On quasi-cyclic interleavers for parallel turbo codes *

Joseph J. Boutros and Gilles Zémor †

November 29, 2005

Abstract

We present an interleaving scheme that yields quasi-cyclic turbo codes. We prove that randomly chosen members of this family yield with probability almost 1 turbo codes with asymptotically optimum minimum distance, i.e. growing as a logarithm of the interleaver size. These interleavers are also very practical in terms of memory requirements and their decoding error probabilities for small block lengths compare favorably with previous interleaving schemes.

Index Terms: quasi-cyclic codes, convolutional codes, turbo codes, minimum distance, iterative decoding.

1 Introduction

It is now well known that the behaviour of turbo codes, although very powerful under high noise, exhibits an error floor phenomenon that can be explained by poor minimum distance properties. More specifically, it can be shown that for randomly chosen interleavers, the expected minimum distance of a classical two-level turbo code remains constant [21][19], i.e. does not grow with block length. Can the error floor behaviour of turbo codes be improved by designing the interleaver in a way that differs from pure random choice ?

This question has been addressed by many authors and the answer has been shown to be affirmative. Two types of approaches have been used in trying to find improved interleavers: the first tries to modify as little as possible a randomly chosen interleaver by combinatorially avoiding configurations that yield small-weight codewords. This is in principle manageable since the expected number of codewords of constant weight in a turbo code remains constant (does not grow with block length). Indeed, this approach has met with significant success: perhaps the most widely known design of this type is the S-*random* interleaver [13], that focuses on eliminating codewords of low weight corresponding

*Presented in part at the 2004 IEEE International Symposium on Information Theory, Chicago, Illinois, USA.

†J.J. Boutros and G. Zémor are with École Nationale Supérieure des Télécommunications, 46 rue Barrault, 75634 Paris 13, France. Email: {boutros,zemor}@enst.fr

to information sequences of weight 2. A more recent scheme of Truhachev et al. [30] weeds out all small-weight turbo-codewords in a way that is reminiscent of Gallager's method of excluding small cycles when constructing the parity-check matrix of an LDPC code, see also [3] for a similar result. They obtain turbo codes whose minimum distance grows proportionally to $\log N$ where N denotes the interleaver size. This asymptotic result is essentially the best possible since it was shown by Breiling [8] that $D/\log N$ must be upper bounded by a constant, where D and N denote respectively the minimum distance and interleaver size of the turbo code.

The second approach tries to find interleavers with structure, in particular algebraic structure, rather than mimic random choice. Besides enhanced performance, an additional motivation is to have a permutation with a short description that will save on the memory required to store the interleaver connections. With this last feature in mind, a particularly promising family of interleavers was proposed by Tanner in [29] and consists of *quasi-cyclic* permutations that yield quasi-cyclic turbo codes. Encouraging simulation results for the simpler RA codes were obtained in [28], hinting at good minimum distance properties of quasi-cyclic turbo-like codes. To quote from the conclusion of [29]:

"We conjecture that the class of quasi-cyclic permutations that will create quasi-cyclic turbo codes is rich, rich enough to contain codes that will perform as well or better than random interleavers."

In the present work we take up this challenge and study *random* quasi-cyclic permutations. Our approach borrows from both the unstructured, almost random, and the algebraic, structured, design strategies: our interleavers have structure, and the inherent advantageous storage properties, and yet involve a certain amount of random choice. Our main result is to show that the typical minimum distance of the associated turbo code grows linearly with $\log N$, i.e. has optimal growth, thus justifying Tanner's conjecture. Furthermore, for moderate lengths these interleavers turn out to be not only practical, but very efficient, comparing favorably with random, S-random, and all known interleavers in a number of instances.

The paper is organized as follows. In section 2 we give a short summary of previous work on interleaver design. In section 3 we describe our family of quasi-cyclic interleavers. Section 4 is devoted to proving that a randomly chosen interleaver from this family will have asymptotically optimal minimum distance with high probability (Theorem 9). Section 5 gives experimental results for short lengths ($N = 400$ and $N = 1600$).

2 Previous work on interleaver design

Classical channel coding systems using a serial concatenation of Reed-Solomon codes and binary convolutional codes include a matrix interleaver (also called *block interleaver*) which enables one to split the error bursts generated by the Viterbi decoder before applying an algebraic Berlekamp-Massey decoding [7][20]. The early research on interleavers for digital communications has been preceded by the invention of burst-error-correcting cyclic and burst-error-correcting convolutional codes [22]. Although the low density parity check codes developed by R. Gallager [16] integrated random interleaving of the parity-check matrix columns, no serious study on interleavers was known until the work by Ramsey on optimum interleavers for infinite length sequences [24]. For example, a type I Ramsey interleaver guarantees an input separation n_1 and an output separation n_2 with a minimum delay equal to $n_2(n_1-1)$, where n_1 and n_2 are two positive integers satisfying $n_2 < n_1 < 2n_2$, n_1 and n_2+1 are relatively prime.

The separation guaranteed by Ramsey interleavers has been named *spreading* after the invention of parallel turbo codes based on binary systematic recursive convolutional constituents [5][6]. Finite length interleavers or permutations designed for parallel turbo codes have been extensively studied during the last decade. The amount of publications on the subject ranges in the hundreds and cannot be listed here in full. The following selection of interleaver families is an attempt to give a meaningful picture of the state of research and to summarize the main techniques. As mentioned earlier they can be crudely partitioned into two categories: mostly random interleavers with a weak structure, requiring an exhaustive description of the permutation, or strongly structured with short representations.

- **Purely random interleavers**.

  These interleavers are built from permutations on N integers selected at random. Here, N denotes the interleaver size. The original turbo codes ($N = 65536$ bits) were designed with purely random interleavers. Without any interleaver optimization, the error rate performance of parallel turbo codes can be enhanced via primitive feedback polynomials in the turbo code convolutional constituents [4].

- **Random interleavers with a weak deterministic structure**.

  This family includes the S-random or spread interleaver proposed by Divsalar and Pollara [13]. The S-random interleaver π is constructed at random, it must satisfy the constraint $|\pi(i) - \pi(j)| > S$ for all $|i-j| < S$, where the maximal theoretical value of the spread S is $\sqrt{N}$. High spread random (HSR) interleavers proposed by Crozier [11] belong to this family. They rely on the maximization of the spread $S = \min\{|i-j| + |\pi(i) - \pi(j)|\}$ (also defined in [2] for arithmetic and random interleavers). The spread of HSR interleavers is upper bounded by $\sqrt{2N}$. The permutation described by Truhachev et.al. [30] that guarantees an asymptotically optimal minimum distance is mostly random with a weak deterministic structure.

- **Deterministic algebraic/arithmetic interleavers**.

  Many algebraic permutations have been suggested or specifically developed for parallel turbo codes. In spite of (or perhaps because of) their very low memory, they tend to exhibit intermediate or poor error rate performance. A short selection consists of the interleavers described by Berrou and Glavieux [6], by Andrews et.al. [2], Sadjadpour et.al. [25], Bravo and Kumar [9]. The Relative Prime and the Golden interleavers described by Crozier et.al. [10] belong to this family of deterministic interleavers. More recently, good interleavers based on permutation polynomials have been proposed by Sun and Takeshita [27].

- **Deterministic interleavers with a weak random structure**.

  We mention two types of deterministic interleavers where randomness has been added in order to unbalance somewhat the algebraic structure. Dithered golden interleavers [10] and dithered relative prime (DRP) interleavers [12]. DRP interleavers exhibit excellent error rate performance. They are obtained in 3 steps: 1- application of a small permutation (input dithering) to the interleaver input, e.g., a size 8 permutation applied $N/8$ times, 2- a relative prime permutation $j = s + ip$, where j is the read position, i is the write position, s is a shift and p is prime relative to N, 3- an output dithering similar to the input one.

- **Interleavers from the graphical structure of codes**.

  Cayley-Katz graphs with large girth have been used to design Generalized Low Density (GLD) codes with binary BCH constituents [23]. Similar application was made by Vontobel [31] to design turbo code interleavers from large girth graphs. Interleavers based on large girth graphs are all deterministic. Yu et.al. [32] also designed good interleavers by looking at the loop distribution in the turbo code structure. Such interleavers are random with a weak deterministic structure.

- **Interleavers by other criteria**.

  Abbasfar and Yao [1] recently proposed good interleavers that eliminate codewords with Hamming weight less than a certain distance. The construction algorithm is based on a two dimensional representation of the permutation. This representation previously inspired Crozier [11] in his design of dithered-diagonal interleavers. The interleaver design by distance spectrum shaping can be classified in the class of random interleavers with a weak deterministic structure. Finally, we mention the interleavers designed by Hokfelt et.al. [18] via the minimization of the correlation between extrinsic informations under iterative decoding.

The bi-dimensional quasi-cyclic interleaver described in the next section combines randomness and determinism in an almost equal manner. When designed from a square matrix, its quasi-cyclicity period is $\sqrt{N}$, meaning that a set of $2\sqrt{N}$ integers is needed to save the bi-dimensional interleaver into memory, rather than the N integers needed for a purely random permutation.

3 Bi-dimensional, or quasi-cyclic interleavers

For simplicity, we restrict ourselves to the classical turbo code construction with a fixed constituent convolutional code C_0 of rate $R_0 = 1/2$. The turbo encoder takes an information sequence $\mathbf{s}$ of N bits, produces a first sequence of N check bits by submitting

$$\sigma = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 \\ 3 & 2 & 0 & 4 & 1 \end{pmatrix} \qquad X_0 = 0, X_1 = 3, X_2 = 4, X_3 = 2, X_4 = 1.$$

$$\underset{\mathbf{A}}{\begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 & 9 \\ 10 & 11 & 12 & 13 & 14 \\ 15 & 16 & 17 & 18 & 19 \\ 20 & 21 & 22 & 23 & 24 \end{bmatrix}} \quad \underset{\mathbf{B}}{\begin{bmatrix} 3 & 2 & 0 & 4 & 1 \\ 8 & 7 & 5 & 9 & 6 \\ 13 & 12 & 10 & 14 & 11 \\ 18 & 17 & 15 & 19 & 16 \\ 23 & 22 & 20 & 24 & 21 \end{bmatrix}} \quad \underset{\mathbf{C}}{\begin{bmatrix} 3 & 12 & 5 & 19 & 21 \\ 8 & 17 & 10 & 24 & 1 \\ 13 & 22 & 15 & 4 & 6 \\ 18 & 2 & 20 & 9 & 11 \\ 23 & 7 & 0 & 14 & 16 \end{bmatrix}}$$

$$(\pi(0), \pi(1), \pi(2), \ldots, \pi(24)) =$$
$$(3, 12, 5, 19, 21, 8, 17, 10, 24, 1, 13, 22, 15, 4, 6, 18, 2, 20, 9, 11, 23, 7, 0, 14, 16).$$

Figure 1: Example: construction of π, $N = 25$, $n_1 = n_2 = 5$. Write $0, 1, \ldots, N-1$ in a square array $\mathbf{A}$, apply σ to permute columns, giving $\mathbf{B}$, and rotate column j cyclically, by X_j mod 5, giving array $\mathbf{C}$. Read off the rows to get $\pi(0), \pi(1), \pi(2), \ldots$.

$\mathbf{s} = s_0, s_1, \ldots s_{N-1}$ to an encoder for C_0, and a second sequence of a further N check bits by submitting a permuted version $s_{\pi(0)}, s_{\pi(1)}, \ldots, s_{\pi(N-1)}$, of $\mathbf{s}$ to the encoder for C_0. The overall turbo code rate is $R = 1/3$ and the *interleaver* is the permutation π on the ordered set of information coordinates $\mathcal{N} = \{0, 1, \ldots, N-1\}$.

There will be a 2-dimensional structure inherent to our choice of permutation π, therefore we shall restrict ourselves to the case when $N = n_1 \times n_2$ is a composite integer.

Let π be a permutation on $\mathcal{N} = \{0, 1, \ldots, N-1\}$ defined as follows. For any $(i, j) \in \{0, 1, \ldots, n_1 - 1\} \times \{0, 1, \ldots, n_2 - 1\}$ define the function

$$\begin{aligned} \phi : \{0, 1, \ldots, n_1 - 1\} \times \{0, 1, \ldots, n_2 - 1\} &\rightarrow \mathcal{N} \\ (i, j) &\mapsto i \times n_2 + j \end{aligned}$$

Let σ be a permutation of $\{0, 1, \ldots, n_2 - 1\}$ and let $(X_j)_{j=0\ldots n_2-1}$ be a family of integers mod n_1. Define the permutation Π on $\{0, 1, \ldots, n_1 - 1\} \times \{0, 1, \ldots, n_2 - 1\}$ by

$$\Pi(i, j) = (i + X_j \bmod n_1, \sigma(j)).$$

Finally define the permutation $\pi = \phi \Pi \phi^{-1}$ on the set $\mathcal{N}$. A small example is given in Figure 1.

The quasi-cyclic nature of the permutation π just defined is stressed in the following Lemma, a direct consequence of the definition.

Lemma 1 *A permutation π belonging to the class defined above satisfies, for any $x, x' \in \mathcal{N}$ such that $x' = x + n_2$ mod N,*

$$\pi(x') = \pi(x) + n_2 \bmod N.$$

If we make the trellis of the constituent convolutional code tail-biting, and if we write the check bits of the turbo code in the proper order, we obtain a quasi-cyclic turbo code [29]. For this reason, permutations of the above type will be called (n_1, n_2)-*quasi-cyclic* (or simply quasi-cyclic).

We shall take instances of (n_1, n_2)-quasi-cyclic permutations π by choosing the permutation σ randomly, with uniform distribution, among permutations of $\{0, 1, \ldots, n_2 - 1\}$, and by choosing the $X_i, i = 0 \ldots n_2 - 1$ randomly, with uniform distribution, among the set of integers mod n_1. We choose the X_i to be independent of each other and of σ. This is a way of choosing π uniformly in the class of (n_1, n_2)-quasi-cyclic permutations.

As a first comment, we may note that π has quite a lot more structure than a totally random permutation. A certain amount of randomness remains however; to quantify it somewhat, suppose for example that $n_1 = n_2 = n = \sqrt{N}$ (we shall see experimentally in section 5 that $n_1 = n_2$ is a good choice), we see that π is defined by $\log n! + n \log n \approx \sqrt{N} \log N$ random bits as opposed to the $N \log N$ bits that define an otherwise unstructured permutation.

Our strategy will be probabilistic, i.e. we will estimate the probability that the permutation π produces turbo code words of small weight $w << \log N$ and show that this probability must be vanishingly small. Interestingly, over all permutations π, the *expected* number of turbo codewords of small weight w does not vanish with N. This is because of an all-or-nothing phenomenon. Permutations of the above type produce either no turbo codewords of small weight, or relatively many (at least $n = \sqrt{N}$).

4 Minimum distance analysis

For any two integers x and y of $\mathcal{N}$ let us denote by $d(x, y)$ the circular distance between x and y, i.e. the smallest non-negative integer d such that $x + d = y \bmod N$ or $x - d = y \bmod N$. Let us draw an edge between x and y whenever $d(x, y) = 1$, giving $\mathcal{N}$ a circular structure: by an interval of $\mathcal{N}$ we shall mean a sub-path of $\mathcal{N}$.

Let $\mathbf{s} = s_0, \ldots, s_{N-1}$ be an information sequence, and let $\mathbf{v} \subset \mathcal{N}$ be the support of $\mathbf{s}$. The information sequence $\mathbf{s}$ generates a path in the (tail-biting) trellis of the convolutional code C_0. Consider the partition $\mathcal{N} = \mathcal{Z} \cup \mathcal{T}$ where $\mathcal{Z}$ is defined as the set of coordinates i for which the path associated to $\mathbf{s}$ goes from the zero state to the zero state. For every $i \in \mathcal{Z}$ we have $s_i = 0$ and the corresponding check bit is also 0. The complement $\mathcal{T}$ of $\mathcal{Z}$ is a union of intervals $\mathcal{T} = [a_1, b_1] \cup [a_2, b_2] \cup \ldots \cup [a_m, b_m]$. The intervals $[a_j, b_j]$ are sometimes called simple trellis paths, or simple error events in the convolutional coding terminology. The trellis of a recursive convolutional code has the property that the zero state can only be left at time t if $s_t = 1$ and it can only be reached from a nonzero state at time t if $s_t = 1$. This means that $\mathbf{v} \subset \mathcal{T}$ and every interval $[a_j, b_j], j = 1 \ldots m$ starts and ends with an element of the support $\mathbf{v}$ of $\mathbf{s}$. A recursive convolutional code also has the property of outputting a steady stream of non-zero symbols during the time it goes through a simple trellis path, i.e. during the time it is fed the information bits s_t for $t \in [a_j, b_j]$. In other words, there exists a constant λ, depending only on C_0, such that

Rate 1/2 RSC codes		
Octal Generators	Number of States	Parameter λ
(7, 5)	4	1/2
(13, 15)	8	2/5
(17, 15)	8	1/2
(37, 21)	16	1/4
(23, 35)	16	4/11

Table 1: Rate 1/2 recursive systematic convolutional codes. The parameter λ is the minimal ratio of Hamming weight to trellis length among all codewords.

the total weight of the convolutional codeword associated to the information sequence $\mathbf{s}$ is at least $\lambda \sum_{j=1}^{m} d(a_j, b_j)$. Some numerical values of λ are given in Table 1. Those values are found by classical transfer function techniques [14] as described in section 4.7 of [26]. Using the state transition matrix, λ is equal to the minimal ratio of Hamming weight to trellis length over all cycles determined by raising the state transition matrix to powers less than or equal to the number of non-zero states. Summarizing:

Facts: Associated to any information sequence $\mathbf{s}$ of support $\mathbf{v}$ there is a subset $\mathcal{T}(\mathbf{s}) \subset \mathcal{N}$ such that

1. $\mathcal{T}(\mathbf{s}) = [a_1, b_1] \cup [a_2, b_2] \cup \ldots \cup [a_m, b_m]$ is a union of intervals of $\mathcal{N}$
2. $\mathbf{v} \subset \mathcal{T}(\mathbf{s})$
3. for any $j = 1 \ldots m$, $|\mathbf{v} \cap [a_j, b_j]| \geq 2$
4. on input $\mathbf{s}$ the convolutional encoder outputs at least $\lambda \sum_{j=1}^{m} d(a_j, b_j)$ nonzero symbols, for some positive constant λ

Let us call the *trellis weight* of $\mathbf{s}$ the quantity $\sum_{j=1}^{m} d(a_j, b_j)$ defined above, denote it by $W_T(\mathbf{s})$. Now the turbo code word associated to $\mathbf{s}$ has its weight lower-bounded by both convolutional codewords corresponding to the input $\mathbf{s}$ and to the permuted input $\mathbf{s}^{\pi}$. Since the maximum weight of the two convolutional codewords is lower-bounded by half their sum, Fact 4 above implies:

Lemma 2 *If the information sequence* $\mathbf{s}$ *produces a turbo codeword of Hamming weight* w*, then* $W_T(\mathbf{s}) + W_T(\mathbf{s}^{\pi}) \leq 2w/\lambda$*.*

This last lemma says that low-weight turbo codewords can only exist if there is an information sequence $\mathbf{s}$ such that both $\mathbf{s}$ and $\mathbf{s}^{\pi}$ have small trellis weight. Now the decomposition of the supports of $\mathbf{s}$ and $\mathbf{s}^{\pi}$ into simple trellis paths is rather awkward to handle probabilistically, so we shall introduce a related concept that will be easier to deal with. The following definition is purely combinatorial.

Definition 3 *Let* $\mathbf{x} = x_0, x_1, \dots x_\ell$, *$\ell$ odd, be an even-numbered sequence of elements of* $\mathcal{N}$. *Let* $y_i = \pi(x_i), i = 0 \dots \ell$ *for some permutation* π *of* $\mathcal{N}$. *Let us call the* π-weight *of* $\mathbf{x}$ *the quantity:*

$$w_\pi(\mathbf{x}) = \sum_{1 \le i, 2i < \ell} d(x_{2i-1}, x_{2i}) + d(x_0, x_\ell) + \sum_{0 \le i, 2i+1 \le \ell} d(y_{2i}, y_{2i+1}).$$

Note that the quantity w_π is essentially the summarized distance of [30]. The reason for introducing the above definition lies in the following lemma.

Lemma 4 *If there exists a codeword of weight* w *in the turbo code with interleaver* π*, then there exists an even-numbered sequence* $\mathbf{x}$ *of distinct elements of* $\mathcal{N}$ *of* π*-weight* $w_\pi(x) \le 2w/\lambda$.

Proof : Let $\mathbf{s}$ be the information sequence corresponding to the turbo codeword of weight w and let $\mathbf{v}$ be its support. Note that the support of $\mathbf{s}^\pi$ is $\mathbf{u} = \pi^{-1}(\mathbf{v})$. Let $\mathcal{T}(\mathbf{s}) = [a_1, b_1] \cup [a_2, b_2] \cup \dots \cup [a_m, b_m]$ be the decomposition of $\mathcal{T}(\mathbf{s})$ into m intervals given by Fact 1 and let $\mathcal{T}(\mathbf{s}^\pi) = [a'_1, b'_1] \cup [a'_2, b'_2] \cup \dots \cup [a'_k, b'_k]$ be the corresponding decomposition for the permuted version $\mathbf{s}^\pi$ of the information sequence. Now consider the bipartite graph whose vertex set is made up of the two sets A and B where A is the set of the k intervals $[a'_i, b'_i], j = 1 \dots k$, and B is the set of the m intervals $[a_j, b_j], j = 1 \dots m$. Put an edge between interval $[a'_i, b'_i]$ and $[a_j, b_j]$ for every $x \in \pi^{-1}(\mathbf{v}) \cap [a'_i, b'_i]$ such that $\pi(x) \in [a_j, b_j]$ (multiple edges may occur). Fact 2 implies that the minimum degree of the bipartite graph is at least 2. Therefore there exists an (even-length) elementary cycle in the graph, i.e. a string of distinct vertices $V_0, V_1, \dots V_\ell$, ℓ odd, where the interval V_i belongs to A (respectively B) for i even (respectively odd) and where there is an edge between V_i and V_j whenever $i - j = \pm 1 \bmod \ell + 1$. For $0 \le i, 2i < \ell$ the edge between V_{2i} and V_{2i+1} is defined by an element of $V_{2i} \cap \mathbf{u}$ that we denote x_{2i}, and an element of $V_{2i+1} \cap \mathbf{v}$ that we denote y_{2i} and that equals $y_{2i} = \pi(x_{2i})$. Similarly, for $1 \le i = 1, 2i < \ell$, every edge between V_{2i} and V_{2i-1} is associated to $x_{2i-1} \in V_{2i}$ and $y_{2i-1} \in V_{2i-1}$ with $y_{2i-1} = \pi(x_{2i-1})$. Finally let $x_\ell \in V_0$ and $y_\ell = \pi(x_\ell) \in V_\ell$ correspond to the edge between V_0 and V_ℓ.

We have constructed a sequence $\mathbf{x} = x_0, x_1, \dots, x_\ell$ of elements of the support $\mathbf{u} = \pi^{-1}(\mathbf{v})$ of $\mathbf{s}^\pi$ such that $\{x_{2i-1}, x_{2i}\} \subset V_{2i}$, $1 \le i, 2i \le \ell$, $\{x_\ell, x_0\} \subset V_0$, and $\{y_{2i}, y_{2i+1}\} \subset V_{2i+1}$, $0 \le i, 2i+1 \le \ell$: see Figure 2. Therefore, denoting by $\mathcal{L}(V)$ the length of an interval V, we have

$$w_\pi(\mathbf{x}) \le \sum_{i=0}^{\ell} \mathcal{L}(V_i) \le W_T(\mathbf{s}) + W_T(s^\pi)$$

which proves the result by Lemma 2. ∎

We shall now study the probability that a sequence of small π-weight exists. We need some more notation.

Let $\mathbf{r} = r_1, r_2, \dots, r_\ell$, ℓ odd, be a sequence of integers modulo N. Let $|r_i|$ denote the smallest absolute value of a (possibly negative) integer equal to r_i modulo N, and

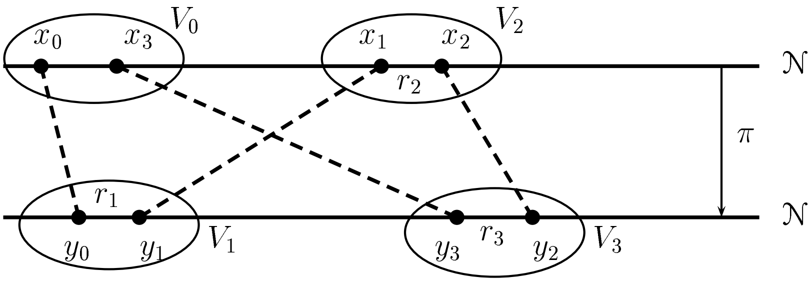

Figure 2: $\ell = 3$. The cycle V_0, V_1, V_2, V_3 defined in the proof of Lemma 4, the associated sequences $\mathbf{x} = x_0, x_1, x_2, x_3$, $\mathbf{y} = y_0, y_1, y_2, y_3$, $\mathbf{r} = r_1, r_2, r_3$.

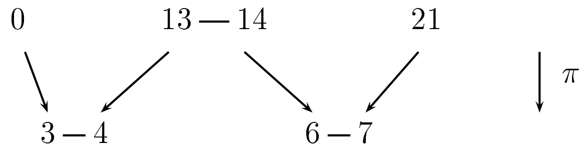

Figure 3: Let π be the same as in Figure 1. The sequence $\mathbf{r} = 1, 1, 1$ together with $x_0 = 0$ define $\mathbf{x} = \mathbf{x}(\mathbf{r}, x_0) = 0, 13, 14, 21$. We have $w_\pi(\mathbf{x}) = 1 + 1 + 1 + d(0, 21) = 7$.

let $\|\mathbf{r}\| = |r_1| + |r_2| + \ldots + |r_\ell|$. Let $x_0 \in \mathcal{N}$. Together x_0 and $\mathbf{r}$ uniquely define the $(\ell+1)$-sequence $\mathbf{x} = \mathbf{x}(\mathbf{r}, x_0) = x_0, x_1, \ldots, x_\ell$ and $\mathbf{y} = \mathbf{y}(\mathbf{r}, x_0) = y_0, \ldots, y_\ell$ such that

1. $y_i = \pi(x_i), i = 0 \ldots \ell$,
2. for all $i \geq 0$ such that $2i + 1 \leq \ell$, $y_{2i+1} = y_{2i} + r_{2i+1} \bmod N$
3. for all $i \geq 1$ such that $2i < \ell$, $x_{2i} = x_{2i-1} + r_{2i} \bmod N$

Note that

$$w_\pi(\mathbf{x}(\mathbf{r}, x_0)) = \|\mathbf{r}\| + d(x_0, x_\ell). \tag{1}$$

Finally, let us say that the sequence $\mathbf{r}$ *M-cycles* at x_0 if

$$w_\pi(\mathbf{x}(\mathbf{r}, x_0)) \leq M.$$

The definitions are illustrated in Figure 3.

Lemma 4 translates directly into the following, given that the x_i are distinct only if all the r_i are non-zero.

Lemma 5 *If there exists a turbo codeword of weight w then there exists $x_0 \in \mathcal{N}$ and a non-zero sequence $\mathbf{r} = r_1, \ldots, r_\ell$, $r_i \neq 0, i = 1 \ldots \ell$, ℓ is odd, that $2w/\lambda$-cycles at x_0.*

We now have everything in place for doing the probabilistic analysis. Let $Z_{\mathbf{r},x_0}$ be the Bernoulli random variable equal to 1 if the sequence $\mathbf{r}$ M-cycles at x_0 and equal to 0 otherwise. The set of all permutations π of the set $\mathcal{N}$ is endowed with two probability measures, namely:

- the uniform probability measure P_r, in other words $P_r(\pi) = 1/N!$ for all π.
- the quasi-cyclic probability measure P_q defined by $P_q(\pi) = 1/(n_1^{n_2} n_2!)$ if π is (n_1, n_2)-quasi-cyclic and $P_q(\pi) = 0$ otherwise. Note that, as mentioned in section 3, this is equivalent to choosing the permutation σ randomly, with uniform distribution, among permutations of $\{0, 1, \ldots, n_2 - 1\}$, and by choosing the $X_i, i = 0 \ldots n_2 - 1$ randomly, independently of each other and of σ, and with uniform distribution, among the set of integers mod n_1.

Lemma 6 *Let $M < N$, x_0 and $\mathbf{r} = r_1, \ldots, r_\ell$ be given, $r_i \neq 0, i = 1 \ldots \ell$, ℓ is odd. We have:*

1. *If $\|\mathbf{r}\| \geq M$ then $P_r[Z_{\mathbf{r},x_0} = 1] = P_q[Z_{\mathbf{r},x_0} = 1] = 0$.*
2. *If $\|\mathbf{r}\| < M$ then $P_r[Z_{\mathbf{r},x_0} = 1] < 2M/(N-1)$.*
3. *If $\|\mathbf{r}\| < M < n_2$, then $P_q[Z_{\mathbf{r},x_0} = 1] < \frac{2M}{n_1(n_2-1)}$.*
4. *If $\|\mathbf{r}\| < M$ and $r_i = 0 \bmod n_2$ for every $i = 1 \ldots \ell$, then $P_q[Z_{\mathbf{r},x_0} = 1] = 1$.*

Proof : Point 1 is a direct consequence of (1).

To see Point 2, consider $x_1, \ldots, x_\ell$ as random variables. Conditional on the position of $x_{\ell-1}$, i.e. on the event $x_{\ell-1} = k$, the distribution of x_ℓ is, since $r_\ell \neq 0$, uniform on the set $\mathcal{N} \setminus \{k\}$. Therefore

$$\begin{aligned} P_r(Z_{\mathbf{r},x_0} = 1) &= \sum_k P_r[d(x_\ell, x_0) \leq M - \|\mathbf{r}\| \mid x_{\ell-1} = k] P_r[x_{\ell-1} = k] \\ &\leq \sum_k \frac{1 + 2(M - \|\mathbf{r}\|)}{N-1} P_r[x_{\ell-1} = k] \quad = \quad \frac{1 + 2(M - \|\mathbf{r}\|)}{N-1} \\ &< \frac{2M}{N-1}. \end{aligned}$$

To see Point 3 argue as follows: write $d(x_0, x_\ell) = qn_2 + \rho$, $0 \leq \rho < n_2$. Since we have supposed $M < n_2$ we have $Z_{\mathbf{r},x_0} = 1$ if and only if

(*a*) $q = 0$

(*b*) $\rho \leq M - \|\mathbf{r}\|$.

Since $r_\ell \neq 0$ and $M < n_2$ imply that $r_\ell \neq 0 \bmod n_2$, we can argue as in point 2, replacing the random permutation π of $\{0, 1, \ldots, N-1\}$ by the random permutation σ of $\{0, 1, \ldots, n_2 - 1\}$, to obtain that the event (*b*) occurs with probability not more than $2M/(n_2 - 1)$. By construction of the quasi-cyclic permutation the event (*a*) is independent of (*b*) and occurs with probability $1/n_1$.

Point 4 is simply due to the fact that for quasi-cyclic π we have $\pi(x_0 + n_2) = \pi(x_0) + n_2 \bmod N$ for any $x_0 \in \mathcal{N}$, therefore $d(x_0, x_\ell) = \sum_{1 \leq i \leq \ell} r_i < M$. ∎

Next, we shall study the expected number of couples $(\mathbf{r}, x_0)$ for which $\mathbf{r}$ M-cycles at x_0, i.e. the expectation of the random variable

$$Z = \sum_{x_0 \in \mathcal{N}, r_i \neq 0, \|\mathbf{r}\| < M} Z_{\mathbf{r}, x_0}. \tag{2}$$

Since $\mathbf{r}$ must have only non-zero terms, its length ℓ cannot exceed its norm $\|\mathbf{r}\|$. The number of sequences of given norm m, length ℓ and non-negative terms is exactly $\binom{m}{\ell}$, so that the number of terms in the sum (2) is not more than

$$N \sum_{1 \leq m < M} \sum_{0 \leq \ell \leq m} 2^{\ell} \binom{m}{\ell} = N \sum_{1 \leq m < M} 3^m < N 3^M / 2.$$

From this and Point 3 of Lemma 6 we obtain therefore:

Lemma 7 *Let $M < n_2$. The expected number $E_q[Z]$ of couples $(\mathbf{r}, x_0)$ such that $\mathbf{r}$ M-cycles at x_0 satisfies, for the probability measure P_q,*

$$E_q[Z] \leq M 3^M (1 - 1/n_2)^{-1}.$$

Notice that Point 2 of Lemma 6 would give essentially the same estimate of the expected value of Z for uniformly random π. However, the crucial property of the class of quasi-cyclic permutations that will make a big difference between choosing π uniformly random and quasi-cyclic-random is the following direct consequence of Lemma 1:

Lemma 8 *If the sequence $\mathbf{r}$ M-cycles at x_0 for a quasi-cyclic π, then $\mathbf{r}$ M-cycles at $x_0 + n_2 \bmod N$ for π. In particular, Z is a multiple of n_1 for every quasi-cyclic permutation π.*

This means that $E_q[Z] = \sum_{z \geq n_1} z P_q[Z = z] \geq n_1 P_q[Z > 0]$. We have therefore that whenever the quantity $E_q[Z]/n_1$ is made to be vanishing with N, the probability that there exists a sequence $\mathbf{x}$ of π-weight not more than M tends to zero. Putting together Lemma 7 and Lemma 4 we obtain this section's main result :

Theorem 9 *For any constant $C < \lambda/2$ and block length $N = n_1 n_2$ chosen to satisfy $n_2 > \frac{2C}{\lambda} \log_3 n_1$, the minimum distance of the random quasi-cyclic turbo code satisfies, with probability that tends to 1 as n_1 tends to infinity,*

$$D \geq C \log_3 n_1$$

In particular $D \geq \frac{C}{2} \log_3 N$ when $n_1 = n_2$.

5 Experimental results and concluding comments

In this section, we provide computer simulation results for the word error rate (WER) of parallel turbo codes using our new family of interleavers and comparing it with S-random and random interleavers. The output of the turbo encoder is modulated via a binary phase shift keying (BPSK) modulation and transmitted over an ideal additive white gaussian noise (AWGN) channel. The turbo decoder performs iterative a posteriori probability estimation by applying the forward-backward algorithm to each convolutional constituent.

Word error rate versus signal-to-noise ratio results are depicted in Figures 4 and 5. In the first example, Fig. 4 illustrates the performance of a rate 1/2 turbo code with an 8-state recursive systematic convolutional constituent $(13, 15)_8$. These octal generators have been adopted in the European third generation mobile radio standard UMTS [15]. In this example the interleaver size is $N = 400$, and we compare the performance of a random interleaver, a S-random interleaver, and two quasi-cyclic interleavers with $n_1 = n_2 = 20$: one quasi-cyclic permutation was randomly chosen and the other corresponds to the best we were able to find. The exact quasi-cyclic permutation is given in Table 2: note the short description that the quasi-cyclic structure allows. As shown in Fig. 4, the bi-dimensional interleaver clearly outperforms the spread interleaver. The increase in minimum distance can in principle also be validated numerically by measuring the turbo code minimum distance using the algorithm proposed by Garello et al. [17].

In the second example, the rate 1/2 turbo code has a recursive systematic convolutional constituent $(37, 21)_8$, the octal generators proposed in the original turbo code by Berrou et al. [5]. The interleaver size is $N = 1600$, $n_1 = n_2 = 40$, and the exact permutation is given in Table 3. As shown in Fig. 5, we again get a significant improvement over the S-random interleaver.

After a number of experiments, it turned out that the optimum choice of the values of n_1 and n_2, at least for the moderate lengths we experimented with, is very close to $\sqrt{N}$. This is a phenomenon for which we were unable to get a satisfying theoretical explanation.

Square Bi-dimensional Quasi-cyclic Interleaver of Size 400																				
σ	2	10	0	9	1	8	4	13	7	14	3	11	6	12	17	5	15	16	18	19
X	6	2	12	0	5	19	3	1	4	17	10	18	9	8	7	11	15	14	13	16

Table 2: Bi-dimensional interleaver of size $400 = 20 \times 20$. The first row defines the column permutation σ and the second row defines the column cyclic shift X. This square interleaver is used in conjunction with RSC(13,15) in Fig. 4.

Square Bi-dimensional Quasi-cyclic Interleaver of Size 1600																				
σ	1	15	17	18	25	39	33	29	19	4	0	37	14	20	27	9	22	31	10	28
	30	36	23	35	7	16	6	2	13	26	3	34	32	21	11	8	5	38	12	24
X	29	30	21	10	39	11	26	4	28	15	22	25	31	3	34	23	18	17	32	27
	0	9	1	19	24	36	2	37	6	35	14	33	20	13	8	12	5	16	38	7

Table 3: Bi-dimensional interleaver of size $1600 = 40 \times 40$. The first two rows define the column permutation σ and the last two rows define the column cyclic shift X. This interleaver is used in conjunction with RSC(37,21) in Fig. 5.

References

[1] A. Abbasfar and K. Yao, "Interleaver design for turbo codes by distance spectrum shaping," in *Proc. IEEE Wireless Comm. and Net. Conf. (WCNC'04)*, vol. 3, pp. 1616-1619, Atlanta, March 2004.

[2] K.S. Andrews, C. Heegard and D. Kozen, "Interleaver design methods for turbo codes," in *Proc. IEEE Int. Symp. Information Theory (ISIT'98)*, pp. 420, Cambridge, MA, Aug. 1998.

[3] L. Bazzi, M. Mahdian and D. Spielman, "The minimum distance of turbo-like codes," Submitted to the IEEE Transactions on Information Theory, May 2003. Downloadable from http://math.mit.edu/∼spielman/Research/mindist.html

[4] S. Benedetto and G. Montorsi, "Design of parallel concatenated convolutional codes," *IEEE Trans. on Communications*, vol. 44, no. 5, pp. 591-600, May 1996.

[5] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding : turbo-codes," *Proceedings of ICC'93*, Geneva, pp. 1064-1070, May 1993.

[6] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: Turbo-codes," *IEEE Trans. on Communications*, vol. 44, pp. 1261-1271, Oct. 1996.

[7] R.E. Blahut, Algebraic Codes for Data Transmission, Cambridge University Press, 2003.

[8] M. Breiling, "A Logarithmic Upper Bound on the Minimum Distance of Turbo Codes," *IEEE Trans. Inform. Theory*, vol. 50, no. 8, pp. 1692–1710, Aug. 2004.

[9] C.J. Corrada Bravo and P.V. Kumar, "Permutation polynomials for interleavers in turbo codes," in *Proc. IEEE Int. Symp. Information Theory (ISIT'03)*, pp. 318, Yokohama, Japan, Jul. 2003.

[10] S.N. Crozier, J. Lodge, P. Guinand, and A. Hunt, "Performance of turbo codes with relative prime and golden interleaving strategies," in *Proc. of the 6th Int. Mobile Satellite Conf. (IMSC'99)*, Ottawa, Ontario, pp. 268-275, June 1999.

[11] S.N. Crozier, "New high-spread high-distance interleavers for turbo-codes," in *Proc. of the 20th Biennial Symp. on Communications*, Queen's University, Kingston, Ontario, Canada, pp. 3-7, May 2000.

[12] S.N. Crozier and P. Guinand, "High-performance low-memory interleaver banks for turbo-codes," in *Proc. of the 54th IEEE Veh. Tech. Conf. (VTC'01)*, Atlantic City, NJ, pp. 2394-2398, Oct. 2001.

[13] D. Divsalar, F. Pollara, "Turbo codes for PCS applications," in *Proc. IEEE Int. Conf. on Communications (ICC'95)*, vol. 1, pp. 54-59, Seattle, June 1995.

[14] D. Divsalar, S. Dolinar, F. Pollara, and R.J. McEliece, "Transfer function bounds on the performance of turbo codes," *JPL TDA Progress Report 42-122*, pp. 44-55, August 1995.

[15] European Telecommunications Standards Institute, "Universal mobile telecommunication system (UMTS); multiplexing and channel coding (FDD)," 3GPP TS 25.212, version 6.2.0, Release 6, 2004.

[16] R.G. Gallager, Low-density parity-check codes, MIT Press, 1963.

[17] R. Garello, P. Pierleoni and S. Benedetto, "Computing the free distance of turbo codes and serially concatenated codes with interleavers: Algorithms and applications," *IEEE Journal on Sel. Areas in Communications*, vol. 19, no. 5, pp. 800-812, May 2001.

[18] J. Hokfelt, O. Edfors and T. Maseng, "Interleaver design for turbo codes based on the performance of iterative decoding," in *Proc. IEEE Int. Conf. on Communications (ICC'99)*, vol. 1, pp. 93-97, Vancouver, June 1999.

[19] N. Kahale and R. Urbanke, "On the minimum distance of parallel and serially concatenated codes," 1997. Downloadable from http://lthcwww.epfl.ch/publications/

[20] S. Lin and D.J. Costello, Error Control Coding: Fundamentals and Applications, Prentice Hall, 2004.

[21] L.C. Perez, J. Seghers, D.J. Costello, "A distance spectrum interpretation of turbo codes," *IEEE Trans. on Inf. Theory*, vol. 42, no. 6, pp. 1698-1709, Nov. 1996.

[22] W.W. Peterson and E.J. Weldon, Jr., Error-correcting codes, The MIT Press, 2nd edition, 1972.

[23] O. Pothier, "Compound codes based on graphs and their iterative decoding," *PhD thesis in communications and electronics, ENST*, Paris, Jan. 2000. Downloadable from http://www.comelec.enst.fr/~boutros/coding

[24] J.L. Ramsey, "Realization of optimum interleavers," *IEEE Trans. Inform. Theory*, vol. IT-16, no. 3, May 1970.

[25] H.R. Sadjadpour, N.J.A. Sloane, M. Salehi and G. Nebe, "Interleaver design for turbo codes," *IEEE Journal on Sel. Areas in Communications*, vol. 19, no. 5, pp. 831-837, May 2001.

[26] R.P. Stanley, *Enumerative Combinatorics*, Monterey, California: Wadsworth & Brooks/Cole, 1986.

[27] J. Sun and O.Y. Takeshita, “Interleavers for turbo codes using permutation polynomials over integer rings,” *IEEE Trans. Inform. Theory*, vol. 51, no. 1, pp. 101-119, Jan 2005.

[28] R. M. Tanner, “On Quasi-Cyclic Repeat-Accumulate Codes,” Proc. of the 37th Annual Allerton Conference on Communication, Control, and Computing, Monticello, Illinois, Sept. 1999.

[29] R. M. Tanner, “Toward an algebraic theory for turbo codes,” in *Proc. of the 2nd International Symposium on Turbo Codes and Related Topics*, Brest, France, Sept. 2000.

[30] D. V. Truhachev, M. Lentmaier and K. S. Zigangirov, “Some results concerning design and decoding of turbo codes,” Problemy Peredachi Informatsii, vol. 37, No. 3, 2001, pp. 190–205.

[31] P.O. Vontobel, “On the construction of turbo code interleavers based on graphs with large girth,” in *Proc. IEEE Int. Conf. on Communications (ICC’02)*, vol. 3, pp. 1408-1412, New York, May 2002.

[32] J. Yu, M.-L. Boucheret and R. Vallet, “Design of turbo codes interleaver by loop distributions,” in *Proc. IEEE Int. Symp. Information Theory (ISIT’02)*, pp. 54, Lausanne, Jul. 2002.

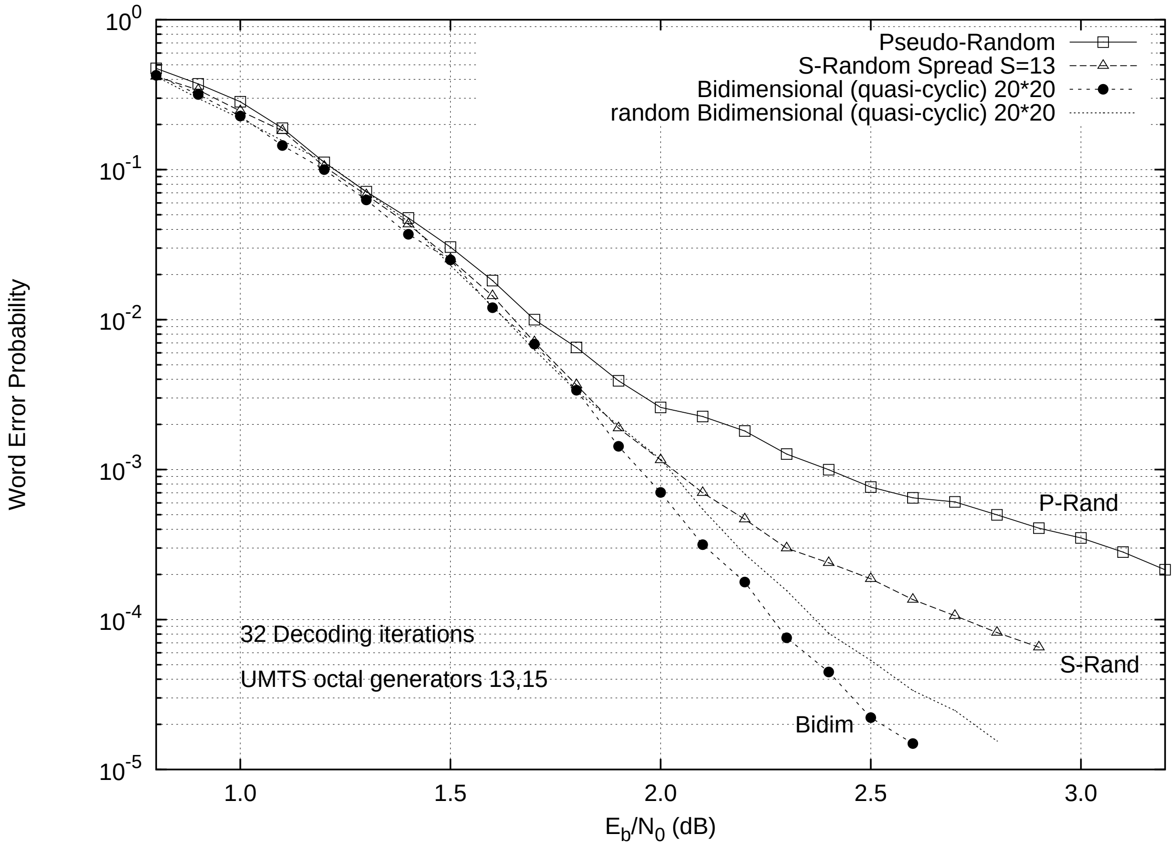

Figure 4: Performance of rate 1/2 turbo code for different interleavers of size 400 bits. Octal generators (13,15), coding rate is raised from 1/3 to 1/2 by puncturing parity bits, 32 decoding iterations, additive white gaussian noise channel, binary phase shift keying modulation. For all points drawn above, at least 100 block errors and 500 bit errors have been measured during Monte Carlo simulation to estimate the word error probability.

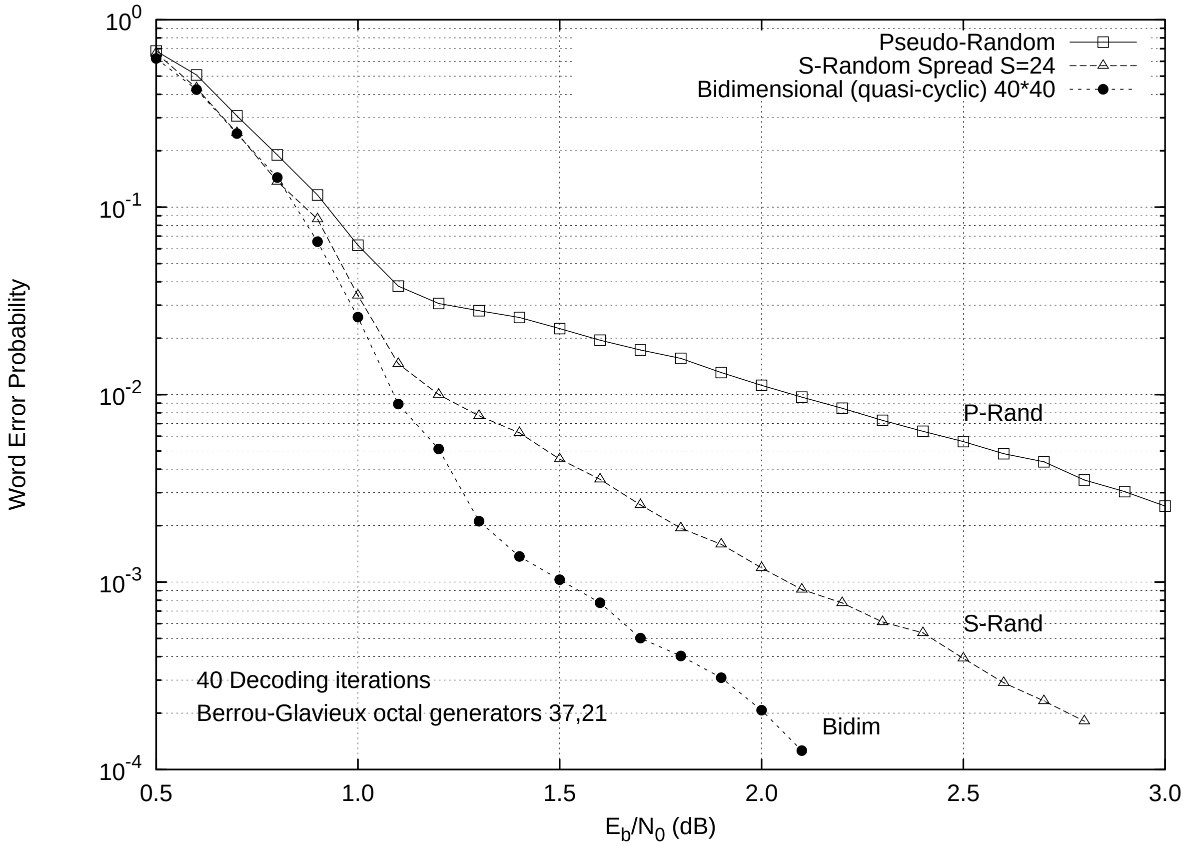

Figure 5: Performance of rate 1/2 turbo code for different interleavers of size 1600 bits. Octal generators (37,21), coding rate is raised from 1/3 to 1/2 by puncturing parity bits, 40 decoding iterations, additive white gaussian noise channel, binary phase shift keying modulation. For all points drawn above, at least 100 block errors and 500 bit errors have been measured during Monte Carlo simulation to estimate the word error probability.