Int. J. Electron. Commun. (AEÜ) 69 (2015) 1550–1556

Contents lists available at ScienceDirect

International Journal of Electronics and Communications (AEÜ)

journal homepage: www.elsevier.com/locate/aeue

SHORT COMMUNICATION

The limitation of permutation polynomial interleavers for turbo codes and a scheme for dithering permutation polynomials

Jonghoon Ryu [a], Lucian Trifina [b,*], Horia Balta [c]

[a] Samsung Electronics, Inc., Suwon, Republic of Korea
[b] "Gheorghe Asachi" Technical University, Faculty of Electronics, Telecommunications and Information Technology, Department of Telecommunications, Bd. Carol I, No. 11 A, 700506 Iasi, Romania
[c] University Politehnica of Timisoara, Faculty of Electronics and Telecommunications, Department of Telecommunications, V. Parvan 2, 300223 Timisoara, Romania

ARTICLE INFO

Article history:
Received 5 February 2014
Accepted 16 June 2015

Keywords:
Permutation polynomial
Quadratic permutation polynomial
Turbo codes
Interleaver

ABSTRACT

In this letter, partial upper bounds on minimum distance for turbo codes with permutation polynomial (PP) based interleavers over integer rings are derived using the fact that PPs are equivalent to a family of linear permutation polynomials (LPPs). It is shown that upper bounds on minimum distance of turbo codes using higher order PP based interleavers are bounded by a function of the number of equivalent LPPs for PPs. Besides, it is shown that when the constant terms of LPPs are dithered, the resulting dithered LPP interleavers perform better than the quadratic permutation polynomial (QPP) based interleavers used in long term evolution (LTE) standard or than other good QPP or cubic permutation polynomial (CPP) based interleavers given in the literature.

1. Introduction

Permutation polynomial (PP) based interleavers over integer rings have been widely studied [1–3,6–8]. In particular quadratic permutation polynomial (QPP) based interleavers were emphasized due to their simple implementation [3] as well as excellent performance [1]. In [1], upper bounds on minimum distance of turbo codes with QPP based interleavers are shown.

Higher order PP based interleavers have also been investigated for better performance and implementation, in particular for cubic permutation polynomial (CPP) based interleavers [2,3]. However little is known for minimum distance of turbo codes with higher order PP based interleavers. In this letter, the technique shown in [3] is used to decompose higher order PPs into linear permutation polynomials (LPPs) and partial upper bounds on the minimum distance for turbo codes using higher order PP based interleavers are shown.

It is also shown that when the constant terms of the LPPs which are equivalent to PPs are dithered, better frame error rate (FER) performance is obtained.

For a more succinct writing, in the following, PP based interleavers are denoted as PP.

2. LPP representation of higher order PPs

In this section, previous results on higher order PPs are briefly reviewed and upper bounds on the minimum distance for turbo codes using PPs are shown. Firstly, the equivalence of PPs and a family of LPPs is shown. In the following, a parallel LPP (PLPP) is defined.

Definition 2.1. [3] Let $p(x)$ be an interleaver such that

$$p(x) = \begin{cases} p_0(x) = P_{1,0}x + P_{0,0}, & \mathrm{mod}(x, L) = 0 \\ p_1(x) = P_{1,1}x + P_{0,1}, & \mathrm{mod}(x, L) = 1 \\ \cdots \\ p_{L-1}(x) = P_{1,L-1}x + P_{0,L-1}, & \mathrm{mod}(x, L) = L-1, \end{cases}$$

which can be also represented in the following form,

$$p(x) = \begin{cases} p_0(y) = P_{1,0} \cdot Ly + P\prime_{0,0}, & x = Ly \\ p_1(y) = P_{1,1} \cdot Ly + P\prime_{0,1}, & x = Ly + 1 \\ \cdots \\ p_{L-1}(y) = P_{1,L-1} \cdot Ly + P\prime_{0,L-1}, & x = Ly + (L-1), \end{cases}$$

with $1 \le L < N$, where N is the interleaver length, $L|N$ and $0 \le y \le \frac{N}{L} - 1$. Then $p(x)$ is called a PLPP (i.e., $p(x)$ consists of L LPPs).

For each $l = 0, 1, \cdots, L-1$, $p_l(y)$ is a LPP and since a LPP can be implemented using only additions and comparisons, a PLPP can also

* Corresponding author. Tel.: +40 232701679.
E-mail addresses: jonghoon.ryu@samsung.com (J. Ryu), luciant@etti.tuiasi.ro (L. Trifina), horia.balta@upt.ro (H. Balta).

http://dx.doi.org/10.1016/j.aeue.2015.06.007
1434-8411/

Table 1
The least numbers of LPPs for equivalent PLPPs (L) of LTE-QPPs of various lengths N [5].

Lengths range	$L=1$	$L=2$	$L=3$	$L=4$	$L>4$	
40–512	1	55	3		1	60
528–1024		21	3	6	2	32
1056–2048		1	1	26	4	32
2112–6144			1	48	15	64
	1	77	8	80	22	188

be implemented using the same address generation method for a LPP [3].

Although not all PPs are equivalent to PLPPs [3], the following lemma shows that all PPs are equivalent to PLPPs when the interleaver lengths are of the form $N=2^3\cdot M$, where M is a positive integer.

Lemma 2.2. [3] *Let $f(x)=\sum_{k=1}^{K} f_k x^k (\mathrm{mod}\ N)$ be a PP. Suppose that $N=2^3\cdot M$, with M a positive integer. Then $f(x)$ is equivalent to a PLPP and $L\le 2M$.*

Lemma 2.2 is obtained by computing the $f(x)$ at each point $x=2My+l$ and using the modulo operation and the zero polynomials shown in [4,3] to remove quadratic and higher order terms. In particular, a sufficient condition for a QPP to have an equivalent PLPP is given in [3]. By the lemma 2.2, all the LTE-QPPs are equivalent to PLPPs, since their interleaver lengths are multiples of 8. In practice, L's are relatively small numbers compared to interleaver lengths, as shown in Table 1.

For example, let $f(x)=15x+32x^2(\mathrm{mod}256)$, then $f(2y)=15\cdot 2y+32\cdot(2y)^2=15\cdot 2y+32\cdot(2y)^2+256/2\cdot y+256/2\cdot(y)^2=79\cdot 2y$, $f(2y+1)=15\cdot(2y+1)+32\cdot(2y+1)^2=15\cdot(2y+1)+32$.

Note that $(N/2)\cdot y+(N/2)\cdot y^2$ is a zero polynomial for all y. Thus, the equivalent PLPP, with $L=2$, of the previous QPP $f(x)$, is:

$$p(x)=\begin{cases} p_0(x)=79x, & \mathrm{mod}(x,2)=0\\ p_1(x)=15x+32, & \mathrm{mod}(x,2)=1\end{cases}$$

Let $L=4$, then by using a similar method, the equivalent PLPP of the previous QPP $f(x)$, is:

$$p(x)=\begin{cases} p_0(x)=15x, & \mathrm{mod}(x,4)=0\\ p_1(x)=15x+32, & \mathrm{mod}(x,4)=1\\ p_2(x)=15x+128, & \mathrm{mod}(x,4)=2\\ p_3(x)=15x+32, & \mathrm{mod}(x,4)=3\end{cases}$$

Since L is relatively small compared to the interleaver length for QPPs in [5], the interleaver/deinterleaver for the PPs can be efficiently generated as shown in [3]. Note that the number of coefficients of the L LPPs depends only on L, not on the degree of PP. Thus a PP of arbitrary degree can be implemented using L LPPs if it is equivalent to L LPPs.

In the following, upper bounds on minimum distance for turbo codes with PPs using Lemma 2.2 are shown.

Lemma 2.3. *Let the first coefficients of PLPP be equal for all l, i.e., $P_{1,0}=P_{1,1}=\ldots=P_{1,L-1}=P$. Let also m and n be positive integers and $L|(m\cdot(2^\nu-1))$, where ν is the degree of the primitive feedback and monic feedforward polynomials of recursive systematic convolutional codes, which are component codes of a conventional turbo code. If there exists a critical interleaver pattern of size 4 as shown in Fig. 1, the minimum distance of the turbo code with this PLPP interleaver is upper bounded by $(m+n)\cdot 2^\nu+12$.*

Proof. Consider the constituent codewords 1 and 2 generated by the interleaver pattern, both containing two fundamental paths with input sequences of weight 2 as shown in Fig. 1, where there are two error patterns with input sequences of weight 2 at points x_i, $x_i+m\cdot(2^\nu-1)$, and x_j, $x_j+m\cdot(2^\nu-1)$ respectively. It is easy to check that the weight of codeword generated by the constituent code 1 is $2\cdot(m\cdot 2^{\nu-1}+2)$. Let us consider the error sequence with an input sequence of weight 2 x_i, $x_i+m\cdot(2^\nu-1)$. Since the distance between the two points is $m\cdot(2^\nu-1)$ and $L|(m\cdot(2^\nu-1))$, the two points are on the same ith LPP. Thus, each point is mapped to Px_i+L_i and $P(x_i+m\cdot(2^\nu-1))+L_i$, respectively. Since the input sequences for the constituent codes 1 and 2 are mapped by an interleaver, there is a point in the input for the constituent code 1 that is mapped to the point $Px_i+L_i+n\cdot(2^\nu-1)$ in the input for the constituent code 2. Let us call it x_j. Then $Px_j+L_j=Px_i+L_i+n\cdot(2^\nu-1)$. Since the distance between the points x_j, $x_j+m\cdot(2^\nu-1)$ is $m\cdot(2^\nu-1)$ and $L|(m\cdot(2^\nu-1))$, the two points are in the same jth LPP. Finally,

$$\begin{aligned} P(x_j+m\cdot(2^\nu-1))+L_j &= Px_j+L_j+Pm\cdot(2^\nu-1)\\ &= Px_i+L_i+(Pm+n)\cdot(2^\nu-1),\end{aligned}$$

which is equal to $P(x_i+m\cdot(2^\nu-1))+L_i+n\cdot(2^\nu-1)$. Thus, an input sequence of weight 4 exists for PLPP with L LPPs and the weight of the corresponding codeword is $2\cdot(m\cdot 2^{\nu-1}+2)+2\cdot(n\cdot 2^{\nu-1}+2)+4=(m+n)\cdot 2^\nu+12$. □

It should be mentioned that the upper bound in Lemma 2.3 assumes that the first coefficients of PLPP are all equal. This constraint was also imposed for the computation of L for LTE-QPPs in Table 1.

In Table 2, upper bounds on the minimum distance for turbo codes with PPs when $\nu=3$ are shown. The result in Lemma 2.3 is similar to Tables II and III in [1], however, Lemma 2.3 can also be applied to higher order PPs.

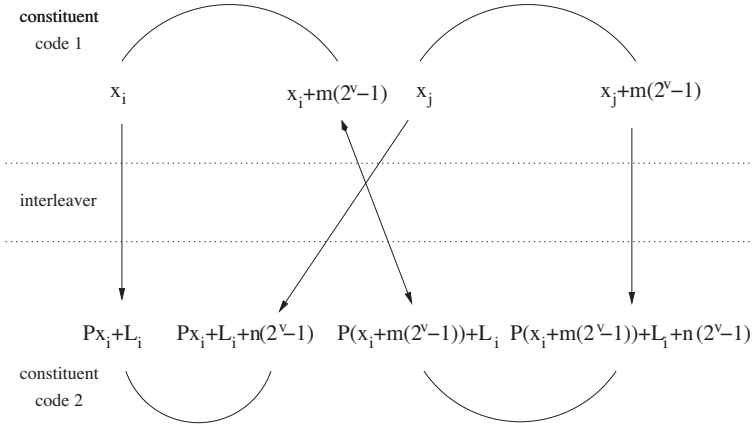

Fig. 1. Critical interleaver pattern of size 4 (Fig. 3 in [1]).