Deterministic Interleavers for Turbo Codes with Random-like Performance and Simple Implementation

Carlos J. Corrada-Bravo

Department of Computer Science
University of Puerto Rico
San Juan, PR 00931
E-mail: ccorrada@cnnet.upr.edu

Ivelisse Rubio

Department of Mathematics
University of Puerto Rico
Humacao, PR 00791
E-mail: ive@cuhwww.upr.clu.edu

Abstract: *In this work we present a class of interleavers for turbo codes generated using monomials over finite fields that achieve as good or better performance than random interleavers while still very simple to implement. We have found a general case that outperforms the random interleaver for every size of the finite field tested. We also raise the question of which are the parameters needed to determine how good an interleaver is.*

Keywords: interleaver, deterministic interleaver, turbo codes, permutation polynomial, ease of implementation

1. INTRODUCTION

The interleaver plays a fundamental role in the performance of turbo codes. In the original implementation, a random interleaver is used and in [8] it was explained why this kind of interleaver worked and suggestions on desirable properties of a good interleaver for turbo codes where given. The parameters that are normally used are the dispersion and the spreading (see [2]). Reference [7] introduced a class of semi-random interleavers. These have been shown to have the best performance, whenever a suitable semi-random interleaver with the desired parameters exists. However, existence is not always guaranteed and even when they do exist one has to resort to computer searches to identify them.

The fact that these interleavers (random and s-random) are found by computer search implies that they have to be stored in memory. To avoid this problem, researchers have considered deterministic constructions that can be generated on the fly and that perform as well as random interleavers. Among the deterministic constructions is the construction in [5] which works whenever the block length is a power of 2 and the work presented here that works for block lengths equal to a prime number.

The permutations presented in this work not only do not have to be stored in memory but also the cycle length is equal to two (the permutation is its own inverse). This implies that the same implementation used at encoding can be used at decoding. Therefore, these interleavers are much easier to implement and, as it is shown in Figure 1, they perform as well as or better than random interleavers.

2. INTERLEAVERS

Consider $\mathbf{F}_q$, the finite field with q elements. It is well known that the function $\pi : \mathbf{F}_q \longrightarrow \mathbf{F}_q$ defined by $\pi(x) = x^i$ produces a permutation of the elements in $\mathbf{F}_q$ if and only if $gcd(i, q-1) = 1$. Results from previous simulations suggested that the relation of the length of the cycles of the permutation and the length of the cycle of the convolutional code influence the performance of the code. Because of this we are interested in permutations of $\mathbf{F}_q$ that decompose in cycles of the same length (excluding the fixed points) and are obtained using monomials x^i. These monomials have been characterized in [3] and [4].

2.1. Cyclic Decomposition of Permutations Given by Monomials

The following theorems characterize the exponents i such that x^i gives a permutation of $\mathbf{F}_q$ that decomposes in cycles of the same length j (excluding the fixed points). Let $ord_p(i)$ denote the order of i modulo p; this is, it is the smallest non-negative integer j such that $i^j \equiv 1 \ (mod\ p)$.

Theorem 1. *Let $q-1 = p_0^{k_0} p_1^{k_1} \cdots p_r^{k_r}$. The permutation of $\mathbf{F}_q$ given by x^i decomposes in cycles of length j or 1 if and only if one of the following holds for each $l = 0, \cdots, r$:*

1. $i \equiv 1 \ \left(mod\ p_l^{k_l}\right)$
2. $j = ord_{p_l^{k_l}}(i)$ *and* $j \mid (p_l - 1)$
3. $j = ord_{p_l^{k_l}}(i)$, $k_l \geq 2$ *and* $j = p_l$

The location of the fixed elements affect the parameters of the interleaver. For example, consecutive fixed points will give bad spreading. In the case where 0 and 1 are fixed by the permutation, we can always avoid this problem by considering the permutation of $\mathbf{F}_q^*$. This is why this is an important case to consider.

When $\{0, 1, -1\}$ or $\{0, 1\}$ are the only elements fixed by the permutation, the characterization of the monomials is the following:

Theorem 2. *Let $q-1 = p_0^{k_0} p_1^{k_1} \cdots p_r^{k_r}$, $p_0 = 2$, $k_0 = 0, 1$. The permutation of $\mathbf{F}_q$ given by x^i decomposes in cycles with the same length j and $\{0, 1, -1\}$ or $\{0, 1\}$ are the only fixed elements if and only if $j = ord_{p_l^{k_l}}(i)$ and $j|(p_l - 1)$ for every $p_l \neq 2$.*

In the case where $q - 1 = 2^k p_1^{k_1} \cdots p_r^{k_r}$ and $k \geq 2$ one has that there are only one or two monomials x^i that give permutations that decompose in cycles of the same length and have $\{0, 1, -1\}$ as the only fixed elements. Also, when this happens the length of the cycles is always 2. This implies that these permutations are their own inverse and hence the same permutation can be used for encoding and decoding.

Theorem 3. *Let $q - 1 = p_0^{k_0} p_1^{k_1} \cdots p_r^{k_r}$, where $p_0 = 2$, $k_0 \geq 2$. The permutation of $\mathbf{F}_q$ given by x^i decomposes in cycles of the same length j and $\{0, 1, -1\}$ are the only fixed elements if and only if $j = ord_{p_l^{k_l}}(i)$ for every $p_l \neq 2$, $j = ord_{2^h}(i)$ for $2 \leq h \leq k_0$, and $j = 2$.*

Corollary 1. *Let $q - 1 = p_0^{k_0} p_1^{k_1} \cdots p_r^{k_r}$, where $p_0 = 2$, $k_0 > 2$. The permutation of $\mathbf{F}_q$ given by x^i decomposes in cycles of the same length j and $\{0, 1, -1\}$ are the only fixed elements if and only if $j = 2$ and $i = q - 2$ or $i = \frac{q-3}{2}$.*

Corollary 2. *Let $q - 1 = 4p_1^{k_1} \cdots p_r^{k_r}$. Then the permutation given by $x^i : \mathbf{F}_q \to \mathbf{F}_q$ decomposes into cycles with the same length j and $\{0, 1, -1\}$ are the only elements fixed by x^i if and only if $j = 2$ and $i = q - 2$*

Using the above theorems we can construct the monomials x^i that give permutations of $\mathbf{F}_q$ that decompose in cycles of certain length j.

2.2. Dispersion of the Permutation Given by x^{p-2}

The reasons why some turbo codes perform better than others are not well understood, however one of the parameters associated to the performance is the dispersion of the interleaver. The *dispersion* of an interleaver π measure the "randomness" of the interleaver and it is defined as the number of elements in the set

$$D(\pi) = \{(j - i, \pi(j) - \pi(i)) \mid 1 \leq i < j \leq n\}.$$

The *normalized dispersion* is $\gamma = \frac{2|D(\pi)|}{n(n-1)}$, where n is the number of symbols in the sequence block. The closest to 1 that the normalized dispersion is, the better dispersion the interleaver has. For example random interleavers have dispersion around 0.8.

From now on, let p be a prime number. As we will see in the next section, in all the cases tested, permutations of $\mathbb{Z}_p$, obtained using x^{p-2}, where p is a prime, always perform as well or better than random interleaver. However, as we will see next, the dispersion for these interleavers is always close to 0.5 (see Section 3 for a discussion on the parameters). The following results will be used to prove the bounds on the dispersion of the permutations of $\mathbb{Z}_p$ obtained with monomials x^{p-2}.

The two ideas behind are that for each value of $s = i - j$, (1) the differences $\pi(i) - \pi(j)$ have different values for $j < \frac{p-s}{2}$, s odd and $j \leq \frac{p-s-1}{2}$ s even; and (2) we have the same set of differences for $j > \frac{p-s}{2}$ or $j > \frac{p-s-1}{2}$.

Proposition 1. *Let p be a prime, $s \in \{1, 2, \cdots, p-2\}$ and consider the following polynomial in $\mathbf{F}_p[x]$,*

$$d(x) = (a+s)^{p-2} - a^{p-2} - (a+x+s)^{p-2} + (a+x)^{p-2},$$

where a is such that $1 \leq a$, $a + s \leq p - 1$. Then $\alpha = 0$ and $\alpha = -2a - s$ are the only roots of $d(x)$ in $\mathbb{Z}_p$ such that $a + \alpha + s, a + \alpha \in \mathbb{Z}_p^*$.

Proof: Let $q(x) = d(x)(a+s)a(a+x+s)(a+x) \in \mathbb{Z}_p[x]$. Note that $q(x)$ and $d(x)$ have the same roots α satisfying that $a + \alpha + s$, $a + \alpha \in \mathbb{Z}_p^*$. Also, as a function $q(x) : \mathbb{Z}_p \longrightarrow \mathbb{Z}_p$,

$q(x) = (a+s)^{p-1}a(a+x+s)(a+x) - (a+s)a^{p-1}(a+x+s)(a+x) - (a+s)a(a+x+s)^{p-1}(a+x) + (a+s)a(a+x+s)(a+x)^{p-1} = a(a+x+s)(a+x) - (a+s)(a+x+s)(a+x) - (a+s)a(a+x) + (a+s)a(a+x+s) = (a+x+s)(a+x)(a-a-s) - (a+s)a(a+x-a-x-s) = -s\left[(a+x+s)(a+x) - (a+s)a\right] = -sx(x+2a+s).$

This implies that, for $\alpha \in \mathbb{Z}_p$, $d(\alpha) = 0$ if and only if $\alpha = 0$ or $\alpha = -2a - s$. □

Corollary 3. *Let p be a prime, $s \in \{1, 2, \cdots, p-2\}$ and consider $d(x) = (a + s)^{p-2} - a^{p-2} - (a + x + s)^{p-2} + (a + x)^{p-2} \in \mathbb{Z}_p[x]$, where a is such that $1 \leq a$, $a + s \leq p - 1$. Then $\alpha = 0$ is the only root of $d(x)$ in $\mathbb{Z}_p$ such that $a + \alpha + s, a + \alpha \in \mathbb{Z}_p^*$ if and only if $a = \frac{p-s}{2}$.*

Proof: From Proposition 1, one has that $d(x)$ has only one root if and only if $-2a - s = 0$. This happens if and only if $a = \frac{p-s}{2}$ □

Proposition 2. *Let $\pi(x) = x^{p-2} : \mathbb{Z}_p \longrightarrow \mathbb{Z}_p$, and fix $s \in \{1, \cdots, p-2\}$. Consider the finite sequence of differences $d_j = \pi(i) - \pi(j)$, where $1 \leq j < i \leq p - 1$ and $i - j = s$. Each difference in the sequence appears exactly twice if and only if s is even.*

Proof: Since $i - j = s$, one has that $d_j = \pi(i) -$

$\pi(j) = (j+s)^{p-2} - j^{p-2}$. If there is another difference d_k, such that $d_k = d_j$ and $j < k$, then $k = j + x$ for some $x \in \mathbb{Z}_p$ such that $1 \leq k \leq p-1$ and $k+s \leq p-1$. Also, one has that

$$d_k = (k+s)^{p-2} - k^{p-2} = (j+x+s)^{p-2} - (j+x)^{p-2},$$

and $(j+s)^{p-2} - j^{p-2} = (j+x+s)^{p-2} - (j+x)^{p-2}$. This means that the number of differences that are equal to d_j is the number of non-zero roots of the polynomial $p(x) = (j+s)^{p-2} - j^{p-2} - (j+x+s)^{p-2} + (j+x)^{p-2}$. From Proposition 1 and Corollary 3 we know that the polynomial has one root $\alpha \neq 0$ if and only if $j \neq \frac{p-s}{2}$.

If s is even, since p is odd and j is an integer, we have that $j \neq \frac{p-s}{2}$, and hence $d(x)$ has only one root $\alpha \neq 0$. This means that each difference appears exactly twice in the sequence.

If s is odd, then for $j = \frac{p-s}{2}$ the polynomial has $\alpha = 0$ as the only root and the difference $d_j, j = \frac{p-s}{2}$ appears only once. □

Corollary 4. *Let $\pi(x) = x^{p-2} : \mathbb{Z}_p \longrightarrow \mathbb{Z}_p$, and fix $s \in \{1, \cdots, p-2\}, s$ odd. Consider the sequence of differences from the previous proposition. The difference d_j appears in the sequence exactly once if and only if $j = \frac{p-s}{2}$. Otherwise, the difference appears exactly twice.*

Proposition 3. *Let $\pi(x) = x^{p-2} : \mathbb{Z}_p \longrightarrow \mathbb{Z}_p$, and fix $s \in \{1, \cdots, p-2\}$. Consider $D_s := \{d_j = \pi(i) - \pi(j) \mid 1 \leq j < i \leq p-1,\ i-j = s\}$. Then*

$$|D_s| = \begin{cases} \frac{p-s}{2} & : \ s \text{ odd} \\ \frac{p-s-1}{2} & : \ s \text{ even} \end{cases}$$

Proof: The number of elements in the sequence of differences $d_j = \pi(i) - \pi(j)$ is the number of possible j such that $s+j = i \leq p-1$. This is, $1 \leq j \leq p-1-s$, hence the number of possible differences is $p-s-1$.

From Proposition 2, if s is even, each difference appears exactly twice in the sequence and therefore $|D_s| = \frac{p-1-s}{2}$.

If s is odd, by Corollary 4 each difference d_j appears exactly twice in the sequence, except for the case where $j = \frac{p-s}{2}$, which appears once. Hence, $|D_s| = \frac{p-s-2}{2} + 1 = \frac{p-s}{2}$. □

Theorem 4. *Consider the interleaver $\pi(x) = x^{p-2} : \mathbb{Z}_p \longrightarrow \mathbb{Z}_p$. Let $D(\pi) = \{(j-i, \pi(j) - \pi(i)) \mid 1 \leq i < j \leq p\}$. Then, π has normalized dispersion γ, where $\frac{p-1}{2p} \leq \gamma \leq \frac{p+3}{2p}$.*

Proof: The normalized dispersion of π is $\gamma = \frac{2|D(\pi)|}{p(p-1)}$. The number of elements in $D(\pi)$ is $|D(\pi)| = \sum_{s=1}^{p-1} |D_{s'}|$, $D_{s'} := \{d_j = \pi(i) - \pi(j) \mid 1 \leq j < i \leq p,\ i-j = s\}$.

Note that $D_{s'} = D_s \cup \{\pi(p) - \pi(j) \mid p - j = s\}$ and hence $|D_s| \leq |D_{s'}| \leq |D_s| + 1$. From this one has that

$$\sum_{s=1}^{p-1} |D_s| \leq |D(\pi)| \leq \sum_{s=1}^{p-1} (|D_s| + 1) = \sum_{s=1}^{p-1} |D_s| + (p-1).$$

By Proposition 3,

$$\begin{aligned} \sum_{s=1}^{p-1} |D_s| &= \sum_{\substack{s=1 \\ s \text{ odd}}}^{p-2} \frac{p-s}{2} + \sum_{\substack{s=2 \\ s \text{ even}}}^{p-1} \frac{p-s-1}{2} \\ &= \frac{p-1}{2} + 2 \sum_{s=3}^{p-2} \frac{p-s}{2} \\ &= \frac{p-1}{2} + 2\left(1 + 2 + \cdots + \frac{p-3}{2}\right) \\ &= \frac{p-1}{2} + (\frac{p-3}{2})(\frac{p-3}{2} + 1) \\ &= \frac{(p-1)^2}{4}. \end{aligned}$$

Hence,

$$\frac{\frac{2(p-1)^2}{4}}{p(p-1)} \leq \gamma \leq \frac{2\left(\frac{(p-1)^2}{4} + (p-1)\right)}{p(p-1)},$$

and the result follows from this. □

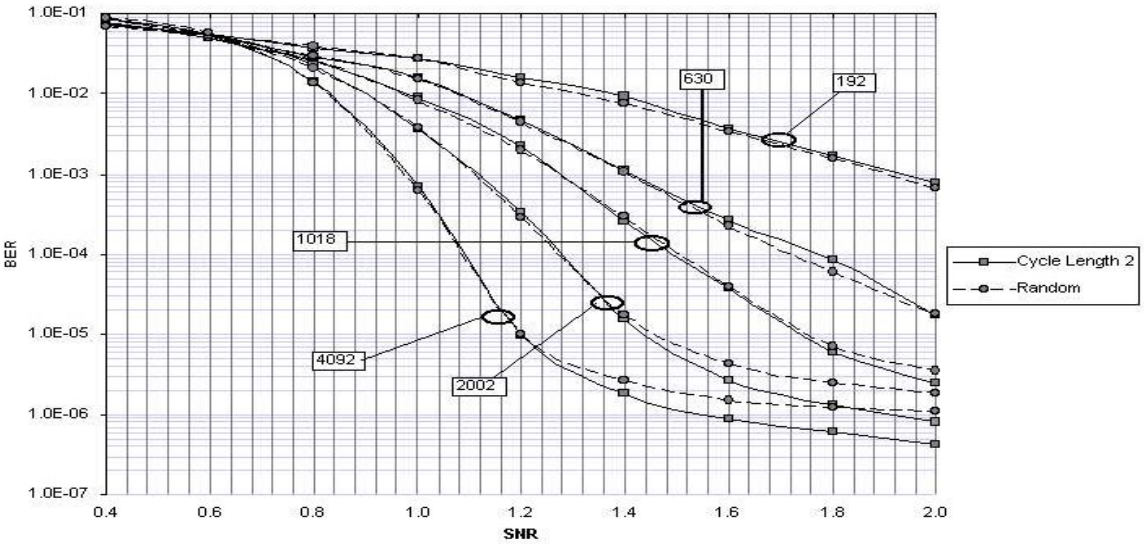
3. SIMULATION RESULTS AND FUTURE WORK

The following results were obtained using convolutional codes with transfer function $\frac{1+D+D^2+D^3}{1+D^2+D^3}$ and the specified interleaver. There is no puncturing, hence we obtain a turbo code with rate= 1/3. We have found that for all primes p tested, the interleaver generated with x^{p-2} performs as well or better than a random interleaver[1]. This can be seen in Figure 1, where we have random interleavers with dispersion around 0.8 and spreading equal to 1 and the interleavers generated by the monomial x^{p-2} that have dispersion around 0.5 and spreading equal to 1.

Further questions have resulted from this work. Our simulations show that although our interleavers do not have better dispersion or spreading than random interleavers, they perform as well or better (see Table 1 and Figure 1) than them. Another example are the Welch-Costas interleavers, which have perfect dispersion of $\gamma = 1$ but do not perform better than a random interleaver.

[1]We have found cases where other choices of x^i are better for a specific block lengths. However, the case of x^{p-2} always performs well.

Figure 1: BER of random interleavers and from the monomial x^{p-2} with cycle length of two.



Graphs with large girth have been used for the construction of regular and irregular low density parity check (LDPC) codes and recently, in [9] the author derived interleavers for turbo codes from graphs which have large girth. The *girth* (the length of the shortest cycle) of the turbo code graph, capture the relation between the cycle length of interleavers and the cycle length of the convolutional codes. We are carrying further studies in this relation in an attempt to answer the question as to which other parameters are necessary to established how an interleaver is going to perform. With this approach we could be able to predict the performance of a turbo code with a particular interleaver, based on the cycle length of the convolutional code and the cycle structure of the interleaver. Hence, removing of the analysis (up to a degree) the painstaking and time consuming task of simulation.

4. IMPLEMENTATION

As we have mentioned before there are various characteristics that makes our interleavers easier to implement.

- They do not have to be stored in memory
  - They are generated from finite fields and therefore technology like shift registers can be used to generate them on the fly.
  - Another option is to hard wired them. Since their cycles are of length two a cross over for each pair will suffice.
- They are its own inverse and therefore the machinery used for encoding can be used for decoding.
- Since the case where $i = p - 2$ works for every p an adaptive element can be build to increase or decrease block length according to the channel.

REFERENCES

[1] C. J. Corrada Bravo and P. V. Kumar, "Permutation Polynomials for Interleavers in Turbo Codes", *2003 IEEE ISIT-2003*, Yokohama, Japan, June 2003.

[2] C. Heegard, S. Wicker, *Turbo Codes*, Kluwer Academic Publishers, 1999.

[3] I. Rubio, "Cyclic Decomposition of Monomial Permutations", M.S. Thesis, University of Puerto Rico, December, 1988.

[4] I. Rubio and C. Corrada-Bravo, "Cyclic Decomposition of Permutations of Finite Fields Obtained Using Monomials and Applications to Turbo Codes", to be submitted to the Proceedings of Finite Fields and Applications Symposium, May 2003.

[5] O. Takeshita and D. Costello "New deterministic interleaver designs for turbo Codes", *IEEE-IT*, Vol. 46, pp. 1988-2006, Sept. 2000.

[6] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near shannon limit error-correcting coding and decoding: Turbo-codes," in *Proc. of ICC'93*, Geneva, Switzerland, May 1993, pp. 1064–1070.

[7] S. Dolinar and D. Divsalar, "Weight distributions of turbo codes using random and nonrandom interleavers," Tech. Rep. 42-122, JPL, Pasadena, CA, Aug 1995.

[8] S. Benedetto and G. Montorsi, "Design of parallel concatenated convolutional codes," *IEEE-IT*, vol. 44, pp. 591–600, May 1996.

[9] P. O. Vontobel, "On the Construction of Turbo Code Interleavers Based on Graphs with Large Girth", *Proc. IEEE Intern. Conf. Communications*, Vol.3, pp.1408-1412, New York, NY, USA, Apr. 28-May 2, 2002.