Reducing the complexity of distance measurement methods for circular turbo codes that use structured interleavers

Youssouf Ould-Cheikh-Mouhamedou*

College of Engineering, Prince Sultan Advanced Technologies Research Institute (PSATRI), King Saud University, P.O.Box: 800, Riyadh 11421, Saudi Arabia

SUMMARY

The knowledge of turbo code’s minimum Hamming distance \( d_{\text{min}} \) and its corresponding codeword multiplicity \( A_{\text{min}} \) is of a great importance because the error correction capability of a code is strongly tied to the values of \( d_{\text{min}} \) and \( A_{\text{min}} \). Unfortunately, the computational complexity associated with the search for \( d_{\text{min}} \) and \( A_{\text{min}} \) can be very high, especially for a turbo code that has high \( d_{\text{min}} \) value. This paper introduces some useful properties of turbo codes that use structured interleavers together with circular encoding. These properties allow for a significant reduction of search space and thus reduce significantly the computational complexity associated with the determination of \( d_{\text{min}} \) and \( A_{\text{min}} \) values. © 2014 The Authors. International Journal of Communication Systems published by John Wiley & Sons, Ltd.

Received 11 September 2013; Revised 6 November 2013; Accepted 4 December 2013

KEY WORDS: turbo codes; circular encoding; structured interleavers; distance measurement methods; minimum Hamming distance

1. INTRODUCTION

Turbo codes [1] are widely used in many mobile and satellite communication systems such as the Enhanced Voice-Data Optimized (EV-DO) [2], the Long Term Evolution (LTE) [3] of the 3rd Generation Partnership Project (3GPP), and the digital video broadcasting with return channel via satellite (DVB-RCS) [4]. The error rate performance of turbo codes is divided into two important regions. The first region, known as the waterfall region, is associated with low-to-moderate signal-to-noise ratio (SNR) values. The second region, known as the ‘error floor’ region, is associated with moderate-to-high SNR values. The authors of [5, 6] presented a method for performance evaluation of turbo-like codes based on estimating the probability density function of the bit log-likelihood-ratio using higher-order statistics.

The error floor [7] is mainly determined by the turbo code’s minimum Hamming distance \( d_{\text{min}} \) and its corresponding codeword multiplicity \( A_{\text{min}} \). The higher the \( d_{\text{min}} \) value and/or the lower the \( A_{\text{min}} \) value is, the lower the error floor is. The value of \( d_{\text{min}} \) does not depend only on interleaver size, but also on the constraint lengths of the component encoders as well as the proper selection of feedback and feedforward polynomials of the component encoders. That is, the higher the constraint length value is, the higher the \( d_{\text{min}} \) value is, assuming the use of a proper interleaver design. However, for fixed component encoders and fixed trellis termination scheme (e.g., circular

---

*Correspondence to: Youssouf Ould-Cheikh-Mouhamedou, College of Engineering, Prince Sultan Advanced Technologies Research Institute (PSATRI), King Saud University, P.O.Box: 800, Riyadh 11421, Saudi Arabia.
†E-mail: ycheikh@ksu.edu.sa

This is an open access article under the terms of the Creative Commons Attribution-NonCommercial License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited and is not used for commercial purposes.

© 2014 The Authors. International Journal of Communication Systems published by John Wiley & Sons, Ltd.
(also called tail-biting) [8], dual-termination [9], or state-mapping [10]), the value of $d_{\text{min}}$ depends entirely on the deployed interleaver. The better the interleaver design is, the higher the $d_{\text{min}}$ value is.

In [11], it has been shown that a parallel concatenated turbo code that uses tail-biting component encoders and quasi-cyclic permutation (i.e., structured interleaver) can be viewed as a quasi-cyclic turbo code, whose equation can be studied on an extension field of $GF(2)$. This creates a meaningful bridge between the rich algebraic theory of cyclic and the quasi-cyclic codes—that have led to design of large minimum Hamming distance block codes—and the problem of interleaver design and convolutional code selection for turbo codes. The work in [11] inspired the authors of [12] to study quasi-cyclic interleavers. The authors of [12] proposed a bidimensional quasi-cyclic interleaver and showed that, with probability that tends to 1 as the block length tends to infinity, the associated turbo code’s minimum Hamming distance is of order $\log(K)$, where $K$ is the size of the interleaver.

Other structured interleavers, capable of achieving high $d_{\text{min}}$ values, have also been introduced. Examples of such interleavers are the dithered relative prime (DRP) interleavers [13], the almost regular permutation (ARP) interleavers [14], and the quadratic permutation polynomial-based (QPP) interleavers [15]. The author of [16] presented some upper bounds on the best achievable $d_{\text{min}}$ values for turbo codes that use sufficiently large QPP interleavers. Through the use of exhaustive computer search, it has been verified that these upper bounds are tight for larger interleaver lengths. Furthermore, the author of [16] introduced a new set of improved QPP interleavers for 3GPP LTE turbo codes [17].

The determination of $d_{\text{min}}$ and $A_{\text{min}}$ values is a real challenge, especially when dealing with a well-designed interleaver that yields high $d_{\text{min}}$ value. This is because the determination of $d_{\text{min}}$ and $A_{\text{min}}$ values is typically based on searching all potential input sequences that can generate the $d_{\text{min}}$ value; and the number of potential input sequences (i.e., the search space) increases rapidly with a small increase in $d_{\text{min}}$ value.

This paper introduces a set of properties for structured interleavers and discusses their impacts on the search for $d_{\text{min}}$ and $A_{\text{min}}$ values when dealing with turbo codes that use circular encoding. Some of these properties have been used in [18, 19] to reduce the computational complexity associated with the determination of $d_{\text{min}}$ and $A_{\text{min}}$ values. However, the use of these properties in [18, 19] was based on intuitive reasoning rather than theoretical analysis. The main contribution of this paper is the introduction of a number of theorems and their proofs which form the theoretical foundation for the properties used in [18, 19]. To the best of my knowledge, this paper is the first one that provides theoretical analyses on how to explore the properties of structured interleavers and circular encoding to reduce the complexity associated with the determination of $d_{\text{min}}$ and $A_{\text{min}}$.

2. SYSTEM MODEL

Let $u = (u_0, \cdots, u_{K-1})$ be an input sequence of size $K$ information bits that enters the turbo-code encoder. Let $c = (u, p_1, p_2)$ be the output sequence (i.e., codeword) of size $N$ data bits that corresponds to $u$, where $u$ is the systematic part (i.e., the input sequence itself), $p_1$ is the first parity part generated by the first recursive systematic convolutional (RSC) constituent encoder that operates on the $K$ uninterleaved information bits $u$, and $p_2$ is the second parity part generated by the second constituent RSC encoder that operates on the $K$ interleaved information bits $v$. The bits in $v$ are constructed from the input sequence $u$ using the interleaver $\pi$. The value at position $i$ in $v$ is identical to the value at position $\pi(i)$ in $u$ (i.e., $v(i) = u(\pi(i))$, often written as $v_i = u_{\pi(i)}$). Consequently, $v$ can be expressed as $v = (u_{\pi(0)}, \cdots, u_{\pi(K-1)})$.

2.1. Encoding model

Each constituent RSC encoder uses circular encoding to avoid the use of termination bits that induces a code rate penalty. In circular encoding, the encoder starts the encoding at a sequence dependent state (called the circular state) $S_e$ and ends the encoding in the same state. The existence of circular state $S_e$ is guaranteed, if $K$ is not a multiple of $(2^\delta - 1)$, where $\delta$ is the number of shift registers of the constituent RSC encoder. The determination of $S_e$ requires a pre-encoding operation.
in which the RSC encoder starts in the all-zero state, encodes the sequence, and store the last state as \( S_0^K \). The circular state \( S_c \) is then obtained using the expression \( S_c = (I \oplus G^K)^{-1} \cdot S_0^K \) [8], where \( \oplus \) denotes the bit-by-bit exclusive OR operation, \( I \) the identity matrix, and \( G \) the generator matrix. The circular states can be pre-computed and stored for the desired interleaver length.

2.2. Interleaving model

The deployed interleaver is a structured interleaver from the DRP, ARP, or QPP family. Each one of these interleavers has the following interesting structural property [13]:

\[
\pi ([i + M]_K) = [\pi(i) + Mp]_K, \quad \forall i = 0, \ldots, K - 1
\]

where \([x]_K\) is \( x \)-modulo-\( K \). \( M \) is the size of repeating index increments of \( \pi \). \( K \) must be a multiple of \( M \) and the integer values \( p \) and \( K \) must be relatively prime to ensure that the interleaver references all \( K \) indices. Let vector \( q \) denotes the first \( M \) indices of \( \pi \) (i.e., \( q(0) = \pi(0), \ldots, q(M - 1) = \pi(M - 1) \)). It follows from (1) that the entire structured interleaver \( \pi \) is completely determined by the vector \( q \) and the parameters \( M \), \( p \), and \( K \).

Let \( \pi^{-1} \) denote the deinterleaver (i.e., the inverse interleaver) which is given by \( \pi^{-1}(\pi(i)) = i, \forall i = 0, \ldots, K - 1 \), where \( K \) is the size of the interleaver \( \pi \). Remember that, by definition, the value at position \( \pi(i) \) in \( u \) is interleaved using the interleaver \( \pi \) to the position \( i \) in \( v \). Substituting \( j \) for \( \pi(i) \), the expression \( i = \pi^{-1}(\pi(i)) \) becomes \( i = \pi^{-1}(j) \). This means that the position \( j \) in \( u \) is interleaved to the position \( \pi^{-1}(j) \) in \( v \).

3. SOME PROPERTIES OF STRUCTURED INTERLEAVERS

The following theorems and proofs introduce some useful properties of turbo codes that use circular encoding together with structured interleavers. Theorem 1 introduces a property of structured interleavers. This property will be used to facilitate the proof of Theorem 2.

A ‘shift’ of a sequence \( u \) by \( z \) positions refers to a circular shift of \( u \) to the left by \( z \) positions and is denoted \( u \downarrow z \). That is, if \( u = (u_0, u_1, \ldots, u_{z-1}, u_z, u_{z+1}, \ldots, u_{K-1}) \), then \( u \downarrow z = (u_z, u_{z+1}, \ldots, u_{K-1}, u_0, u_1, \ldots, u_{z-1}) \).

**Theorem 1**

Adding the value \( M \) to all indices of the structured interleaver \( \pi \), given by (1), yields an interleaver \( \pi' = \pi \uparrow \mu \), where \( \mu = [\lambda M]_K \) and \( \lambda \) is the congruent solution of the congruence equation \( p\lambda \equiv 1 \pmod{K} \).

**Proof**

By using (1), each element of the interleaver \( \pi \) can be expressed as \( \pi([i]_K) = \pi([i - M]_K + Mp]_K \). Applying this equation recursively \( \lambda \) times yields the expression \( \pi([i]_K) = [\pi([i - \lambda M]_K + \lambda Mp]_K \). Now, if there exists an integer \( \lambda \) such that \([\lambda M]_K = M \) is satisfied, then \([\pi([i - \lambda M]_K + \lambda Mp]_K \) is simply \([\pi([i - \lambda M]_K + M]_K \) (i.e., \( \pi([i]_K) = [\pi([i - \lambda M]_K + M]_K \). Substituting \( \ell \) for \( i - \lambda M \) yields \( [\pi([\ell + \lambda M]_K) = [\pi([\ell]_K) + M]_K \). Because by definition \( \pi'(\ell) = [\pi([\ell]_K) + M]_K \), then \( \pi'(\ell) = [\pi([\ell + \lambda M]_K) \). This means that \( \pi' \) is simply a circular shift of \( \pi \) to the left by \( [\lambda M]_K \) indices (i.e., \( \pi' = \pi \uparrow \mu \), where \( \mu = [\lambda M]_K \).

The proof of the existence of \( \lambda \) is based on the linear congruence theorem. Let \( a, b, \) and \( m \) be integers, and let \( g \) be the greatest common divisor of \( a \) and \( m \) (denoted by \( gcd(a, m) \)). According to the linear congruence theorem, the congruence equation \( ax \equiv b \pmod{m} \) has exactly \( g \) congruent solutions if \( g \) divides \( b \).

The term \([\lambda M]_K = M = [M]_K \) is simply saying that \( \lambda M p \) and \( M \) are congruent modulo \( K \) (i.e., \( \lambda M p \equiv M \pmod{K} \)). The congruence equation \( \lambda M p \equiv M \pmod{K} \) simplifies to \( p\lambda \equiv 1 \pmod{K} \) and because \( p \) and \( K \) are, by the definition of structured interleavers, relatively
prime (i.e., \( \gcd(p, K) = 1 \)), then there exits a congruent solution for \( \lambda \). In fact, there is a single congruent solution because \( g = 1 \).

The next theorem is introduced to pave the way for the main theorems, namely, Theorem 3 and Theorem 4.

Theorem 2

Let \( u = (u_0, u_1, \ldots, u_{M-1}, u_M, \ldots, u_{K-1}) \) be the input sequence that has by definition the corresponding interleaved sequence \( v = (u_{

50 \pi(0)}, u_{\pi(1)}, \ldots, u_{\pi(M-1)}, u_{\pi(M)}, \ldots, u_{\pi(K-1)}) \). For unpunctured turbo codes that use circular encoding together with a structured interleaver of length \( K \) and \( M \) repeating index increments, the input sequence \( \overrightarrow{u}^M \) has the corresponding interleaved sequence \( \overrightarrow{v}^\tau \) where \( \tau = \left\lfloor \pi^{-1}(M) - \pi^{-1}(0) \right\rfloor_K \) (i.e., \( \overrightarrow{v}^\tau \) is a circular shift of \( u \) by a multiple of \( M \) positions).

Proof

The proof consists of two parts. The first part shows the circular property and the second part determines the number of shift positions \( \tau \).

By definition, the input sequence \( \overrightarrow{u}^M \) has the corresponding interleaved sequence \( \overrightarrow{v}' = (\overrightarrow{v}^\tau_M(0), \overrightarrow{v}^\tau_M(1), \ldots, \overrightarrow{v}^\tau_M(K-1)) \). Because the position \( i \) in \( \overrightarrow{u}^M \) has the bit value \( u(i + M)_K \), it follows, by substituting \( \pi(j) \) for \( i \), that \( \overrightarrow{v}^M(\pi(j)) = \overrightarrow{u}((\pi(j) + M)_K) \). Consequently, \( \overrightarrow{v}' \) can be expressed as \( \overrightarrow{v}' = (\overrightarrow{u}^\pi(0)_K, \overrightarrow{u}^\pi(1)_K, \ldots, \overrightarrow{u}^\pi(K-1)_K) \). By using Theorem 1, \( \overrightarrow{v}' \) can be expressed as \( \overrightarrow{v} = (u_{\pi(0)}, u_{\pi(1)}, \ldots, u_{\pi(K-1)}) \). Because \( \pi' \) is simply a circular shift of \( \pi \) by a multiple of \( M \) positions, then \( \overrightarrow{v}' \) is also a circular shift of \( v \) by a multiple of \( M \) positions.

The number of shift positions \( \tau \) is determined as follows. Before the circular shift, \( u_M \) was at position \( M \) in \( u \) and also at position \( \pi^{-1}(M) \) in \( v \). After the circular shift, \( u_M \) is at position 0 in \( \overrightarrow{u}^M \) and also at position \( \pi^{-1}(0) \) in \( \overrightarrow{v}' \). If \( \pi^{-1}(0) < \pi^{-1}(M) \), then there is a circular shift to the left by \( \tau = \pi^{-1}(M) - \pi^{-1}(0) = \left\lfloor \pi^{-1}(M) - \pi^{-1}(0) \right\rfloor_K \) positions. If \( \pi^{-1}(0) > \pi^{-1}(M) \), then there is a circular shift to the right by \( \tau = \left\lfloor \pi^{-1}(M) - \pi^{-1}(0) \right\rfloor_K = \pi^{-1}(M) - \pi^{-1}(0) \) positions. Therefore, a circular shift to the right by \( \tau \) positions is the same as a circular shift to the left by \( \pi \) positions. Consequently, if \( \pi^{-1}(0) > \pi^{-1}(M) \), then there is a circular shift to the left by \( K + (\pi^{-1}(M) - \pi^{-1}(0)) = \pi^{-1}(M) - \pi^{-1}(0) \) positions.

Note that when combining Theorems 1 and 2, the number of shift positions \( \tau \) can be determined, without the computation of the inverse interleaver \( \pi^{-1} \), by simply solving the congruence equation \( p\lambda \equiv 1 \pmod{K} \) and then setting \( \tau = \lfloor \lambda M \rfloor_K \).

Theorem 3

For unpunctured turbo codes that use circular encoding together with a structured interleaver of length \( K \) and \( M \) repeating index increments, the codewords \( c \) and \( c' \) generated by \( u \) and \( \overrightarrow{u}^M \), respectively, have the same Hamming weight \( w \).

Proof

Let \( c = (u, p1, p2) \) be the codeword generated by \( u \). When applying Theorem 2, then the second parity of the codeword \( c' \) generated by the interleaved version of \( \overrightarrow{u}^M \) is \( \overrightarrow{p}^{2\tau} \), where \( \tau \) is a multiple of \( M \). Because the first parity of \( c' \) is simply \( p1^M \), then \( c' = (\overrightarrow{u}^M, \overrightarrow{p1}^M, \overrightarrow{p}^{2\tau}) \). Let \( w_H(c) = w \) denote the Hamming weight of \( c \). Because each value of \( c' \) is simply a shifted version of its corresponding value in \( c \), then \( w_H(c') = w \).

Theorem 4

This theorem deals with punctured turbo codes, where the repeating puncturing masks for the systematic part \( u \), the first parity part \( p1 \), and the second parity part \( p2 \) are of lengths \( M_s \), \( M_{p1} \), and \( M_{p2} \), respectively. For this punctured turbo codes that use a circular encoding together with a structured interleaver of length \( K \) and \( M \) repeating index increments, the codewords \( c \) and \( c' \)
generated by \( u \) and \( \overline{u}^T \), respectively, have the same Hamming weight \( w \), where \( T \) is the least common multiple (LCM) of \( M \), \( M_s \), \( M_{p1} \), and \( M_{p2} \).

**Proof**

Let \( w = w_0 + w_1 + w_2 \) be the Hamming weight of the punctured codeword \( c = (u, p1, p2) \) generated by the input sequence \( u \), where \( w_0 = w_H(u) \), \( w_1 = w_H(p1) \), and \( w_2 = w_H(p2) \). Let \( T \) be the minimum shift value such that \( \overline{u}^T \) generates the codeword \( c' = \left( \overline{u}^T, \overline{p1}^T, \overline{p2}^T \right) \) with the same Hamming weight \( w \). The value of \( \tau \) is a multiple of the value of \( T \), as shown in Theorem 2. If \( T \) is selected such that \( w_0 = w_H(\overline{u}^T) \), \( w_1 = w_H(\overline{p1}^T) \), and \( w_2 = w_H(\overline{p2}^T) \), then the Hamming weight of \( c' \) is guaranteed to be the same as the Hamming weight of \( c \).

Because of the presence of puncturing masks, the condition \( \left[ w_0 = w_H(\overline{u}^T) \right. \) and \( w_1 = w_H(\overline{p1}^T) \) \) and \( w_2 = w_H(\overline{p2}^T) \) \] is guaranteed to be satisfied only if \( [T \) is equal to \( M \) or a multiple of \( M_s \), and at the same time, \( T \) is equal to \( M_{p1} \) or a multiple of \( M_{p1} \), and at the same time, \( \tau \) is equal to \( M_{p2} \) or a multiple of \( M_{p2} \) (i.e., \( T \) is equal to \( M_{p2} \) or a multiple of \( M_{p2} \) because \( \tau \) is a multiple of \( T \)). Further, because of the nature of the structured interleaver, \( T \) must be also equal \( M \) or a multiple of \( M \). Consequently, the minimum value of \( T \) is the least common multiple of \( M_s, M_{p1}, M_{p2}, \) and \( M \). \( \square \)

4. IMPACT ON DISTANCE MEASUREMENT METHODS

This section discusses how the theorems, discussed in the previous section, can be used to (i) reduce the complexity of distance measurement methods and (ii) provide an upper bound on codeword multiplicity.

4.1. Complexity reduction

For unpunctured turbo codes, it was shown in Theorem 3 that if \( u_1 \) generates codeword \( c_1 \) with \( w_H(c1) = w \), then \( \overline{u}_1^M \) will also generate codeword \( c'_1 \) with \( w_H(c'_1) = w \). Applying this result recursively shows that all shift versions \( \overline{u}_1^M, \overline{u}_1^{2M}, \ldots, \overline{u}_1^{(L-1)M} \) will also generate codewords with Hamming weight \( w \), where \( L = \frac{K}{M} \). This means that if a distance measurement method tests the input sequence \( u_1 \) and finds out that it yields a codeword \( c_1 \) with Hamming weight \( w \), then it does not need to test all circular shifts of \( u_1 \) by a multiple of \( M \) positions because they are guaranteed to yield the same Hamming weight \( w \). Consequently the search space for a specific distance, let say \( d_{min} \), is reduced by a factor of \( L \).

For punctured turbo codes, Theorems 4 shows, when applied recursively, that all input sequences \( u_1, \overline{u}_1^T, \overline{u}_1^{2T}, \ldots, \overline{u}_1^{(L-1)T} \) generates codewords with the same Hamming weight, where \( L' = \frac{K}{T} \). Consequently the search space for a specific distance, let say \( d_{min} \), is reduced by a factor of \( L' \). A Summary of complexity reduction for both unpunctured and punctured turbo codes is provided in Table I.

The magnitude of complexity reduction is demonstrated using the powerful structured QPP interleavers, which are generated using quadratic polynomial of the form \( f(x) = f_1 x + f_2 x^2 \ (mod \ K) \), where \( f_1 \) and \( f_2 \) are nonnegative integers. For a QPP interleaver of length \( K \), the size of the repeating index increments \( M \) is given by \( M = \frac{K}{gcd(2 \cdot f_2, K)} \) [20], where \( gcd(a, b) \) denotes the greatest common divisor of \( a \) and \( b \). For the interleaver of length \( K = 1600 \), the 3GPP

| Table I. A Summary of complexity reduction for unpunctured and punctured turbo codes. |
|-----------------------------------|-----------------|-----------------|
| Unpunctured turbo code            | Punctured turbo code |
| Interleaver length                | \( K \)          | \( K \)          |
| Repeating pattern length          | \( M \)          | \( T = LCM (M, M_s, M_{p1}, M_{p2}) \) |
| Complexity reduction              | \( L = \frac{K}{M} \) | \( L' = \frac{K}{T} \) |
LTE turbo codes use the polynomial $f(x) = 17x + 80x^2 \pmod{1600}$ [16], which means that $M = 1600/gcd(2 \cdot 80, 1600) = 10$. Consequently, for this interleaver, the complexity associated with the determination of $d_{\text{min}}$ can be reduced by a factor of $K/M = 1600/10 = 160$.

4.2. Upper bound on codeword multiplicity

It has been shown for unpunctured turbo codes that all input sequences $u_1, \hat{u}_1^M, \hat{u}_1^{2T}, \cdots, \hat{u}_1^{(L-1)M}$ generate codewords with the same Hamming weight $w$, where $L = \frac{K}{d}$. It has also been shown for punctured turbo codes that all input sequences $u_1, \hat{u}_1^T, \hat{u}_1^{2T}, \cdots, \hat{u}_1^{(L-1)T}$ generate codewords with the same Hamming weight $w'$, where $L' = \frac{K}{d}$. For illustration purpose, let assume that the only input sequences that generate codewords with Hamming weights $w$ and $w'$ are the ones listed previously. Consequently, there are ‘at most’ $L$ and $L'$ codewords with Hamming weights $w$ and $w'$.

The reason for saying ‘at most’ because one or several shift versions of $u_1$ might produce identical codewords. The following examples illustrate this. The codeword caused by all-one input sequence $u_1 = (1, 1, \cdots, 1, 1)$ has a multiplicity of 1 because all shift versions of $u_1$ by a multiple of $M$ produce a codeword identical to the codeword produced by $u_1$. Another example, for $M = 4$, the codeword caused by $u_2 = (e_1, e_1, \cdots, e_1)$ and $e_1 = (1, 0, 0, 0)$ have a multiplicity of 1 because all shift versions of $u_2$ by a multiple of 4 produce a codeword identical to the codeword produced by $u_2$.

Last example, for $M = 4$ and $K = 32$, let $d_{\text{min}}$ be caused by two identical events $e_2 = (1, 0, 0, 1)$ and $e_3 = (1, 0, 0, 1)$, where $e_2$ is at position 0 and $e_3$ is at position 16 and are separated by 12 zeros (i.e., $u_{\text{min}} = (e_2, 0000, 0000, 0000, e_3, 0000, 0000, 0000)$). Because a shift of $u_{\text{min}}$ by 16 positions results in the same $u_{\text{min}}$, the codeword multiplicity is $\frac{32}{4} - 1 = 7$.

From the discussion in the previous text, it follows that if any distance, let say $d_{\text{min}}$, is caused by $\ell$ unique input sequences (i.e., no circular shift of one is identical to the other), then the codeword multiplicity is upper bounded by $\ell \cdot L$ in case no puncturing is applied and $\ell \cdot L'$ in case puncturing is applied. In fact, extensive test results for different structured interleaves have shown that the multiplicity of codewords of Hamming weight $d_{\text{min}}$ are exactly $\ell \cdot L$ and $\ell \cdot L'$ for unpunctured and punctured turbo code, respectively (see references [18, 21]). In general, it is fair to say that the upper bound on multiplicity is very tight for very low-weight codewords (i.e., $d_{\text{min}}$ and next few distances) and the upper bound gets looser for higher distances. In any case, the multiplicity of codewords of Hamming weight $d_{\text{min}}$ can be determined accurately by simply counting only unique shift versions of each one of the $\ell$ input sequences that cause $d_{\text{min}}$.

4.3. Experimental results

To assess the actual reduction in computational complexity, the following setting has been used:

(1) The 4-state turbo codes that use circular encoding is considered. The 4-state turbo codes use recursive systematic convolutional constituent encoders each with feedback and feedforward polynomials of 7 and 5 in octal, respectively. The deployed interleaver is the QPP interleaver, discussed in the previous text (generated by the polynomial $f(x) = 17x + 80x^2 \pmod{1600}$), of length $K = 1600$ and $M = 10$ (i.e., the size of repeating index increments is 10). For unpunctured turbo code, the code rate is 1/3. For punctured turbo code, the code rate is 1/2 and is obtained using the puncturing masks [1, 1, 0, 0] and [1, 1, 0, 0] for systematic part, first parity part, and second parity part, respectively. The values 0 and 1 indicate punctured (removed) and kept bits, respectively. The puncturing masks have the lengths $M_s = 1$, $M_{p1} = 2$, and $M_{p2} = 4$, which implies $T = \text{LCM} (M, M_s, M_{p1}, M_{p2}) = \text{LCM} (10, 1, 2, 4) = 20$.

(2) The $d_{\text{min}}$ and $A_{\text{min}}$ are computed using the double-impulse distance estimation method, which assumes that the all-zero codeword has been transmitted, but when decoding the decoder is forced to produce a non-zero codeword that have the bit value 1 at positions $i$ and $j$ in the systematic part. All possible combinations for $i$ and $j$ are tested (i.e., $i = (0, 1, \cdots, K - 1)$ and $j = (i + 1, \cdots, K - 1)$), and the value of $d_{\text{min}}$ is the one associated with the codeword that has the lowest Hamming distance. More details about the double-impulse method are available in [22].
The values $d_{\text{min}}$ and $A_{\text{min}}$ are for the classical case where all possible positions (i.e., $i = (0, 1, \cdots, K - 1)$ and $j = (i + 1, \cdots, K - 1)$) are tested. The values $d_{\text{min}}^*$ and $A_{\text{min}}^*$ are the values when applying Theorem 3 (for unpunctured turbo code) and Theorem 4 (for punctured turbo code), respectively. That is, for unpunctured turbo code, the test space is reduced to the combinations $i = (0, 1, \cdots, M - 1)$ and $j = (i + 1, \cdots, K - 1)$; and for punctured turbo code, the test space is reduced to the combinations $i = (0, 1, \cdots, T - 1)$ and $j = (i + 1, \cdots, K - 1)$, where $K = 1600$, $M = 10$, and $T = 20$. The ratio $\omega^*_i / \omega^*$ is the actual reduction in complexity obtained from the computational experiment, where $\omega$ is the time needed to compute $d_{\text{min}}$ and $\omega^*$ the time needed to compute $d_{\text{min}}^*$.

The reported CPU times in Table II were obtained with a 3.20 GHz Intel Xeon Processor W3670. Table II shows that the experimental results are inline with the theoretical results. That is,

1. The same minimum Hamming distance has been found for both cases,
2. The reduction in computational complexity is essentially in the same range (i.e., theoretical and actual complexity differ only by a factor of about 2), and
3. The codeword multiplicity are within acceptable range to each other. This is because the packet error rate (PER) depends linearly on the codeword multiplicity (e.g., if the true PER is $10^{-6}$, then a pessimistic mismatch in codeword by a factor of 2 will yield a PER of $2 \cdot 10^{-6}$, which is essentially the same as the true PER).

5. CONCLUSIONS

Some useful distance properties of structured interleavers have been presented. These properties are useful when searching for the minimum Hamming distance ($d_{\text{min}}$) of a turbo code that uses structured interleaver together with circular encoding because these properties allow for a significant reduction of search space and thereby reduce significantly the computational complexity associated with the determination of $d_{\text{min}}$ value. Further, an upper bound on codeword multiplicity has been presented. The knowledge of $d_{\text{min}}$ and its corresponding codeword multiplicity ($A_{\text{min}}$) is of a great importance because the error correction capability of a turbo code is strongly tied to the values of $d_{\text{min}}$ and $A_{\text{min}}$.

ACKNOWLEDGEMENTS

Special thank to Dr. Eirik Rosnes (was with the Selmer Center, Department of Informatics, University of Bergen, Norway, and is now with Ceragon Networks, Kokstad, Norway) for his helpful comments and suggestions. Also, special thank to the National Plan for Sciences, Technology and Innovation (NPST) for funding this work under the research project 11-ELE1880-02.

REFERENCES

1. Berrou C, Glavieux A, Thitimajshima P. Near Shannon limit error-correcting coding and decoding: Turbo-codes. Proceedings IEEE International Conference on Communications (ICC’93), Geneva, Switzerland, May 1993; 1064–1070.
2. Bayat O. Intersymbol interference cancellation in cdma 1xEVDO network. International Journal of Communication Systems; 2012. DOI: 10.1002/dac.2418.

© 2014 The Authors. International Journal of Communication Systems published by John Wiley & Sons, Ltd.

Int. J. Commun. Syst. 2015; 28:1572–1579
DOI: 10.1002/dac
3. Bouras C, Kanakis N, Kokkinos V, Papazois A. Application layer forward error correction for multicast streaming over LTE networks. *International Journal of Communication Systems*, 2013; 26:1459–1474.

4. Celandroni N, Ferro E, Gotta A. RA and DA satellite access schemes: a survey and some research results and challenges. *International Journal of Communication Systems*, 2013. DOI: 10.1002/dac.2498.

5. Abedi A, Thompson ME, Khandani AK. Application of cumulant method in performance evaluation of turbo-like codes. *Proceedings IEEE International Conference on Communications (ICC’07)*, Glasgow, Scotland, June 2007; 986–989.

6. Abedi A, Thompson ME, Khandani AK. Application of cumulant method in performance evaluation of turbo-like codes. *IEEE Transactions on Communications*, 2007; 55:2037–2041.

7. Ould-Cheikh-Mouhamedou Y. A simple and efficient method for lowering the error floors of turbo codes that use structured interleavers. *IEEE Communications Letters*, 2012; 16:392–395.

8. Berrou C, Douillard C, Jézéquel M. Multiple parallel concatenation of circular recursive convolutional (CRSC) codes. *Annals of Telecommunications*, 1999; 54:166–172.

9. Guinand P, Lodge J. Trellis termination for turbo encoders. *Proceedings 17th Biennial Symposium on Communications*, Kingston, Canada, May 30–June 1, 1994; 389–392.

10. Sun J, Takeshita OY. Extended tail-biting schemes for turbo codes. *IEEE Communications Letters*, 2005; 9:252–254.

11. Tanner RM. Toward and algebraic theory for turbo codes. *Proceedings 2nd International Symposium Turbo Codes*, Brest, France, September 2000; 17–25.

12. Boutros JJ, Zemor G. On quasi-cyclic interleavers for parallel turbo codes. *IEEE Transactions on Information Theory*, 2006; 52:1732–1739.

13. Crozier S, Guinand P. High-performance low-memory interleaver banks for turbo-codes. *Proceedings 54th IEEE Vehicular Technology Conference, (VTC’01)*, Atlantic City New Jersey, USA,(www.crc.ca/tc), October 2001; 2394–2398.

14. Berrou C, Saouter Y, Douillard C, Kerouédan S, Jézéquel M. Designing good permutations for turbo codes: towards a single model. *IEEE International Conference on Communications (ICC’04)*, Paris, France, June 2004; 341–345.

15. Sun J, Takeshita OY. Interleavers for turbo codes using permutation polynomials over integer rings. *IEEE Transactions on Information Theory*, 2005; 51:101–119.

16. Rosnes E. On the minimum distance of turbo codes with quadratic permutation polynomial interleavers. *IEEE Transactions on Information Theory*, 2012; 58:4781–4795.

17. “3rd generation partnership project; technical specification group radio access network; evolved universal terrestrial radio access (eutra); multiplexing and channel coding (release 8), 3gpp ts 36.212 v8.5.0,” December 2008.

18. Ould-Cheikh-Mouhamedou Y, Crozier S, Kabal P. Efficient distance measurement method for turbo codes that use structured interleavers. *IEEE Communications Letters*, 2006; 10:477–479.

19. Crozier S, Guinand P, Hunt A. On designing turbo-codes with data puncturing. *9th Canadian Workshop on Inform. Theory (CWIT’05)*, Montreal, Quebec, Canada, June 2005; 32–35.

20. Takeshita OY. Permutation polynomial interleavers: an algebraic-geometric perspective. *IEEE Transactions on Information Theory*, 2007; 53:2116–2132.

21. Ould-Cheikh-Mouhamedou Y, Crozier S, Kabal P. Distance measurement method for double binary turbo codes and a new interleaver design for DVB-RCS. *Proceedings IEEE Global Telecommunications Conference (GLOBECOM ’04)*, Dallas, Texas, USA, November 29–December 3, 2004.

22. Ould-Cheikh-Mouhamedou Y, Crozier S, Kabal P. Comparison of distance measurement methods for turbo codes. *9th Canadian Workshop on Inform. Theory (CWIT’05)*, Montreal, Quebec, Canada, June 2005; pp. 36–39.
