Semi-Random Interleaver Design Criteria

Christine Fragouli and Richard D. Wesel *
Electrical Engineering Department
University of California, Los Angeles

christin@ee.ucla.edu, wesel@ee.ucla.edu

Abstract

A spread interleaver [1] of length N is a semi-random interleaver based on the random selection without replacement of N integers from 1 to N under a design constraint. This paper extends the spread-interleaver design method to multiple error events, based on the interleaver's role in overall error event distances. The extension helps to explain why the spread interleaver is specifically designed to be semi-random. Simulation results show the performance achieved for a symbol-interleaved parallel concatenated trellis coded modulation (PCTCM) scheme [2].

1 Introduction

Interleaver design for turbo codes is mainly targeted towards lowering the error floor, which is the flattening of the error-rate curve that turbo codes exhibit for moderate and high values of SNR. A good interleaver design may also lead to earlier decoder convergence.

In this scope, Perez et al. [3] have shown that the turbo code asymptotic performance approaches the free-distance asymptote. The error floor observed with turbo codes is due to their relatively small free distance and consequently relatively flat free-distance asymptote. Thus to lower the error floor the free-distance of the turbo code must be maximized for a fixed interleaver length. Based on this observation various methods (for example [4, 5]) have been proposed to maximize the turbo code output distance by designing the interleaver tailored to specific constituent codes, and trying to break turbo code error events of low output weight. Some methods (for example [6, 7]) also propose interleaver design through cost function minimization.

This paper starts with the spread-interleaver introduced by Divsalar and Pollara and extends the design criteria to multiple error events based on the interleaver's role in the overall error event distances. This extension provides an understanding for the reason the spread interleaver is specifically designed to be semi-random as is explained in detail in following sections.

The paper is organized as follows: Section 2 describes the interleaver's role in the overall turbo code error events. Section 3 reviews the spread-interleaver, and extends the design criteria to multiple error-events. Section 4 provides motivation for the extended design criteria. Section 5 presents simulation results and finally Section 6 concludes the paper.

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

2 Review of Interleaver Role

Consider a parallel concatenated turbo code as in Fig. 1. The interleaver takes the input block of the upper encoder and produces the input block for the lower encoder. During decoding, an error event at the output

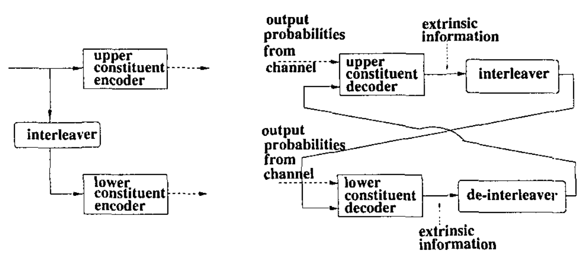


Figure 1: Parallel Concatenated Turbo Code.

of the upper decoder will be interleaved and spread to different error events of the lower decoder. The interleaver determines which error events exchange a-priori information during decoding. For simplicity from now on we refer to the interleaver input as input to the upper constituent encoder and to the interleaver output as input to the lower constituent encoder.

The role of the interleaver is to interconnect the error

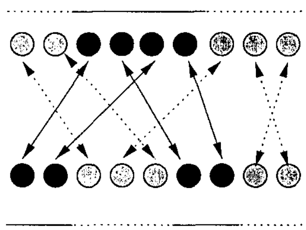


Figure 2: Two Sets Of Interrelated Error Events

events of the constituent encoders in such a way that the total output weight of a turbo encoder codeword accumulates distance from as many distinct error events as possible from each constituent encoder.A specific interleaving pattern leads to the partition of the upper and lower encoder inputs into sets of interrelated error events. Fig. 2 illustrates two such sets of interrelated error events.

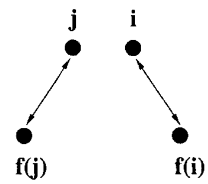
The output weight associated with a turbo code error event is the sum of the output weights of all the constituent error events belonging in the same set. The bigger a set is, the more weight it accumulates. A single constituent error event in the upper encoder with many input errors will induce many error events inthe lower encoder. Thus the overall turbo code error-event will have large distance even if the upper encoder error event by itself has very small output distance.

This observation implies that the constituent events with a small number of inputs lead to the lowest output distance. Indeed a commonly used example is that when a constituent encoder has error events of input weight one, these error events unavoidably map to input-weight-one error events in the second constituent encoder. This lead to very small total output weight, unless the constituent encoders have infinite impulse responses.

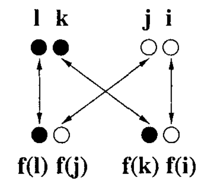
3 Design Criteria

This section presents design criteria for constructing semi-random interleavers. An interleaver of length N is completely described by a mutually exclusive and collectively exhaustive listing of the integers from 1 to N. Define $f(i)$ to be the integer in the i^{th} position in the list. The input symbol in position i before interleaving is in position $f(i)$ after interleaving.

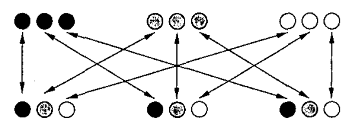
Fig. 3(a) depicts a way to have small number of interconnected error events. The component symbols of one error event in the upper encoder become part of the same error event in the second encoder. This case can be avoided by using the spread interleaver introduced by Divsalar and Pollara. The spread interleaver is described in [1] as a semi-random interleaver based on the random selection without replacement of N integers from 1 to N, where N is the interleaver length under the following constraint.



(a) Case for S_1, S_2.



(b) Case for T_1,T_2.



(c) Another case for T_1, T_2.

Figure 3: Some examples of interrelated error events.

Constraint 1 *The i^{th} randomly selected integer $f(i)$ must be rejected if there exists j such that:*

$$0 < i - j \leq S_1, \qquad |f(i) - f(j)| \leq S_2 \qquad (1)$$

This constraint guarantees that if two symbols i, j are within distance S_1 in the upper constituent encoder, they cannot be mapped to distance less than S_2 in the lower constituent encoder.

An extension of this procedure is to consider multiple error events in the upper encoder. As an example Fig. 3(b) depicts two error events of the upper encoder that interchange their component symbols. Weight accumulation stops in two steps. To avoid this situation we define two more parameters T_1 and T_2 and impose on the construction of the spread interleaver an additional constraint. Again randomly select without replacement integers from 1 to N, and if the i^{th} selection $f(i)$ satisfies the Constraint 1 described previously, check if the following condition is also satisfied.

Constraint 2 *The i^{th} randomly selected integer $f(i)$ must be rejected if there exist j, k, l such that:*

$$\begin{array}{ll} 0 < i - j \leq T_1, & |f(i) - f(k)| \leq T_2, \\ 0 < |k - l| \leq T_1 & |f(j) - f(l)| \leq T_2 \end{array} \qquad (2)$$

for which $j, k, l < i$.

This constraint guarantees that two relatively close component symbols i and j in the upper encoder, do not have $f(k)$ near $f(i)$ and $f(l)$ near $f(j)$ in the lower encoder, with k and l near each other in the upper encoder. Figs. 3(b) and 3(c) illustrate error events that are avoided.

This procedure can be extended to three error events in the upper encoder. Define parameters X_1 and X_2 and impose on the semi-random interleaver the following additional condition to satisfy.

Constraint 3 *The i^{th} randomly selected integer $f(i)$ must be rejected if there exist j, k, l, m and n such that:*

$$\begin{aligned} 0 < i - j \le X_1, &\qquad |f(i) - f(k)| \le X_2, \\ 0 < |k - l| \le X_1 &\qquad |f(j) - f(m)| \le X_2 \\ 0 < |m - n| \le X_1 &\qquad |f(n) - f(l)| \le X_2 \end{aligned} \tag{3}$$

where $j, k, l, m, n < i$.

Fig. 4 illustrates an example of an avoided error event. Extension to more than three error events is conceptually straightforward but leads to increased complexity.

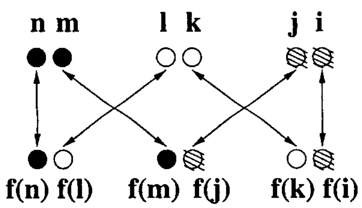


Figure 4: Case for Parameters X_1, X_2.

4 Motivation

To motivate the introduction of Constraints 2 and 3 consider the following example for a symbol interleaved system with constituent encoders of rate 4/3 employing 8-PSK [2].

The element (i,j) of Table 1 is the minimum squared Euclidean distance associated with a constituent encoder error event of symbol-wise length j, $j = 2 \ldots 17$, and input symbol-wise Hamming distance i, $i = 2 \ldots 4$. If we have an interleaver that satisfies Constraint 1 with $(S_1, S_2) = (10, 10)$, the minimum squared Euclidean distance error event of type Fig. 3(a) that may happen is 5.27 (let the upper error event have length 11 and the lower 2 and both of them input weight

Table 1: Squared Euclidean Distance for Error Events. The columns refer to the error event's length, and the rows to the input symbol-wise Hamming weight.

	2	3	4	5	6	7	8	9
2	1.17	1.17	1.76	2.34	2.34	2.34	2.34	2.93
3	–	0.59	0.59	0.59	1.17	1.17	1.75	1.75
4	–	–	0.59	0.59	0.59	0.59	0.59	0.59

	10	11	12	13	14	15	16	17
2	2.93	4.10	4.10	4.10	4.10	4.69	4.69	5.87
3	2.34	2.34	2.34	2.92	2.92	2.92	2.92	3.51
4	1.17	1.17	1.76	1.76	1.76	2.34	2.34	2.93

2,$\Rightarrow$ $1.17 + 4.1 = 5.27$). For the case depicted in Fig. 3(b) though, if the constituent error events have length 2 or 3 the associated squared Euclidean distance is $1.17 \times 4 = 4.68$. For the case depicted in Fig. 3(c) the minimum squared Euclidean distance $0.59 \times 6 = 3.54$. Thus these error events dominate the performance and should be dealt with before further increasing S_1 and S_2. Similarly, the minimum squared Euclidean distance associated with the error event depicted in Fig. 4 is $1.17 \times 6 = 7.02$, so this parameter also determines the performance for S_1 and S_2 larger than 16.

The symbol interleaved codes in particular might benefit from the extended interleaver design because of the larger multiplicity of error events with small number of inputs as opposed to the smaller number of such error events for bit interleaved codes.

The design method does not guarantee the existence of an interleaver that satisfies the design criteria. Whether such an interleaver exists at all depends upon the interleaver length N and the specific design parameter values $S_{1,2}$, $T_{1,2}$ and $X_{1,2}$.

If only Constraint 1 is imposed, a necessary and sufficient condition for at least one interleaver of length N to exist is:

$$N \ge S_1 S_2$$

To prove that this is a sufficient condition we show how to construct two different interleavers of length $S_1 S_2$. The proof for the necessary part is provided in [8].

Consider a block interleaver of dimensions $S_2 \times S_1$. Write the numbers $1 \ldots N$ by columns. Each column has length S_2 so the elements in each row differ by exactly S_2:

$$\begin{bmatrix} 1 & S_2 + 1 & \ldots & (S_1 - 1)S_2 + 1 \\ 2 & S_2 + 2 & \ldots & (S_1 - 1)S_2 + 2 \\ \vdots & \vdots & & \vdots \\ S_2 & 2S_2 & \ldots & S_1 S_2 \end{bmatrix}$$

Reading though the rows starting from the upper left corner we get an interleaver that doesn't satisfy Constraint 1:

$$[1\ S_2+1\ldots(S_1-1)S_2+1:2\ S_2+2\ldots\ldots S_1S_2].$$

The number S_2+1 is within distance S_1-1 to number $2 \Rightarrow |i-j| = S_1-1$ and $|f(i)-f(j)| = S_2-1$. There are at least two ways to avoid this:

1. Start from the lower left end, ie form the interleaver:
$[S_2\ 2S_2\ \ldots\ S_1S_2\ :\ S_2-1\ \ldots\ \ldots\ :2\ S_2+2\ \ldots\ (S_1-1)S_2+2\ :\ 1\ S_2+1\ \ldots\ (S_1-1)S_2+1]$.

2. Start from the upper right corner, ie form the interleaver:
$[(S_1-1)S_2+1\ \ldots\ S_2+1\ 1\ :\ (S_1-1)S_2+2\ \ldots\ S_2+2\ 2\ :\ \ldots\ \ldots\ S_2]$.

As has been noted in the literature, this kind of block interleaver does not perform well. Observe that Constraint 2 is consistently not met, since for each neighbor integers k, l the intergers $k+1$ and $l+1$ are also neighbor. Thus the spread interleaver is specifically designed to be semi-random; it is not sufficient to satisfy Constraint 1.

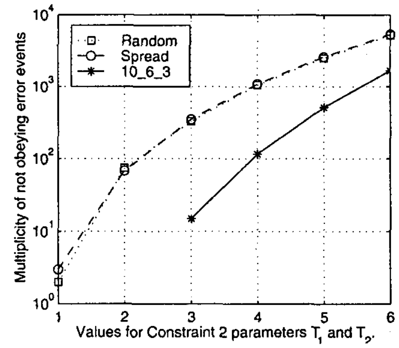
5 Simulation Results

This section provides simulation results for 2 bits/sec/Hz employing 8-PSK, and 4 bits/sec/Hz employing 64-QAM= $2 \times$ 8-PAM. The simulation system is a symbol-interleaved parallel-concatenated trellis-coded modulation (PCTCM) [2]. The interleavers used in the simulations are uniform random or semi-random, as specified in each case. To describe a semi-random interleaver we give the constraint parameters in the following order: (S_1-S_2, T_1-T_2, X_1-X_2).

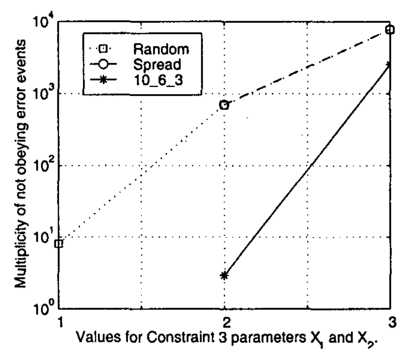
5.1 Multiplicities

For the simulation results we use the best interleavers we have identified up to now, generated semi-randomly under different constraints as described in Section 3. These interleavers exactly meet Constraint 1, but not Constraints 2 and 3. In practice at some point during the interleaver construction there is no unassigned number that satisfies these constraints. If a constraint cannot be met, the constraint parameters are gradually relaxed until the best available choice is identified. Thus, although the produced interleavers do not perfectly meet Constraints 2 and 3, they have a smaller multiplicity of minimum distance error events than an interleaver designed without taking these constraints into account.

For example the $(10-10, 6-6, 3-3)$ interleaver of length 4,096, always meets the $(S_1-S_2) = (10-10)$ constraint. Fig. 5(a) shows the multiplicity of times it fails Constraint 2 as a function of the value of $T_1 = T_2$ and Fig. 5(b) shows the multiplicity of times it fails Constraint 3 as a function of the value of $X_1 = X_2$ (zero multiplicity is not plotted). For comparison these figures also examine a uniform random interleaver, and a spread interleaver (designed imposing only Constraint 1) of the same length.



(a) Constraint 2.



(b) Constraint 3.

Figure 5: Multiplicity of error events that fail the indicated constraint for a uniform random interleaver, an interleaver designed only with Constraint 1, and the $(10-10, 6-6, 3-3)$ interleaver.

5.2 BER Performance

For 2 bits/sec/Hz PCTCM the constituents encoders implement a 4/3 code with 1 parity and 2 systematic outputs, have 4 memory elements, and employ 8-PSK modulation (labeling clock-wise: {7,6,5,4,3,2,1,0}). The simulated code in octal notation, has generator polynomials: $\{027, 01, 06, 010, 014, 011, 03\}$. The encoders structure is described in [2].

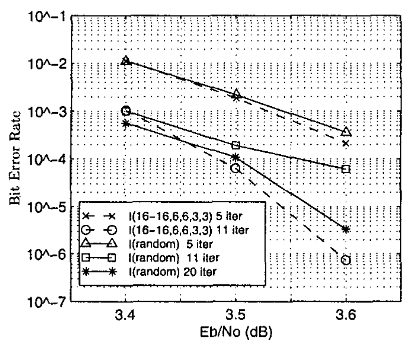


Figure 6: 2 bits/sec/Hz/ turbo code employing 8-PSK, with interleaver length 3,000 symbols. Constrained capacity=2.8 dB.

Observe that the semi-random interleaver (16-16,6-6,3-3) achieves BER of 10^{-6} with 11 iterations, while the uniform random interleaver needs 20 iterations to achieve slightly worse performance.

For 4 bits/sec/Hz PCTCM with 8×8 PAM=64 QAM, the constituents encoders implement a 4/3 code with 1 parity and 2 systematic outputs and have 4 memory elements. The 8 PAM labeling (left to right) is: {0,1,2,3,6,7,4,5}. The simulated code has generator polynomials in octal notation: $\{023, 01, 05, 013, 015, 015, 03\}$, and the interleaver has length 4,096 symbols.

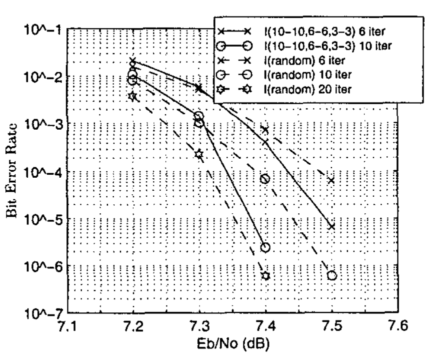


Figure 7: 4 bits/sec/Hz/ turbo code,with interleaver length 4,096 symbols and 8 PAM. Constrained Capacity 6.6 dB.

6 Conclusions

The natural extention of the semi-random interleaver design constraint, is to take into account multiple error events. This paper introduced extended constraints and tried to give some understanding why the spread-interleaver has to be semi-random, why just satisfying its single design constraint is not sufficient.

The design method does not guarantee the existence of an interleaver that satisfies the design constraints. We presented simulation results with interleavers that do not perfectly meet the constraints, but have a smaller multiplicity of minimum distance error events than an interleaver designed without taking these constraints into account.

References

[1] D. Divsalar, S. Benedetto, F. Pollara, and G. Montorsi. Turbo Codes: Principles and Applications. *Lecture Notes*, Octomber 1997.

[2] C. Fragouli and R. Wesel. Symbol Interleaved Parallel Concatenated Trellis Coded Modulation. *ICC Communications Miniconference*, June 1999.

[3] L. Perez, J. Seghers, and D. Costello. A Distance Spectrum Interpretation of Turbo Codes. *IEEE Transactions on Info. Theory*, November 1996.

[4] J. Andersen. Interleaver Design For Turbo Coding . *International Symposium on Turbo Codes*, 1997.

[5] M. Oberg and P. Siegel. Lowering The Error Floor For Turbo Codes . *International Symposium on Turbo Codes*, 1997.

[6] F. Daneshgaran and M. Mondin. Design of Interleavers For Turbo Codes Based On A Cost Function. *International Symposium on Turbo Codes*, 1997.

[7] J. Hokfelt and T. Maseng. Methodical Interleaver Design For Turbo Codes . *International Symposium on Turbo Codes*, 1997.

[8] C. Fragouli and R. Wesel. Turbo Encoder Design For Symbol Interleaved Parallel Concatenated Trellis Coded Modulation. *IEEE Transactions on Communications*, submitted April 1999.