S_{GG}(I, D) & S_{GB}(I, D) \\ S_{BG}(I, D) & S_{BB}(I, D) \end{bmatrix}, \quad (19)$$

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:

$$(1/q)I^{d_H(u_e, u_r)}D^{d(x_e, x_r)}, \quad (20)$$

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 $q = 2^k$ possible inputs per trellis branch, $(1/q)$ is 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 in terms of number of operations. In order to reduce complexity, two approximations can be made: (1) Use the same idea presented in [8]: 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 $N = 2^\nu$ states with equal probability.

Define the approximated single-error event function as:

$$E(I, D) = p_s \{S_{GB}(I - S_{BB})^{-1}S_{BG}\mathbf{1}, \quad (21)$$

where $p_s = [\frac{1}{N} \frac{1}{N} \cdots \frac{1}{N}]$ is the probability distribution of the encoder states and $\mathbf{1} = [\mathbf{1} \cdots \mathbf{1}]^T$. This placeholder counts the length, in trellis sections, of an error event. Then, $E(I, D)$ can be written as:

$$E(I, D) = \sum_{i,d} e_{i,d} D^d I^i \quad (22)$$

Now define:

$$E_j(I, D) = [E(I, D)]^j = \sum_{i,d} e_{i,d,j} I^i D^d, \quad (23)$$

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

$$K[l, j] \leq \binom{k-l+j}{j} \approx \frac{k^j}{j!}, \quad (24)$$

ways in the trellis. Note that the two terms in (24) 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 large the upper bound becomes very tight. Also, the symbols of the rest of the $k-l$ positions of both input words are equal and could be almost any of the possible q^{k-l} combinations, which divided by the term q^k 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 (20)). Therefore, for each constituent code,

$$\tilde{A}^C \approx \sum_j \frac{k^j}{j!} E_j(I, D). \quad (25)$$

Using (13), (14) and (25), and using the fact that $\binom{k}{i} \approx k^i/i!$ for k sufficiently large and i small, we get:

$$SER \approx \sum_{i,j_1,j_2,d_1,d_2} \nu \frac{i!}{j_1!j_2!} \frac{k^{(j_1+j_2-i-1)}}{(q-1)^i} e_{i,d_1,j_1}^{C_1} e_{i,d_2,j_2}^{C_1} \lambda^{d_1+d_2}. \quad (26)$$

Therefore, as k 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. II-B. As for linear encoders, feed-forward encoders lead to really bad performance since i can be equal to 1 in which case $j_1 + j_2 - i - 1 = 0$.

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] Christina Fragouli and Richard Wesel. Turbo-encoder design for symbol-interleaved parallel concatenated trellis-coded modulation. *IEEE Trans. on Info. Theory*, 47:873–890, Mar 2001.
- [5] 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.
- [6] C. Berrou and A. Glavieux and P. Thitimajshima. Near shannon limit error-correcting coding and decoding: turbo-codes. pages 873–890, May 1993.
- [7] K.D. Fisher, J.M. Cioffi, W.L. Abbott, P.S. Bednarz, and C.M. Melas. An adaptive RAM-DFE for storage channels. *IEEE Transactions on Communications*, 39:1559–1568, November 1991.
- [8] S. Benedetto and G. Montorsi. Unveiling turbo codes: Some results on parallel concatenated coding schemes. 42, Mar 1996.
- [9] E. Biglieri. High-Level Modulation and Coding for Nonlinear Satellite Channels. 32, May 1984.
- [10] Y. J. Liu, I. Oka, and E. Biglieri. Error probability for digital transmission over non-linear channels with application to TCM. 36, Sep 1990.