Symbol Interleaved Parallel Concatenated Trellis Coded Modulation

Christine Fragouli and Richard D. Wesel
Electrical Engineering Department
University of California, Los Angeles
christin@ee.ucla.edu, wesel@ee.ucla.edu

Abstract—**This paper presents a method for efficient coding at high spectral efficiency using parallel concatenated trellis coded modulation (PCTCM) with symbol interleaving. The constituent encoders are optimized for symbol-wise free distance, and each has an infinite symbol-wise impulse response. In many cases of practical interest, the optimal structure for these constituent encoders connects the memory elements in a single row. Simulation results show that performance is as close as 0.5 dB to constrained capacity.**

I. Introduction

This paper presents a method for parallel concatenated trellis coded modulation (PCTCM) with constituent encoders of rate k/n, $k > 1$. The k binary inputs can be thought of as one symbol input over the extension field $GF(2^k)$. This approach uses one symbol interleaver between the constituent encoders instead of k bit interleavers.

The use of a symbol interleaver implies that the constituent encoders should be optimized for "symbol effective free distance." This term refers to the minimum output distance when the input symbol sequence has exactly two symbols different from zero, as opposed to the usual notion of effective free distance which refers to the minimum output distance for a binary input Hamming distance of two.

Section II presents previous work, and determines where our approach stands. Section III motivates the use of symbol interleaving. Section IV illustrates our proposed method for symbol interleaved PCTCM. Section V derives guidelines for the design of the constituent encoders, extends the effective distance bounds to symbol-wise inputs, and discusses the appropriate encoder structures for constituent encoders. Tables of codes that achieve the upper bound of effective free distance are included. Section VI presents simulation results. Section VII compares bit and symbol interleaving, and Section VIII concludes the paper.

II. Previous Work

For high spectral efficiency, two main approaches are proposed in the literature for the turbo encoder structure, one employing bit interleaving [1], and the other employing symbol interleaving [2], [3].

For bit interleaving, k bit interleavers are used to keep the bit streams separate. The method in [1], for $k/(k+1)$ (k even), uses the $k/2$ systematic outputs and the parity bit of the first encoder and does not transmit the remaining $k/2$ systematic bits. The second constituent encoder is the same as the first, but the roles of the two groups of $k/2$ input bits are reversed. Thus the overall turbo encoder is systematic.

The symbol interleaver in [2], [3], maps even symbol positions to even symbol positions and odd ones to odd. This is equivalent to using two separate symbol interleavers of half the length, one for the odd positions and another for the even ones. The output of the second encoder is de-interleaved and the output symbols from each encoder are punctured alternatively. Again, the overall encoder is systematic.

The following section provides motivation for preferring symbol interleaving to bit interleaving. However, in the approach of [2], the additional structure of the symbol interleaver reduces the interleaving gain, and puncturing complicates the design of the constituent encoders. Therefore in Section IV our proposed approach combines the encoder structure of [1], with a symbol interleaver.

III. Motivation for Using Symbol Interleaving

Parallel turbo code performance is based on the coupling of recursive constituent convolutional encoders with a large random interleaver of length N. The use of this interleaver makes the complexity of the maximum likelihood decoder too large, so a suboptimal iterative decoding structure is used instead.

The iterative decoding structure consists of two separate decoding blocks, each implementing the forward-backward algorithm for the respective constituent encoder. The blocks exchange probabilities for the input symbols.

Consider why this procedure doesn't give the correct result $P(\mathbf{u}, \mathbf{Y_2}|\mathbf{Y_1})$ after one iteration. The problem lies in the assumption that the exchanged input symbol probabilities are independent. This is not true because they are conditioned on the observed output sequence:

$$P(\mathbf{u}|\mathbf{Y_1}) \neq \Pi_t P(\mathbf{u_t}|\mathbf{Y_1})$$

This work was supported by NSF CAREER award #9733089, Conexant Systems Inc., and the Xetron Corporation.

Using bit interleaving leads to the additional assumption, that not only every input symbol is independent from every other input symbol, but also the bits within each symbol are also independent. Again this is not true:

$$P(\mathbf{u_t}|\mathbf{Y_1}) \neq \Pi_i P(u_{t,i}|\mathbf{Y_1})$$

Symbol interleaving avoids this additional assumption of independence.

IV. Proposed Parallel Concatenated Trellis Coded Modulation (PCTCM) Structure

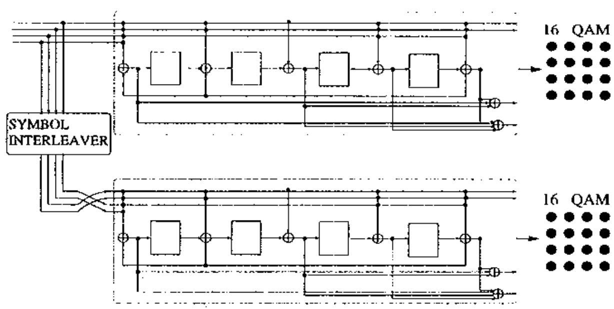

Fig. 1. 2 bits/sec/Hz PCTCM Turbo Code With Rate 4/4 Constituent Encoders

An example of the proposed parallel turbo code structure, that employs 16 QAM modulation in connection with rate 4/4 constituent encoders, is depicted in Fig. 1. The generalization to k/n encoders using 2^n-point constellations is straightforward when k is even.

The upper constituent encoder has as systematic outputs the $k/2$ MSB input bits, while the lower constituent encoder has as systematic outputs the $k/2$ LSB input bits. Thus the systematic bits are evenly divided between the constituent encoders, appear only once at the turbo code output, and there is no need for puncturing or interleaver constraints as in [2].

Our iterative decoder implements the Soft Input Soft Output (SISO) equations appearing in [4], with input bit probabilities substituted by input symbol probabilities.

V. Constituent Encoder Design

In the rest of this paper, we use several variations of effective free distance. The superscript refers to the output distance, Hamming (H) or Euclidean (E) and the subscript to the input weight, bit-wise (b) or symbol-wise (s). For example, d_{s2}^E stands for the output Euclidean distance when the symbol-wise input weight is two.

A. Desired Distance Properties

An analytical upper bound to the bit error probability of turbo codes in [5] identified effective free distance as a key parameter. A similar analysis still holds when the input of the constituent encoders is over GF(2^k), with the slight modification that the input Hamming weight now refers to Hamming weight in the extension Galois field $GF(2^k)$.

Repeating the analysis for symbol-wise input along the lines of [5], two main guidelines for the design of constituent encoders can be derived:

- For a given symbol interleaver length, to achieve interleaver gain, the constituent convolutional encoders must have infinite output weight when the input symbol sequence contains only one symbol different than zero ($d_{s1}^H = \infty$).
- Among the encoders with $d_{s1}^H = \infty$, the ones with the best symbol effective distance profile (d_{s2}^H or d_{s2}^E depending on the application) optimize the turbo code performance.

The first guideline equivalently states that there should be no parallel transitions in the trellis diagram, and was also presented in [2].

B. Distance Upper Bounds

The following theorem, presented in [6] and [7], gives an upper bound on the effective free distance d_{b2}^H.

Theorem 1 (Divsalar and McEliece) *Consider the convolutional codes with k inputs, m memory elements, and r parity (not systematic) outputs. Assume that $d_{b1}^H = \infty$, i.e. the impulse response of every one of the k binary inputs is infinite. Then the highest effective free distance achievable is bounded by:*

$$d_{b2}^H \leq \min\left(\left\lceil \frac{2^m}{k} \right\rceil r, 2r + \left\lfloor \frac{2^{m-1}r}{k} \right\rfloor\right) \qquad (1)$$

This theorem refers to encoders with k binary inputs. For symbol interleaved PCTCM, it is interesting to examine the d_{s2}^H bound:

Theorem 2 *An upper bound to the d_{s2}^H, when $d_{s1}^H = \infty$, with r parity (not systematic) outputs and k binary inputs, is given by substituting k with $2^k - 1$ in Theorem 1:*

$$d_{s2}^H \leq \min\left(\left\lceil \frac{2^m}{2^k - 1} \right\rceil r, 2r + \left\lfloor \frac{2^{m-1}r}{2^k - 1} \right\rfloor\right) \qquad (2)$$

Proof

The proof goes along the same lines as the proof of Theorem 1 in [6]. The main point is the following:

If the feedback polynomial of a convolutional encoder is primitive, then the state diagram has one loop with zero inputs, and not-zero outputs. This loop includes all the $2^m - 1$ states except the all-zero one. A two-input sequence causes the encoder to enter the loop (with the first non zero input) and exit it (with the second non zero input). The output weight of any parity output going around the whole loop is 2^{m-1}. If k binary inputs exist, there are k ways to enter and leave the loop via single input bits, and thus the output weight of the parity output, along the part of the loop that it travels before it exits, can be in the best case, $2^{m-1}/k$. Considering symbol inputs, there are instead $2^k - 1$ ways to join/exit this loop, and thus the output weight can be, in the best case, $2^{m-1}/(2^k - 1)$. Similar arguments apply when the feedback polynomial is not primitive. □

C. Range of Encoders to Search

Up to now, searching for good trellis codes, required examining one code within each group of range equivalent encoders (i.e. equivalent in Forney's sense). So it was sufficient to restrict attention within a set of canonical encoders.

For turbo codes the mapping from input to output sequences plays an important role. The only encoders that may be disregarded are those that are strictly equivalent [8] to codes that have been examined. The natural question is what can we say about canonical encoders now.

The answer is provided by the Rational Form Theorem [9] which states that for any m-memory element convolutional encoder, a strictly equivalent encoder exists of the ordinary memory structure with R rows. The following theorem helps to decide what values of R should be considered.

Theorem 3 *For all (k, m, r) values such that*

$$k < \min\left(2^{m-1} - 2, r(2^{m-2} - 1)\right) \tag{3}$$

the bound of Theorem 1 can only be achieved if the m memory elements are connected in a single row.

Proof

It suffices to show that the use of multiple rows of memories enforces an upper bound on the effective free distance lower than the bound in Theorem 1.

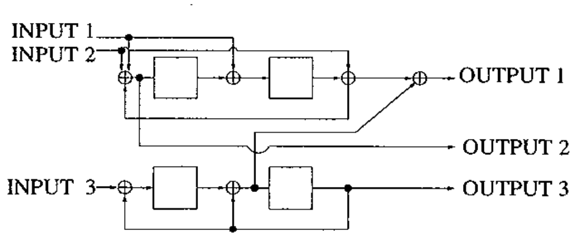

Fig. 2. Encoder structure with two memory chains

Assume that the m memories are connected in R rows with m_j memories in row j, $j = 1\ldots R$, and $\sum_{j=1}^{R} m_i = m$. Let k_j be the number of inputs in row j, $k_j \leq k$, $\sum_{j=1}^{R} k_j \geq k$, and r_j be the number of outputs from row j, $r_j \leq r$. In the example of Fig. 2, $R = 2$, $m = 4$, $m_1 = m_2 = 2$, $k = 3$, $k_1 = 2$, $k_2 = 1$, $r = 3$ and $r_1 = r_2 = 2$.

Think of an input-weight-two sequence, as the superposition of two (infinite in this case) impulse responses. Restricting attention to impulse responses that are both from the same memory chain j, d_{b2}^H is bounded by:

$$d_{b2}^H \leq \min\left(\left\lceil \frac{2^{m_j}}{k_j} \right\rceil r_j, 2r_j + \left\lfloor \frac{2^{m_j-1} r_j}{k_j} \right\rfloor\right) \tag{4}$$

So for the total encoder it holds that:

$$\begin{aligned} d_{b2}^H &\leq \min_{j=1\ldots R}\left(\min\left(\left\lceil \frac{2^{m_j}}{k_j} \right\rceil r_j, 2r_j + \left\lfloor \frac{2^{m_j-1} r_j}{k_j} \right\rfloor\right)\right) \\ &\leq \min_{j=1\ldots R}\left(\min\left(\left\lceil \frac{2^{m_j}}{k_j} \right\rceil r, 2r + \left\lfloor \frac{2^{m_j-1} r}{k_j} \right\rfloor\right)\right) \end{aligned} \tag{5}$$

The min function in Theorem 1 is equal to the first term for $m < m^*$, and to the second term for $m \geq m^*$, where m^* is determined by r and k. Thus:

1. For $m \geq m^*$:

If the m memories are connected in one row:

$$d_{b2}^H \leq 2r + \left\lfloor \frac{2^{m-1} r}{k} \right\rfloor \tag{6}$$

If the m memories are connected in R rows:

$$\begin{aligned} d_{b2}^H &\leq \min_{j=1\ldots R}\left(2r + \left\lfloor \frac{2^{m_j-1} r}{k_j} \right\rfloor\right) \\ &\leq 2r + \left\lfloor \frac{\sum_{j=1}^{R} 2^{m_j-1} r}{k} \right\rfloor \end{aligned} \tag{7}$$

Note that perhaps $m_j \leq m^*$, but (7) is still an upper bound.

2. Similarly, for $m < m^*$:

If the m memories are connected in one row:

$$d_{b2}^H \leq \left\lceil \frac{2^m}{k} \right\rceil r \tag{8}$$

If the m memories are connected in R rows:

$$d_{b2}^H \leq \left\lceil \frac{\sum_{j=1}^{R} 2^{m_j}}{k} \right\rceil r \tag{9}$$

If it holds that $k < 2^m - \sum_{j=1}^{R} 2^{m_j}$ for all possible R and m_j partitions, then equations (8) and (9), are related by a strict inequality. But:

$$\sum_{j=1}^{R} 2^{m_j} \leq \sum_{j=1}^{2} 2^{m_j} \leq 2^{m-1} + 2 \tag{10}$$

for $0 < m_j < m$ and m_j integer, so it suffices that $k < 2^{m-1} - 2$. Similarly, for strict inequality between equations (6) and (7), it suffices that $k < r(2^{m-2} - 1)$. □

Thus in an exhaustive search, if the memory elements connected in a single row give a code with d_{b2}^H higher than the maximum achievable d_{b2}^H with multiple memory rows, there is no need to expand the search to multiple rows. This is very often the case in practice.

Table I provides codes with k inputs, r parity outputs and m memory elements, optimized for d_{b2}^H, and identified through exhaustive search. The generator polynomials in octal notation describe in turn the feedback, the k input connections, and the r parity output connections. For example, the upper constituent encoder in Fig. 1 has generator polynomials: $\{23, 1, 31, 33, 37, 25, 35\}$. Also, d_{b2}^u denotes the upper d_{b2}^H limit for a single memory row, and d_{b2}^m denotes the maximum achievable d_{b2}^H for multiple rows. The codes in Table I can be made systematic by adding $n - r$ systematic outputs. So although their free distance, denoted by d^*, might be zero, the free distance of the complete code will be positive because of the systematic outputs. Codes listed in italics have repeated outputs and do not perform well in simulations [7].

k	r	m	Generator polys	d_{b2}^m	d_{b2}^u	d_{b2}^H	d_{b3}^H	d^*
2	1	3	{11,3,15,12}	*2*	**4**	4	∞	1
2	1	6	{147,1,127,111}	*6*	**18**	18	5	0
2	2	3	{13,1,3,15,17}	*4*	**8**	7	2	1
2	2	3	*{11,3,15,12,12}*	*4*	**8**	8	2	1
2	2	4	{35,1,27,37,33}	*8*	**12**	12	7	0
2	3	3	{15,1,7,17,14,13}	*6*	**12**	10	5	2
2	3	3	*{11,3,15,12,12,12}*	*6*	**12**	12	5	2
3	1	4	{25,1,17,23,27}	*2*	**4**	4	2	0

TABLE I

CODES OPTIMIZED FOR d_{b2}^H.

VI. SIMULATION RESULTS

This section provides examples of the proposed method, for 2 bits/sec/Hz employing 16 QAM and 8 PSK. The interleaver in the simulations is the best uniform interleaver we found after a few trials.

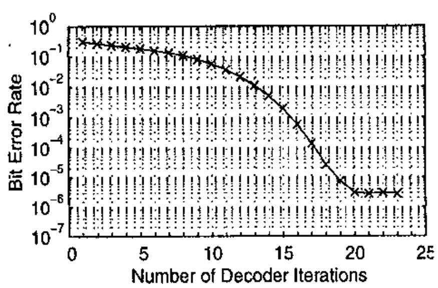

Fig. 3. 2 bits/sec/Hz/ turbo code, at $E_b/N_o = 2.6$ dB, interleaver length 30,000 symbols and 16 QAM.

2 bits/sec/Hz PCTCM with 16 QAM

The constituents encoders implement 4/4 codes with 2 parity and 2 systematic outputs, and have 4 memory elements. Table II contains in octal notation such codes identified through exhaustive search, and optimized in d_{s2}^E for the 16 QAM constellation labeling (labels in hex and in raster order): { 3, 1, 5, 7, 2, 0, 4, 6, A, 8, C, E, B, 9, D, F}. The simulated code is in the first row (nn denotes the number of nearest neighbors). The whole turbo encoder is depicted in Fig. 1. The exhaustive search identified a much larger number of good codes, which can be provided by the authors to the interested reader.

h_0	h_1	h_2	h_3	h_4	h_5	h_6	d_{s2}^E	d_{s3}^E	d_s^*
23	**1**	**31**	**33**	**37**	**25**	**35**	3(4*nn*)	2(2*nn*)	2
23	1	31	33	34	11	37	3(8*nn*)	1(1*nn*)	1
31	1	11	16	37	32	36	3(3*nn*)	1(1*nn*)	1
31	1	11	15	17	27	35	3(3*nn*)	1(1*nn*)	1

TABLE II

CODES OPTIMIZED FOR d_{s2}^E FOR 16 QAM.

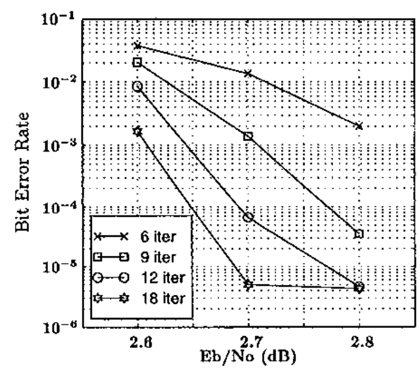

Fig. 4. 2 bits/sec/Hz/ turbo code, with interleaver length 16,000 symbols and 16 QAM. Constrained capacity=2.1 dB.

For interleaver length 30,000 symbols, the performance (Fig. 3) is within 0.85 dB from unconstrained AWGN capacity, and within 0.5 dB from the constrained capacity of 16 QAM over AWGN channel.

For interleaver length 16,000 symbols (Fig. 4), the performance is within 0.6 dB from the constrained capacity.

2 bits/sec/Hz PCTCM with 8 PSK

The constituents encoders implement a 4/3 code with 1 parity and 2 systematic outputs and have 4 memory elements. Table III presents in octal notation codes optimized for d_{s2}^E for the 8 PSK labeling (clock-wise): {7,6,5,4,3,2,1,0}. The simulated code is in the first row. For interleaver length 5,000 symbols (Fig. 5), the performance is within 0.6 dB from the constrained capacity.

VII. COMPARISON OF SYMBOL AND BIT INTERLEAVING

According to our simulation results, symbol interleaving converges at a lower SNR, but appears to have a higher error floor than bit interleaving.

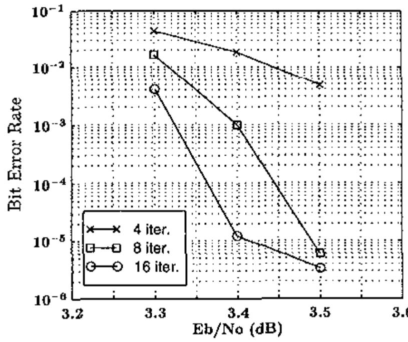

Fig. 5. 2 bits/sec/Hz/ turbo code, with interleaver length 5,000 symbols and 8 PSK. Constrained capacity=2.8 dB.

h_0	h_1	h_2	h_3	h_4	h_5	d^E_{s2}	d^E_{s3}
27	**1**	**6**	**10**	**14**	**23**	1.171573(3nn)	0.585786(5nn)
35	5	7	10	13	22	1.171573(3nn)	0.585786(5nn)
27	13	24	26	31	30	1.171573(3nn)	0.585786(5nn)

TABLE III

CODES OPTIMIZED FOR d^E_{s2} FOR 8 PSK

Convergence at lower SNR may be attributed to the fact that symbol interleaving imposes less assumptions on the iterative decoding (Section III).

The higher error floor indicates that a symbol interleaved turbo encoder has lower free distance than a bit interleaved one. Indeed, interleaving k bits instead of every one symbol allows spreading of the components of one error event to k-times more error events, typically accumulating more distance.

Another way to describe the effect of a symbol interleaver is that using a symbol interleaver is equivalent to using k bit interleavers that implement the same interleaving pattern. This additional structure reduces the interleaver gain.

Moreover, there is less symbol-wise effective free distance available than bit-wise (Theorem 2). Fig. 6 shows bounds on b^H_{b2} and d^H_{s2} for $k = 4$ and $r = 2$.

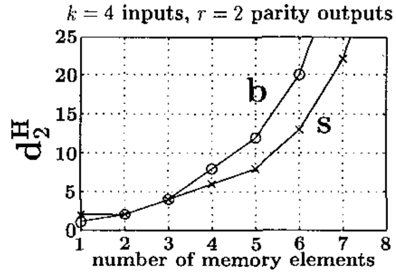

Fig. 6. Comparison of upper bounds on effective distance $\mathbf{d}^{\mathbf{H}}_{\mathbf{2}}$, for b=bit-interleaving, and s=symbol-interleaving.

Bit interleaving may require a smaller number of decoder iterations, because the bit interleaver helps the information to propagate faster between the encoders.

Bit interleaving also requires less memory at the decoder, since $k - 1$ bit reliabilities have to be kept instead of $2^k - 1$ symbol reliabilities.

VIII. CONCLUSIONS

This paper presented a method to achieve high spectral efficiency using symbol interleaved parallel concatenated trellis coded modulation.

The constituent encoder design determined the optimization criteria and extended the effective distance bound to symbol-wise inputs.

Rational form theorem shows that in order to examine all strictly equivalent encoders, it is sufficient to consider only the canonical memory structures with R rows. In many cases, only encoders with the memory elements connected in a single row ($R = 1$), can achieve the maximum output effective distance.

Comparing symbol against bit interleaving, we noted that symbol interleaving imposes less restrictions on the iterative decoding, but on the other hand bit interleaving has a turbo encoder with more effective distance and interleaving gain available. Also bit interleaving requires less decoder iterations and less memory at the decoder.

Simulations results for 2 bits/sec/Hz with 16 QAM and 8 PSK, show that symbol interleaving converges at lower SNR but with a higher error floor as compared to bit interleaving.

REFERENCES

[1] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara. Parallel Concatenated Trellis Coded Modulation. *IEEE International Conference on Communications*, June 1996.

[2] P. Robertson and T. Worz. A Novel Bandwidth Efficient Coding Scheme Employing Turbo Codes. *IEEE International Conference on Communications*, 1995.

[3] P. Robertson and T. Worz. Bandwidth-Efficient Turbo Trellis-Coded Modulation Using Punctured Component Codes. *IEEE Journal in Selected Areas in Communications*, 16(2):206–218, February 1998.

[4] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara. A Soft-Input Soft-Output APP Module for the Iterative Decoding of Concatenated Codes. *IEEE Communications Letters*, 1(1), January 1997.

[5] S.Benedetto and G. Montorsi. Unveiling turbo codes:some results on parallel concatenated coding schemes. *IEEE Transactions on Info. Theory*, 42:409–429, March 1996.

[6] D. Divsalar and F. Pollara. On the Design of Turbo Codes . *TDA Progress Report*, 42:99–121, November 1995.

[7] D. Divsalar and R.J. McEliece. Effective free distance of turbo codes . *Electronics Letters*, 32(5):445–446, February 1996.

[8] R. Wesel. Trellis Code Design For Correlated Fading And Achievable Rates For Tomlinson-Harashima Precoding. *PhD Dissertation*, August 1996.

[9] R. A. Horn and C. R. Johnson. Matrix Analysis. *Cambridge University Press*, 1985.