Algebraic Construction of Interleavers Using Permutation Monomials

Carlos J. Corrada Bravo
Department of Computer Science
University of Puerto Rico
San Juan, PR 00931
email: ccorrada@cnnet.upr.edu

Ivelisse Rubio
Department of Mathematics
University of Puerto Rico
Humacao, PR 00791
email: ive@cuhwww.upr.clu.edu

***Abstract*— We present an algebraic construction for interleavers of length p^r, where p is any prime. These interleavers are very simple to implement and have performance better than random interleavers and other known algebraic constructions. We construct a permutation of $\mathbb{Z}_{p^r}$ using permutations of the elements of the finite field $\mathbf{F}_{p^r}$ given by monomials over the field.**

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

I. Introduction

The interleaver plays a fundamental role in the performance of turbo codes. The actual standard for turbo codes uses interleavers that are generated randomly. A class of S-random interleavers was introduced in [8]. These have shown to have the best performance, whenever a suitable S-random interleaver with the desired parameters exists.

One of the major draw backs of these type of interleavers (random and S-random) is that they have to be found by computer searches and have to be stored in memory, which implies a more complex implementation. Another problem with S-random interleavers is that their existence is not always guaranteed. To avoid these problems, researchers have considered deterministic constructions that can be generated on the fly, could be analyzed a priori and perform as well as random interleavers. Among the deterministic constructions are the construction in [6] which works whenever the block length is a power of 2, the constructions in [11] and the work presented here that works for block lengths equal to powers of any prime number.

The algebraic construction for interleavers of length p^r presented here is very easy to implement and the permutations do not have to be stored in memory. In addition, a class of these permutations are their own inverse, which implies that the same implementation used at encoding can be used at decoding. Turbo codes constructed with these interleavers are much easier to implement and, as it is shown in Figures 1, 3, they perform as well as or better than turbo codes constructed with random and quadratic residue interleavers and are not too far away from the performance of S-random interleavers. Our construction still have to be compared with dithered relative prime (DRP) interleavers.

II. Permutations of $\mathbb{Z}_{p^r}$

Let p be any prime number and $\mathbf{F}_{p^r}$ be the finite field with p^r elements. In this section we will show how to construct permutations of $\mathbb{Z}_{p^r}$ using permutations of $\mathbf{F}_{p^r}$ obtained using monomials. We will order the elements of $\mathbf{F}_{p^r}$ considering its construction as a vector space over $\mathbb{Z}_p$.

It is well known that the multiplicative group $\mathbf{F}_{p^r}^*$ of a finite field $\mathbf{F}_{p^r}$ is cyclic. This means that there exists an element $\alpha \in \mathbf{F}_{p^r}^*$ that generates $\mathbf{F}_{p^r}^*$. This element is called a *primitive element* of $\mathbf{F}_{p^r}$.

Let $g(x) \in \mathbf{F}_{p^r}[x]$ be a monic, irreducible polynomial of degree r that has α, a primitive element of $\mathbf{F}_{p^r}$, as a root. Then we can represent $\mathbf{F}_{p^r}$ as a vector space over $\mathbb{Z}_p$ with basis $\left\{1, \alpha, \alpha^2, \cdots, \alpha^{r-1}\right\}$ or as powers of the primitive element α, $\mathbf{F}_{p^r} = \left\{0, \alpha, \alpha^2, \cdots, \alpha^{p^r-1}\right\}$. The later representation is useful for computations involving multiplications.

We can order the elements of the finite field considering the "base p" number obtained from the scalars of the vector space representation of the element and changing it to base 10. More formally, the ordered set $\{0, \xi_0, \xi_1, \cdots, \xi_{p^r-1}\} = \mathbf{F}_{p^r}$ is such that

$$\xi_n = n_0 + n_1\alpha + n_2\alpha^2 + \cdots + n_{r-1}\alpha^{r-1} \qquad (1)$$

$$n = n_0 + n_1 p + n_2 p^2 + \cdots + n_{r-1}p^{r-1}, \quad 0 \le n_i < p.$$

It is easy to check that this construction gives a bijection between $\mathbb{Z}_{p^r} = \{0, 1, \cdots, p^r - 1\}$ and $\mathbf{F}_{p^r}$. Using this bijection we can construct permutations of $\mathbb{Z}_{p^r}$ from permutations of $\mathbf{F}_{p^r}$. Note that n is the base 10 representation of the base p number $(n_0 n_1 \cdots n_{r-1})_p$. This ordering can be easily generated by the add and carry operation which makes these permutations very easy to implement.

Example 1: Consider $\mathbf{F}_3 = \mathbb{Z}_3$ and let α be a root of the polynomial x^2+x+2. We have that α is a primitive element of $\mathbf{F}_{3^2}$. The following table shows two representations of the non-zero elements of $\mathbf{F}_9$. The first row is the index row used for the ordering; the second row is the corresponding vector space representation of the element of $\mathbf{F}_9$; and the third row has the representation as powers of the primitive element α. Note that, since α is a root of $x^2 + x + 2$, we have that $\alpha^2 = 2\alpha + 1$.

n	1	2	3	4	5	6	7	8
ξ_n	1	2	α	$1+\alpha$	$2+\alpha$	2α	$1+2\alpha$	$2+2\alpha$
α^j	α^0	α^4	α	α^7	α^6	α^5	α^2	α^3

Let $q = p^r$. The monomial function $f : \mathbf{F}_q \longrightarrow \mathbf{F}_q$ defined by $f(x) = x^i$ produces a permutation of the elements in $\mathbf{F}_q$ if and only if $gcd(i, q-1) = 1$. This type of monomials are called *permutation monomials*.

In [1] the performance of turbo codes with interleavers constructed with monomials x^{q-2}, $q = p$ was compared with the performance of turbo codes with random interleavers. Simulations showed that this class of *monomial interleavers* outperform the random interleaver. For the case $q = p$, we evaluated the permutation monomial in the (naturally ordered) elements of $\mathbb{Z}_p$. In this paper, we use the above ordering of $\mathbf{F}_{p^r}$ and permutation monomials to construct permutations of $\mathbb{Z}_{p^r}$ (and hence interleavers of length p^r) using the following results.

Theorem 1: Let $\mathbf{F}_{p^r} = \{0, \xi_0, \xi_1, \cdots, \xi_{p^r-1}\}$ be defined as in (1). Then the function $\pi : \mathbb{Z}_{p^r} \longrightarrow \mathbb{Z}_{p^r}$ defined by $\pi(n) = m$, where $f(\xi_n) = \xi_m$, is a permutation of $\mathbb{Z}_{p^r}$ if and only if $f : \mathbf{F}_{p^r} \longrightarrow \mathbf{F}_{p^r}$ is a permutation of $\mathbf{F}_{p^r}$.

Corollary 1: The function $\pi : \mathbb{Z}_{p^r} \longrightarrow \mathbb{Z}_{p^r}$ defined by $\pi(n) = m$, where $f(\xi_n) = (\xi_n)^i = \xi_m$, is a permutation of $\mathbb{Z}_{p^r}$ if and only if $gcd(i, p^r - 1) = 1$.

Example 2: Consider the ordering of the elements of $\mathbf{F}_9$ given in example 1:

$$\left(0, \alpha^0, \alpha^4, \alpha^1, \alpha^7, \alpha^6, \alpha^5, \alpha^2, \alpha^3\right) = (0, \xi_1, \xi_2, \cdots, \xi_8)\,.$$

Since $gcd(3, 9-1) = 1$, we have that $x^3 : \mathbf{F}_9 \longrightarrow \mathbf{F}_9$ gives the following permutation of $\mathbf{F}_9$:

$$\begin{aligned}&\left(0, \alpha^0, \alpha^4, \alpha^3, \alpha^5, \alpha^2, \alpha^7, \alpha^6, \alpha^1\right)\\ &= (0, \xi_1, \xi_2, \xi_8, \xi_6, \xi_7, \xi_4, \xi_5, \xi_3)\,.\end{aligned}$$

The function $\pi : \mathbb{Z}_9 \longrightarrow \mathbb{Z}_9$ defined by $\pi(n) = m$, where $(\xi_n)^3 = \xi_m$ gives the permutation of $\mathbb{Z}_9$:

$$(0, 1, 2, 8, 6, 7, 4, 5, 3).$$

A. Interleavers with block length p^r

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 close to 0.8.

For the case where $q = p$, a prime, we proved in [1] the following theorem that gives bounds for the dispersion of monomial interleavers of length p constructed with monomials x^{p-2}.

Theorem 2: Let p be a prime and consider the interleaver $\pi(x) = x^{p-2} : \mathbb{Z}_p \longrightarrow \mathbb{Z}_p$. Then, π has normalized dispersion γ, where $\frac{p-1}{2p} \leq \gamma \leq \frac{p+3}{2p}$.

The *spreading* of an interleaver meassures the distance between interleaved symbols that were "close" to each other before interleaving. More formally, an interleaver π is said to have *spreading factors* (s, t) if

$$|i - j| < s \Longrightarrow |\pi(i) - \pi(j)| \geq t.$$

The *s-parameter (spreading)* of an interleaver is the maximum value for s such that $s \leq t$.

Turbo codes constructed with the class of monomial interleavers x^{p-2} in Theorem 2 with dispersion close to 0.5 and spreading 1 performed better than random interleavers with dispersion close to 0.8 and spreading 1. We have not found specific bounds for the dispersion of monomial interleavers with block length $q = p^r$, $r \neq 1$ but our examples suggest that, for monomials of the form x^{q-2}, the dispersion is always close to 0.8. As we see in Figure 3, these interleavers perform better than random and quadratic interleavers.

Another advantage of the monomial interleavers x^{q-2} is that this permutation is its own inverse. This implies that the same implementation used for encoding can be used for decoding.

We are studying the relation between the cycle length of the interleavers and the cycle length of the convolutional code. This is captured in the *girth* of the turbo code graph, which is the length of the shortest cycle of the graph. In [10], turbo codes from graphs with large girth were studied. In [5], we characterized monomial permutations with cycles of the same length. We are using these monomials to study further the relation of the cycle length of the interleaver and the cycle length of the graph and their effect on the performance of the code, hoping to be able to analyze the performance of the turbo code a priori.

III. Simulation Results

Convolutional codes with transfer function $\frac{1+D+D^2+D^3}{1+D^2+D^3}$ were used to compare the performance of different interleavers with different block lengths. There was no puncturing, hence we obtain a turbo code with rate$= 1/3$. The spreading and the dispersion of the monomial interleavers as well as other interleavers are in Tables I and II.

We have found that for all cases where $q = p^r$, p a prime, the monomial interleaver x^{q-2} performs as well or better than a random interleaver. We have found cases where other choices of x^i are better for a specific block lengths. However, the case of x^{q-2} always performs well. This can be seen in Figures 1 and 3. In Figure 1 we compare random interleavers with dispersion close to 0.8 and spreading equal to 1 to interleavers generated by the monomial x^{p-2}, p a prime with dispersion close to 0.5 and spreading equal to 1. In Figure 3 we compare the monomial interleaver x^{q-2} with dispersion close to 0.8 and spreading equal to 1, to various classes of interleavers.

IV. Conclusion and remarks

We presented algebraic constructions for interleavers with block length $q = p^r$ which are very easy to implement. The design uses permutation monomials x^i defined over finite fields $\mathbf{F}_q$.

Our simulations show that monomial interleavers constructed with monomials of the form x^{q-2} always perform

Fig. 1. BER of random interleavers and interleavers from the monomial x^{p-2} with cycle length of two for various block lengths p

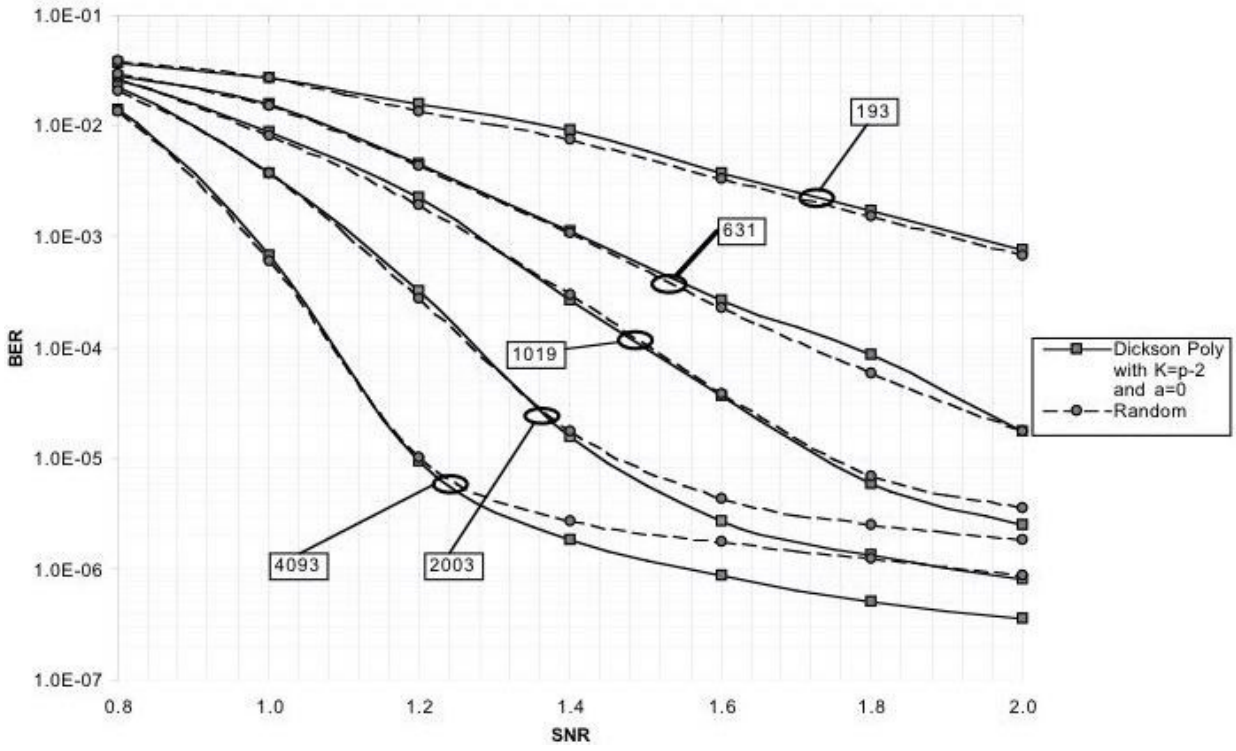

Fig. 2. PER of random interleavers and interleavers from the monomial x^{p-2} with cycle length of two for various block lengths p

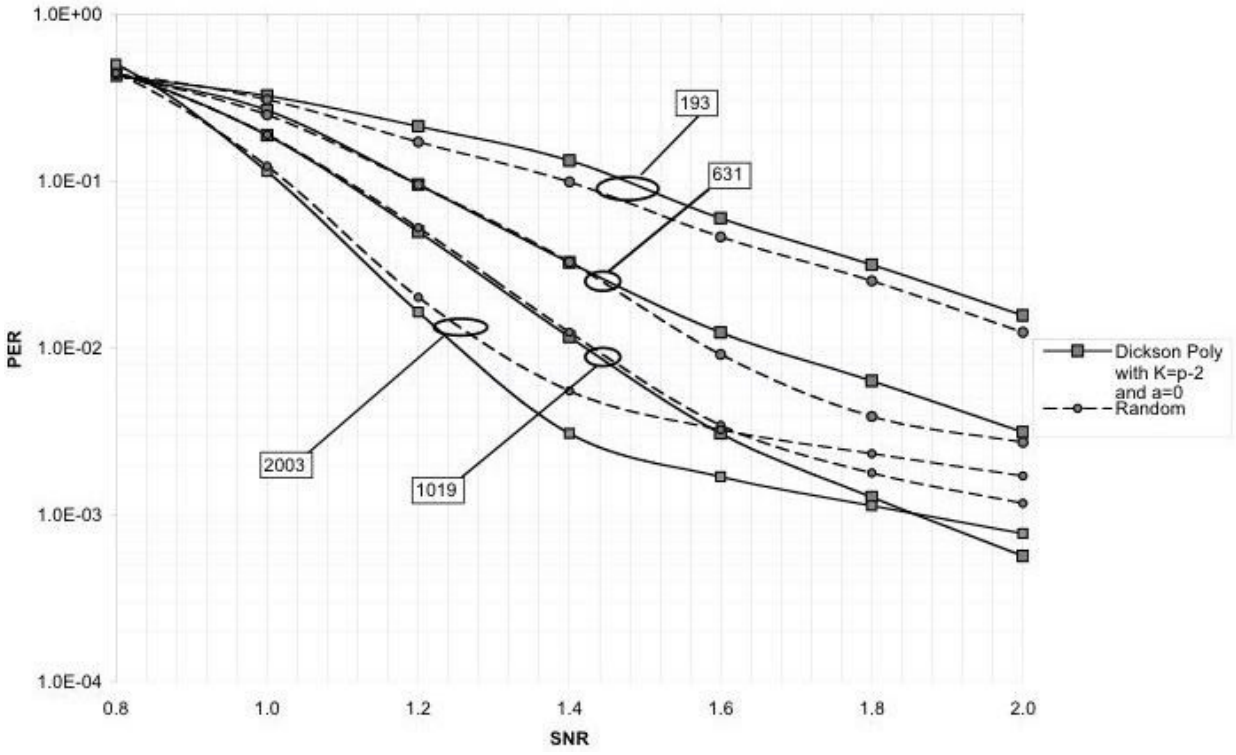

Fig. 3. BER of various interleavers and the interleaver from the monomial x^{q-2} with cycle length of two and block length 1024.

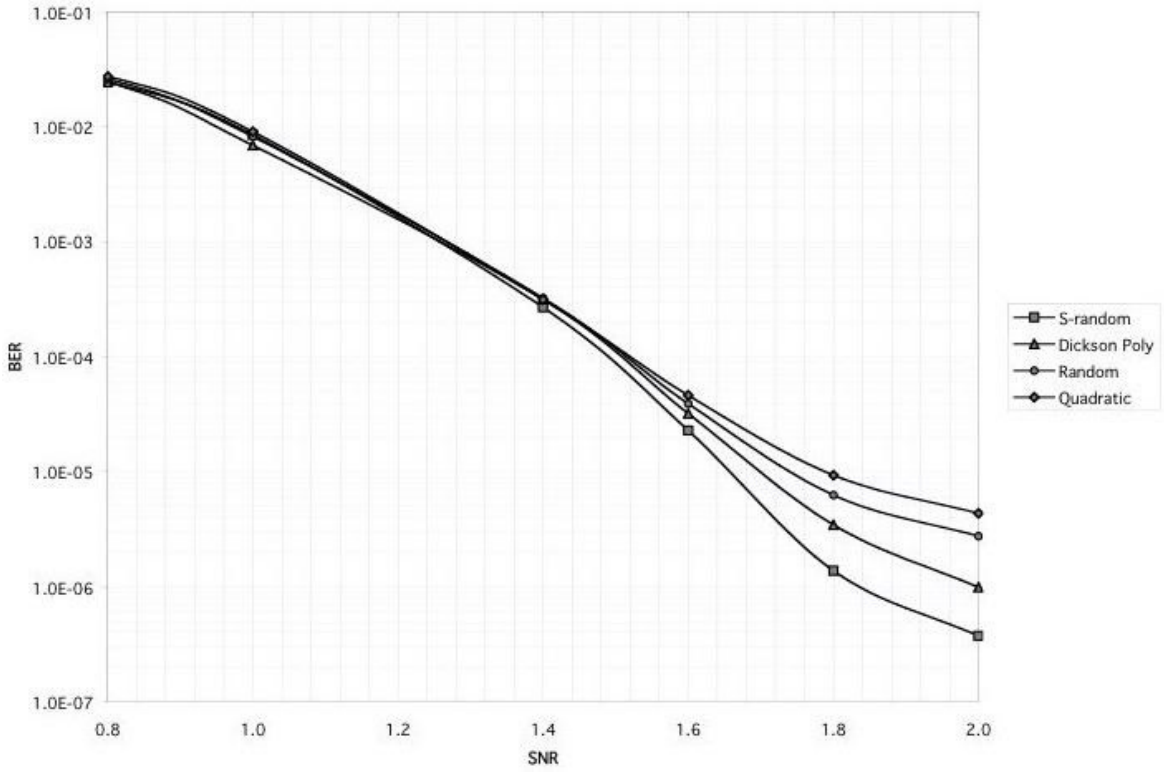

Fig. 4. PER of various interleavers and the interleaver from the monomial x^{q-2} with cycle length of two and block length 1024.

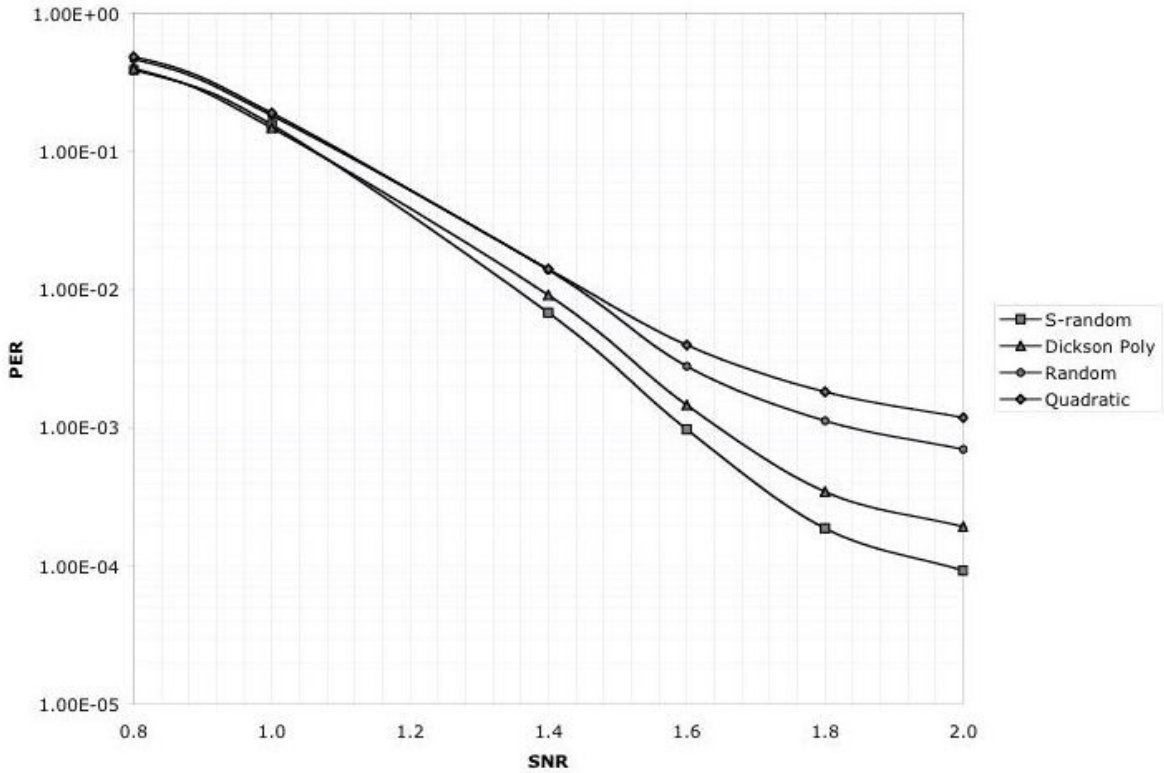

TABLE I

DISPERSION AND SPREADING OF RANDOM INTERLEAVERS AND FROM THE MONOMIAL x^{p-2} WITH CYCLE LENGTH OF TWO WITH VARIOUS BLOCK LENGTH.

Interleaver	Block length	Dispersion	Spreading
Random	193	0.813	1
Monomial	193	0.511	1
Random	631	0.814	1
Monomial	631	0.497	1
Random	1019	0.81	1
Monomial	1019	0.5	1
Random	2003	0.814	1
Monomial	2003	0.511	1
Random	4093	0.815	1
Monomial	4093	0.51	1

TABLE II

DISPERSION AND SPREADING OF VARIOUS INTERLEAVERS WITH BLOCK LENGTH 1024.

Interleaver	Dispersion	Spreading
S-Random	0.813	15
Random	0.814	1
Quadratic	0.74	1
Monomial	0.814	1

as well or better than random interleavers, even when the dispersion or spreading of the monomial interleaver is not better than the other interleavers (see Table I, and Figure 1). S-random interleavers still perform better than monomial interleavers but the implementation advantages should compensate for the difference in performance. Still, as future work our construction have to be compared with the dithered relative prime (DRP) interleavers.

Some of the advantages of the monomial interleavers are:

- 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.
- The permutation of $\mathbb{Z}_q$ obtained with x^{q-2} is its own inverse. Another option for constructing the permutations in the case of the monomials x^{q-2} is to hard wired them. Since the cycles of the permutation are of length two a cross over for each pair will suffice.
- Since the permutation from x^{q-2} is its own inverse, the machinery used for encoding can be used for decoding.
- Since monomial interleavers from x^{q-2} perform well for every $q = p^r$, an adaptive element can be build to increase or decrease block length according to the channel as in the case in CDMA2000 and 3GPP.

REFERENCES

[1] C. J. Corrada-Bravo and I. Rubio, "Deterministic Interleavers for Turbo Codes with Random-like Performance and Simple Implementation", *Proceedings of the 3rd International Symposium on Turbo Codes*, Sept. 2003.

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

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

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

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

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

[7] 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.

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

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

[10] 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.

[11] S. Crozier, J. Lodge, P. Guinand, and A. Hunt, "Performance of Turbo Codes with Relative Prime and Golden Interleaving Strategies", *Sixth International Mobile Satellite Conference (IMSC'99)*, Otawa, Canada, pp 268-275, June 16-18, 1999.