A Novel Method for Obtaining the Pattern of Low-Weight Codeword Components of Recursive Systematic Convolutional Codes

Bohulu Kwame Ackah and Chenggao Han
Graduate School of Informatics and Engineering,
The University of Electro-Communications,
1-5-1 Chofugaoka, Chofu-shi, Tokyo, 182–8585, Japan
Email: {bohulu, han.ic}@uec.ac.jp

Abstract

In this paper, we present a novel low-complexity method for obtaining the pattern of low-weight codeword components for a given recursive systematic convolutional code. We generate a low-weight codeword component pattern list for selected recursive systematic convolutional codes and validate our proposed method by obtaining a union bound, which we compare to simulation results and the union bound obtained via the transfer function method. From the results, we are able to determine which recursive systematic convolutional codes are best suited for use in turbo codes.

I. Introduction

The *turbo code* (TC) [1], introduced by Claude Berrou in 1993 is one of the *forward-error correcting* (FEC) codes that comes very close to satisfying the Shannon limit for AWGN channels. Due to its excellent performance, TCs have been used in many applications, and have been adopted as the channel code for the LTE standard, IEEE 802.16 WiMAX (worldwide interoperability for microwave access) and DVB-RCS2 (2nd generation digital video broadcasting - return channel via satellite) standards [6].

The simplest and most common construction of a TC is to concatenate two *recursive systematic convolutional* (RSC) codes (usually of the same kind) parallely via an interleaver. One of the many reasons why the TC excels as a FEC code is due to its ability to map low-weight parity-check sequences in the first RSC code to high-weight parity-check sequences in the second RSC code using the interleaver, which in turn generates TCs with a large minimum distance value.

The design of a good deterministic interleaver requires the complete knowledge of all the low-weight codeword component patterns in the RSC code and missing even one of these patterns can result in deterministic interleavers that generate TCs with sub-par error correction performance. The transfer function of an RSC code is an interleaver design tool that provides information about the different weights in the code, as well as their corresponding multiplicities (distance spectrum). However, it provides no information with regards to the pattern of the low-weight codeword components. As an added downside, the complexity of calculating the transfer function for a given RSC code increases with the number of states, and other methods such as Mason's Rule [3] have to be used. Research into other methods for finding the distance spectrum have been carried in recent years. In [7], an algorithm for evaluating the input-parity weight distribution of terminated RSC codes is presented, while in [8], the distance spectrum of tail-biting duo-binary RSC codes is calculated using the modified FAST algorithm. These methods also do not reveal the pattern of the low-weight codeword components and to the best of our knowledge, there exists no interleaver design tool that provides knowledge of both the distance spectrum and the low-weight codeword component patterns. Because of this, many of the interleaver design methods end up completely ignoring certain important low-weight codewords. In [5] for example, the interleaver design method does not take into account the existence of low-weight codewords with systematic components of weight 3, especially for the $5/7$ RSC code, where such codewords are dominant.

In this paper, we propose a novel method for revealing the pattern of the low-weight codeword components. The complexity of our proposed method is independent of the number of states of the RSC code and its ability to reveal the low-weight codeword patterns of an RSC code makes it an excellent interleaver design tool. We generate a low-weight codeword component pattern list for specific RSC codes and obtain union bounds using our proposed method. We then validate our method by comparing the proposed union bounds to simulation results and the union bounds obtained via the transfer function method.

The remainder of the research paper is organised as follows. Definitions used in the research paper are introduced in Section II. In Section III, we establish the theoretical foundations for our novel method by discussing the characteristics of the low-weight codewords. Then in Section IV, we present our novel method and use examples to clarify the workings of our proposed method. Validation of our proposed method for specific RSC codes as well as discussion related to turbo code interleaver design is done in Section V and the paper concludes in Section VI.

A. Notations

For two positive integers α and β, the least common multiple of α and β is dentoed as $\mathrm{lcm}(\alpha, \beta)$ while the remainder α divided by β is denoted as $\alpha \mod \beta$. $\alpha|\beta$ implies α is a divisor of β. For an integer pair $(\alpha,\ \beta)$, $(\alpha,\ \beta)\mathrm{mod}$ is shorthand for the operation $(\alpha \bmod \epsilon_0,\ \beta \bmod \epsilon_0)$. For two integer sets $\mathcal{M}$ and $\mathcal{N}$, the tensor product that yields the set consisting of all pairs of $\mathcal{M}$ and $\mathcal{N}$ is denoted as $\mathcal{M} \otimes \mathcal{N}$ and we assume the elements in each resultant pair are sorted in increasing order.

II. Preliminaries

A polynomial in x with degree M is an expression of the form

$$v(x) = \sum_{m=0}^{M} v_m x^m \tag{1}$$

where $v_m,\ 0 \le m \le M$, are called the *coefficients* and $v_M \neq 0$. If $v_M = 1,\ v(x)$ is called a *monic* polynomial. We call the total number of the non-zero coefficients the *Hamming weight* of $v(x)$, denoted as $w_H(v(x))$.

For a prime number p, if the addition and multiplication of two elements in the integer set $\{0, 1, p-1\}$ are performed on the terms $\bmod\ p$, we call the set a Galois field, denoted as $\mathrm{GF}(p)$. If the coefficients in (1) are elements of $\mathrm{GF}(p)$, $v(x)$ is called a *polynomial over* $\mathrm{GF}(p)$.

For two polynomials $v(x)$ and $w(x)$ with degrees M and N, respectively, the addition and multiplication over $\mathrm{GF}(p)$ are defined as

$$v(x) + w(x) = \sum_{m=0}^{\max\{M,N\}} [(v_m + w_n) \mod p] x^m \tag{2}$$

and

$$v(x)w(x) = \sum_{m=0}^{M+N} \sum_{i=0}^{m} [v_i w_{m-i} \mod p] x^m \tag{3}$$

respectively.

We say a monic polynomial is a *prime polynomial* if it cannot be obtained by the multiplication of some lower degree polynomials. For two polynomials $v(x)$ and $w(x)$ over $\mathrm{GF}(p)$, $w(x) \neq 0$, there exists polynomials $q(x)$ and $r(x)$ over $\mathrm{GF}(p)$ such that

$$v(x) = w(x)q(x) + r(x) \tag{4}$$

with $\deg(r(x)) < \deg(w(x))$. We represent $r(x)$ in the expression (4) as

$$r(x) \equiv v(x) \mod w(x) \tag{5}$$

and call it the *remainder polynomial*, while $q(x)$ is called the *quotient polynomial* of the division of $v(x)$ by $w(x)$.

Let $v(x)$ be a prime polynomial over $\mathrm{GF}(p)$ with $\deg(v(x)) := M > 1$ and $\mathcal{V}$ be the set of size p^M containing all polynomials over $\mathrm{GF}(p)$ with degree less than M. Then, the *extension field of* $\mathrm{GF}(p)$, denoted by $\mathrm{GF}\left(p^M\right)$, is the set $\mathcal{V}$ with addition and multiplication over $\mathrm{GF}(p)$, where the multiplication is carried out modulo-$v(x)$ over $\mathrm{GF}(p)$. Each non-zero element in $\mathrm{GF}\left(p^M\right)$ can be represented by a power of x uniquely as $x^m,\ 0 \leq m \leq p^M - 1$.

For each non-zero element of $\mathrm{GF}\left(p^M\right)$, there exist integers ϵ such that $x^\epsilon = 1$ and the least positive integer among them is called the *order* of x. We say that elements with order $\epsilon = p^M - 1$ are *primitive elements*. For $\mathrm{GF}\left(p^M\right)$ generated by a prime polynomial $v(x)$ with $\deg(v(x)) = M$, if x is a primitive element in $\mathrm{GF}\left(p^M\right)$, then $v(x)$ is called a *primitive polynomial*. Finally, the root of $v(x)$, is the non-zero element $\varphi,\ \varphi \in \mathrm{GF}\left(p^M\right)$ such that $v(\varphi) = 0$. If $v(x)$ is a primitive polynomial, the order of φ is $\epsilon = p^M - 1$, otherwise $\epsilon | p^M - 1$. Moreover, the elements $\varphi^i,\ 0 \leq i \leq \epsilon - 1$, are all distinct from each other.

III. The characteristics of the low-weights codewords of RSC code

The outputs of an RSC code are determined by the input bit sequence $b(x)$, the states of the shift registers and the feedforward and feedback connections of the shift registers that can be represented by a generator function.

As an instance, the generator function of a rate $1/2$ RSC code may be written as

$$\left[1 \ \frac{f(x)}{g(x)}\right]$$

where 1 yields the *systematic component* (SC) $b(x)$ while the *parity-check component* (PC) $h(x)$ is associated with the feedforward and feedback connections of the shift registers, specified by $f(x)$ and $g(x)$, respectively. The outputs $c(x)$ are the mixture of the SC and PC as

$$c(x) = b(x^2) + xh(x^2) \tag{6}$$

where

$$h(x) = f(x)g^{-1}(x)b(x) \tag{7}$$

From (6), it is trivial that

$$w_H(c(x)) = w_H(b(x)) + w_H(h(x)) \tag{8}$$

and hence, each low-weight codeword is combination of low-weight SC and PC.

Under the assumption of large frame sizes, the presence of $g^{-1}(x)$ in (7) may involve a particular bit sequence that repeats a large number of times, hence yielding a high-weight PC. Low-weight PCs occur if and only if

$$b(x) \bmod g(x) \equiv 0 \tag{9}$$

The SCs satisfying (9) are called *return-to-zero* (RTZ) inputs. Thus, every RTZ input can be factorized as

$$b(x) = a(x)g(x) \tag{10}$$

and, substituting (10) into (7), we can characterize the low-weight PC as

$$\begin{aligned} h(x) &= f(x) \cdot g^{-1}(x) \cdot a(x)g(x) \\ &= a(x)f(x) \end{aligned} \tag{11}$$

Therefore, in this paper, we attempt to find $a(x)$s satisfying (10) and (11) simultaneously for low-weight $b(x)$ and $h(x)$, respectively. However, since there is no essential mathematical difference between the two equations, in the next section, we present a method for determining the low-weight PC patterns for $2 \leq w_H(h(x)) \leq 3$.

IV. The patterns of the low-weight PCs

We assume $f(x)$ can be factorized into K prime polynomials as

$$f(x) = \prod_{k=0}^{K-1} f_k^{\gamma_k}(x) \tag{12}$$

where $\gamma_0, \gamma_1, \cdots, \gamma_{K-1}$ are positive integers and let φ_k be a root of $f_k(x)$ of order ϵ_k. After that, refering to (11), we consider the solution of

$$h(x) \mod f(x) \equiv 0 \tag{13}$$

We start from the simplest case $K = 1$, *i.e.*, $f(x) = f_0^{\gamma_0}(x)$. Then, (11) indicates that each root of $f(x)$ is also a root of $h(x)$ and we distinguish between the cases $\gamma_0 = 1$ and $\gamma_0 > 1$. For the former case, since all φ_0^i, $0 \le i < \epsilon_0$, are distinct from each other, the equation

$$h(\varphi_0^i) = 0, \quad 0 \le i < \epsilon_0 \tag{14}$$

is a necessary and sufficient condition of (13) while it is necessary but not sufficient for the latter case. For the case $\gamma_0 > 1$, we may derive some solutions by differential equations

$$\left.\frac{d^{(j)}h(x)}{dx^j}\right|_{x=\varphi_0^i} = 0, \quad 0 \le i < \epsilon_0,\ 1 \le j < \gamma_0 \tag{15}$$

However, since operations on the coefficients of the polynomial are performed on the terms $\mod p$, we need further confirmation to remove possible ghost solutions.

For the case where $K > 1$, we may repeat the above discussion for the roots φ_k, $0 < k < K$, and take the intersection of the results to determine the low-weight PCs.

A. The weight-2 PCs

Each weight-2 PC can be written as

$$h(x) = 1 + x^\alpha \tag{16}$$

without loss of generality. Thus from (14), we have

$$(\varphi_0^i)^\alpha = 1, \quad 0 \le i < \epsilon_0 \tag{17}$$

On the other hand, the order of $\varphi_0,\ \epsilon_0$ is the least integer satisfying $\varphi_0^{\epsilon_0} = 1$, thus, α should satisfy the condition

$$\alpha \equiv 0 \mod \epsilon_0 \quad \text{or} \quad \epsilon_0 | \alpha \tag{18}$$

B. The weight-3 PCs

Each weight-3 PC can be written as

$$h(x) = 1 + x^{\alpha} + x^{\beta},\ \alpha < \beta \tag{19}$$

without loss of generality. Thus, (α, β) should satisfy the condition

$$\varphi_0^{\alpha} + \varphi_0^{\beta} = 1 \tag{20}$$

The pairs satisfying (20) can be found by referring to the table of the extended field for $\mathrm{GF}\left(2^M\right)$. Let (m, n) be such a pair, and we let $\mathcal{M} = \{\epsilon_0 \ell + m\}$ and $\mathcal{N} = \{\epsilon_0 \ell + n\}$, $\ell \geq 0$. Then it is obvious that each pair $(\alpha,\ \beta) \in \mathcal{M} \otimes \mathcal{N}$ satisfies (20). For a fixed α, on the other hand, since $\alpha + i$, $0 \leq i < \epsilon_0$, are distinct from each other, any integer β that satisfies (20) must be such that $n \equiv \beta \mod \epsilon_0$.

C. Examples

Example 1. $f(x) = 1 + x + x^2$

Since $x^1 = x$, $x^2 \equiv 1 + x \mod f(x)$, and $x^3 \equiv 1 \bmod f(x)$, $f(x)$ is a primitive polynomial with root φ_0 of order $\epsilon_0 = 3$. Thus, α for the weight-2 PCs in the form $1 + x^{\alpha}$ should be a multiple of 3 and we may express such PCs as $h(x) = 1 + x^{3\ell}$, $\ell > 1$, while the corresponding $a(x)$ is given by

$$a(x) = \sum_{i=0}^{\ell-1} x^{3i}(1 + x)$$

The detailed patterns of $a(x)$ and $h(x)$ for $1 \leq \ell \leq 4$ are listed in Table I.

TABLE I: Weight-2 PCs for $f(x) = 1 + x + x^2$

$a(x)$	$h(x)$
$1 + x$	$1 + x^3$
$1 + x + x^3 + x^4$	$1 + x^6$
$1 + x + x^3 + x^4 + x^6 + x^7$	$1 + x^9$
$1 + x + x^3 + x^4 + x^6 + x^7 + x^9 + x^{10}$	$1 + x^{12}$

To determine the weight-3 PCs, we can see from Table II that there is a pair $(1, 2)$ satisfying $x^1 + x^2 \equiv 1 \mod f(x)$. Thus, if we let $\mathcal{M} = \{3\ell + 1\}_{\ell \geq 0}$ and $\mathcal{N} = \{3\ell + 2\}_{\ell \geq 0}$, $x^{\alpha} + x^{\beta} \equiv 1$

TABLE II: Non-zero Elements of $\mathrm{GF}\left(2^2\right)$ generated by $f(x)=1+x+x^2$

power representation	polynomial representation
$x^0 \;= x^3 = 1$	1
x	x
x^2	$1+x$

TABLE III: Weight-3 PCs for $f(x)=1+x+x^2$

$a(x)$	$h(x)$
1	$1+x+x^2$
$1+x+x^2$	$1+x^2+x^4$
$1+x+x^3$	$1+x^4+x^5$
$1+x^2+x^3$	$1+x+x^5$

$\bmod f(x)$ for each pair $(\alpha,\beta)\in\mathcal{M}\otimes\mathcal{N}$. The PCs $h(x)=1+x^{3\ell_0+1}+x^{3\ell_1+2}$ for $0\le\ell_0,\ell_1\le 1$ and the corresponding $a(x)$ are listed in Table III.

Example 2. $f(x)=1+x+x^2+x^3+x^4$

Since the order of φ_0 is $\epsilon_0=5<15$, $f(x)$ is a prime but not primitive polynomial. The weight-2 PCs $h(x)=1+x^{5\ell}$, $1\le\ell\le 4$, and the corresponding $a(x)$ are shown in Table IV. For weight-3 PCs, on the other hand, we can see from Table V that there is no pair $(m,\ n)$ satisfying $x^m+x^n\equiv 1$, and hence, the given $f(x)$ does not yield weight-3 PCs.

TABLE IV: $f(x)=1+x+x^2+x^3+x^4$

$\alpha(x)=\sum_{i=0}^{\ell-1}x^{5i}(1+x)$	$h(x)=1+x^{5\ell}$
$1+x$	$1+x^5$
$1+x+x^5+x^6$	$1+x^{10}$
$1+x+x^5+x^6+x^{10}+x^{11}$	$1+x^{15}$
$1+x+x^5+x^6+x^{10}+x^{11}+x^{15}+x^{16}$	$1+x^{20}$

Example 3. $f(x)=1+x^2$

TABLE V: Non-zero Elements of $\mathrm{GF}(2^4)$ generated by $f(x) = 1 + x + x^2 + x^3 + x^4$

power representation	polynomial representation
$x^0 = x^5 = x^{10} = x^{15}$	1
$x = x^6 = x^{11}$	x
$x^2 = x^7 = x^{12}$	x^2
$x^3 = x^8 = x^{13}$	x^3
$x^4 = x^9 = x^{14}$	$1 + x + x^2 + x^3$

If we rewrite the polynomial as $f(x) = (1 + x)^2$, the order of the root φ_0 is $\epsilon_0 = 1$. Since $\mathrm{GF}(2)$ has single non-zero element, it does not provide a pair $(m,\ n)$ satisfying $x^m + x^n \equiv 1$ and there are no weight-3 PCs associated with $f(x)$.

Regarding the weight-2 PCs of the form $h(x) = 1 + x^\alpha$, although $(\varphi_0)^\alpha = 1$ implies α can be any positive integer, we can obtain $\alpha(\varphi_0)^{(\alpha-1)} = 0$ from (15) and remove the odd numbers from the solutions. We listed the PCs $h(x) = 1 + x^{2\ell}$, $1 \leq \ell \leq 4$, and corresponding $a(x)$s in Table VI.

TABLE VI: $f(x) = 1 + x^2$

$\alpha(x) = \sum_{i=0}^{\ell-1} x^{2i}$	$h(x) = 1 + x^{2\ell}$
1	$1 + x^2$
$1 + x^2$	$1 + x^4$
$1 + x^2 + x^4$	$1 + x^6$
$1 + x^2 + x^4 + x^6$	$1 + x^8$

Example 4. $f(x) = 1 + x^2 + x^3 + x^4 + x^6$

$f(x)$ can be decomposed into the multiplication of two prime polynomials $f_0(x) = 1 + x + x^2$ and $f_1(x) = 1 + x + x^2 + x^3 + x^4$, and we know from Example 1 and Example 2 that $\epsilon_0 = 3$ and $\epsilon_1 = 5$, respectively. Thus, α for the weight-2 PCs $h(x) = 1 + x^\alpha$ should be a multiple of ϵ_0 and ϵ_1, hence, a multiple of $\mathrm{lcm}(\epsilon_0, \epsilon_1) = 15$, and there are no weight-3 PCs associated with $f(x)$ since $f_1(x)$ does not yield any weight-3 PC as shown in Example 2. Two weight-2 PCs can be found from Table VII with corresponding $a(x)$s.

TABLE VII: $f(x) = 1 + x^2 + x^3 + x^4 + x^6$

$\alpha(x) = \sum_{i=0}^{\ell-1} x^{15i}(1 + x^2 + x^3 + x^6 + x^7 + x^9)$	$h(x) = 1 + x^{15\ell}$
$1 + x^2 + x^3 + x^6 + x^7 + x^9$	$1 + x^{15}$
$1 + x^2 + x^3 + x^6 + x^7 + x^9 + x^{15} + x^{17} + x^{18} + x^{21} + x^{22} + x^{24}$	$1 + x^{30}$

Example 5. $f(x) = 1 + x + x^5$

For this case, the polynomial can be decomposed into the multiplication of $f_0(x) = 1+x+x^2$ and $f_1(x) = 1 + x^2 + x^3$. We know from Example 1 that $\epsilon_0 = 3$ and $\epsilon_1 = 7$. Similar to Example 4, the weight-2 PCs have the general form of $h(x) = 1 + x^{21\ell}$ while the corresponding $a(x)$ can be expressed as $\sum_{i=0}^{\ell-1} x^{21i}(1 + x^2 + x^3 + x^4 + x^6 + x^8 + x^4 + x^6 + x^8 + x^{11} + x^{12} + x^{16})$.

In order to determine weight-3 PCs, we rewrite $\mathcal{M}$ and $\mathcal{N}$ in Example 1 as $\mathcal{M}^0$ and $\mathcal{N}^0$, respectively, and referring to Table VIII, let

$$\begin{aligned}
&\mathcal{M}_0^1 := \{7\ell + 1\}_{\ell \geq 0},\ \mathcal{N}_0^1 := \{7\ell + 5\}_{\ell \geq 0} \\
&\mathcal{M}_1^1 := \{7\ell + 2\}_{\ell \geq 0},\ \mathcal{N}_1^1 := \{7\ell + 3\}_{\ell \geq 0} \\
&\mathcal{M}_2^1 := \{7\ell + 4\}_{\ell \geq 0},\ \mathcal{N}_2^1 := \{7\ell + 6\}_{\ell \geq 0}
\end{aligned} \tag{21}$$

TABLE VIII: Non-zero Elements of $\mathrm{GF}(2^3)$ generated by $1 + x^2 + x^3$

power representation	polynomial representation
$x^0 = x^7$	1
x	x
x^2	x^2
x^3	$1 + x^2$
x^4	$1 + x + x^2$
x^5	$1 + x$
x^6	$x + x^2$

Then, we have

$$(\alpha_0,\ \beta_0) \in \mathcal{M}^0 \otimes \mathcal{N}^0$$

and

$$(\alpha_1,\ \beta_1) \in \bigcup_{i=0}^{2} \mathcal{M}_i^1 \otimes \mathcal{N}_i^1$$

Therefore, by taking the intersection, we can identify $(\alpha, \beta) \in (\mathcal{M}^0 \otimes \mathcal{N}^0) \bigcap \left(\bigcup_{i=0}^{2} \mathcal{M}_i^1 \otimes \mathcal{N}_i^1\right)$. As an example, we listed three valid $h(x) = 1 + x^\alpha + x^\beta$ in Table IX with corresponding $a(x)$s.

TABLE IX: $f(x) = 1 + x + x^5$

$a(x)$	$h(x)$
1	$1 + x + x^5$
$1 + x + x^5$	$1 + x^2 + x^{10}$
$1 + x + x^2 + x^3 + x^4 + x^6 + x^8$	$1 + x^{11} + x^{13}$

V. VALIDITY CONFIRMATION THROUGH THE UNION BOUND

In this section, we obtain a union bound using the low-weight codeword components pattern list and compare it to the union bound obtained via the transfer function as well as simulation results in order to confirm the validity of our proposed method.

A. A novel union bound

To obtain a union bound, let $\mathcal{A}_h(d)$ be the set of all $a(x)$ which yields weight-d PCs *i.e.*, $w_H(h(x)) = w_H(a(x)f(x)) = d$ for $a(x) \in \mathcal{A}_h(d)$. Similarly, we also define $\mathcal{A}_b(d)$ and $\mathcal{A}_c(d)$ for weight-d SCs and codewords, respectively, in the same manner.

Then, for $w_H(b(x)), w_H(h(x)) \geq 2$, we have from (8) that

$$\mathcal{A}_c(d) = \bigcup_{\ell=2}^{d-2} \{\mathcal{A}_b(\ell) \cap \mathcal{A}_h(d-\ell)\} \tag{22}$$

However, to determine $\mathcal{A}_b(\ell)$ or $\mathcal{A}_h(\ell)$ for a large ℓ is a complex task in general. Thus, in this paper, we replace the set $\mathcal{A}_c(d)$ by the following approximated set

$$\mathcal{A}_c(d) \approx \mathcal{A}'_c(d) = \left\{\bigcup_{\ell=2}^{\ell+1} \{\mathcal{A}_b(\ell) \cap \mathcal{A}_h(d-\ell)\}\right\} \bigcup \left\{\bigcup_{\ell=2}^{\ell+1} \{\mathcal{A}_b(d-\ell) \cap \mathcal{A}_h(\ell)\}\right\} \tag{23}$$

where some codewords in $\mathcal{A}_c(d)$ with $\ell \approx d-\ell$ may be ignored in $\mathcal{A}'_c(d)$. Finally, we obtain the following union bound

$$P_b \leq \frac{1}{k} \sum_{d=d_{\text{free}}}^{d_{\max}} \sum_{a(x)\in\mathcal{A}'_c(d)} w_H(a(x)g(x)) Q\left(\sqrt{\frac{2dE_c}{N_0}}\right) \tag{24}$$

where we let $d_{\max} = d_{\text{free}} + 3$.

In order to determine the low-weight codewords, based on $f(x)$, we first generate $\mathcal{A}_h(2) \cup \mathcal{A}_h(3)$, the set consisting of the weight-2 and -3 PCs (if they exist) using our proposed method described in the previous section. After that, we obtain the set consisting of SCs $b(x) = a(x)g(x)$ for $a(x) \in \mathcal{A}_h(2) \cup \mathcal{A}_h(3)$. Similarly, based on $g(x)$, we also obtain the set consisting of $h(x) = a(x)f(x)$ for $a(x) \in \mathcal{A}_s(2) \cup \mathcal{A}_s(3)$. Then, the resulting table consists of SCs and PCs such that $w_H(b(x)) + w_H(h(x)) \leq d_{\max}$.

Example 6. 5/7 RSC code ($f(x) = 1 + x^2,\ g(x) = 1 + x + x^2$)

The characteristics of $f(x)$ as well as the low-weight PCs it generates are shown in Example 3 whiles the characteristics of $g(x)$ as well as the low-weight SCs are shown in Example 1. The SCs and PCs of the 5/7 RSC code are listed in Table X.

TABLE X: SCs and PCs for the $5/7$ RSC code

$w_H(c(x))$	$a(x)$	$b(x)$	$h(x)$
5	1	$1+x+x^2$	$1+x^2$
6	$1+x$	$1+x^3$	$1+x+x^2+x^3$
	$1+x^2$	$1+x+x^3+x^4$	$1+x^4$
	$1+x+x^2$	$1+x^2+x^4$	$1+x+x^3+x^4$
	$1+x+x^3$	$1+x^4+x^5$	$1+x+x^2+x^5$
7	$1+x^2+x^3$	$1+x+x^5$	$1+x^3+x^4+x^5$
	$1+x^2+x^4$	$1+x+x^3+x^5+x^6$	$1+x^6$
8	$1+x+x^3+x^4$	$1+x^6$	$1+x+x^2+x^4+x^5+x^6$
	$1+x^2+x^4+x^6$	$1+x+x^3+x^5+x^7+x^8$	$1+x^8$

Example 7. 37/21 RSC code $(f(x)=1+x+x^2+x^3+x^4,\ g(x)=1+x^4)$

The SCs and PCs for this RSC code are listed in Table XI. The low-weight SCs and PCs were obtained using the methods in Examples 3 and 2, respectively.

TABLE XI: SCs and PCs for the $37/21$ RSC code

$w_H(c(x))$	$a(x)$	$b(x)$	$h(x)$
6	$1+x$	$1+x+x^4+x^5$	$1+x^5$
7	1	$1+x^4$	$1+x+x^2+x^3+x^4$
8	$1+x+x^5+x^6$	$1+x+x^4+x^6+x^9+x^{10}$	$1+x^{10}$

Example 8. $23/35$ RSC code $(f(x)=1+x+x^4,\ g(x)=1+x^2+x^3+x^4)$

Using the procedure shown in Examples 1 and 5, we obtained the low-weight PCs and SCs, respectively. The corresponding SCs and PCs for the 23/35 RSC code are listed Table XI.

B. Numerical results

We verify the validity of our proposed method for the $5/7,\ 37/21$ and $23/35$ RSC codes, assuming a frame size of $N=64$ is used to obtain the simulation results.

From Table X, we observe that every SC such that $w_H(b(x)) > 3$ is either a combination of only weight-2 SCs or only weight-3 SCs or both. This means that when the 5/7 RSC code is used in a TC, the deterministic interleaver should be designed in such a way that it deals

TABLE XII: SCs and PCs for the $23/35$ RSC code

$w_H(c(x))$	$a(x)$	$b(x)$	$h(x)$
7	1	$1+x^2+x^3+x^4$	$1+x+x^4$
	$1+x^2+x^3$	$1+x^7$	$1+x+x^2+x^6+x^7$
9	$1+x+x^2+x^3+x^5$	$1+x+x^3+x^4+x^8+x^9$	$1+x^7+x^9$
	$1+x+x^2+x^3+x^5+x^7+x^8$	$1+x+x^3+x^4+x^7+x^{12}$	$1+x^{11}+x^{12}$
10	$1+x^2+x^3+x^7+x^9+x^{10}$	$1+x^{14}$	$1+x+x^2+x^6+x^8+x^9+x^{13}+x^{14}$

effectively with both weight-2 and weight-3 SCs. While, having to consider weight-3 SCs in the deterministic interleaver design introduces a bit of complexity, it is manageable since there is just a single (m,n) pair that is associated with the weight-3 SCs.

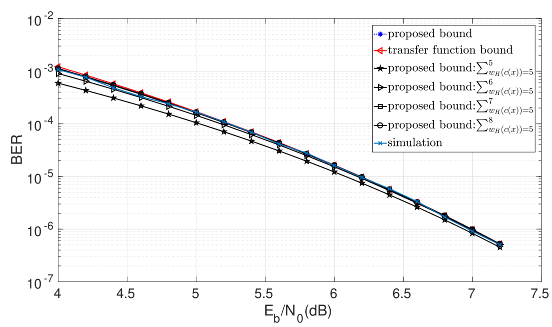

Fig. 1: Old Bound vs New Bound vs Simulation for 5/7 RSC Code

Fig. 1 shows the simulation results for the $5/7$ RSC code as well as the union bound obtained using the transfer function and our proposed method. We can observe that the accuracy of our proposed bound increases with the number of terms used in the approximation. Also, as E_b/N_0 increases, the proposed bound as well as the transfer function bound and the simulation results tend to converge. The fact that just a single (m,n) pair needs to be considered for weight-3 SCs during interleaver design makes the 5/7 RSC code attractive for use in TCs.

Table XI confirms the non-existence of weight-3 SCs and PCs in the 37/21RSC code, and that every SC such that $w_H(b(x)) > 2$ is a combination of weight-2 SCs. As such, deterministic interleaver design for this RSC code is relatively simpler, since only weight-2 SCs need to be considered.

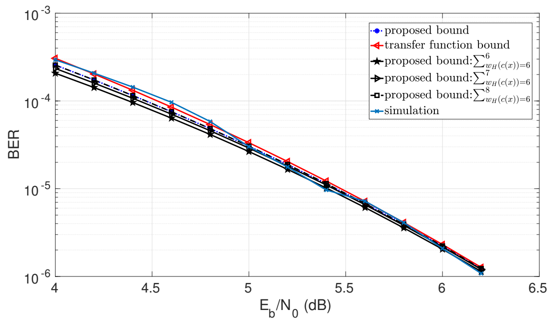

Fig. 2: Old Bound vs New Bound vs Simulation for 37/21 RSC Code

Simulation results, the transfer function bound and our proposed bound are shown in Fig. 2 for the $37/21$ RSC code. By observation, we can draw conclusions similar to Example V-B with respect to the accuracy of our proposed bound . Given that weight-2 SCs and PCs are sufficient to derive the union bound, and deterministic interleaver design requires focusing on weight-2 SCs only, this RSC code is highly recommended for use in TCs.

From Table XII, we observe that there are no weight-3 SCs. However, there exists weight-4 SCs that are not a combination of weight-2 SCs and when this RSC code is used in TCs, the deterministic interleaver needs to be designed to cater for weight-2 SCs as well as such weight-4 SCs.

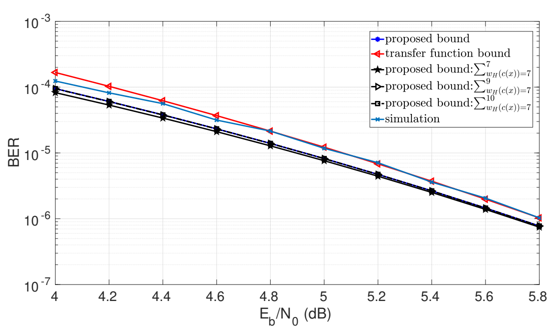

Fig. 3: Old Bound vs New Bound vs Simulation for 23/35 RSC Code

Fig. 3 shows the simulation results for the $23/35$ RSC code as well as the union bounds obtained using the transfer function as well as our proposed method. Even though the accuracy of our proposed bound increases with the number of terms use, it neither converges with the transfer function bound nor the simulation results as E_b/N_0 increases. Even though it is possible to improve the accuracy of our proposed bound by considering SCs and PCs of weight-4, the added complexity as a result of considering weight-4 SCs in the interleaver design process makes this RSC code very unattractive for use in TCs.

VI. Conclusion

In this paper, we proposed a novel method for listing the patterns of the SCs and PCs ($2 \leq w_H(b(x)), w_H(h(x)) \leq 3$) of a low-weight RSC codeword, given the RSC code and a codeword cut-off weight, d_{max}. Compared to the transfer function method, it has low complexity and the knowledge of the SC and PC patterns makes it a very useful for interleaver design. To validate our method, we compared the union bound obtained using our novel method with the union bound obtained via the transfer function as well as the simulation results for three RSC codes. Results suggest that RSC codes that can be sufficiently characterized by our novel method are much more attractive for use in TCs due to lower complexity required in the deterministic interleaver design.

References

[1] C. Berrou, A. Glavieux and P. Thitimajshima, ”Near Shannon limit error-correcting coding and decoding: Turbo-codes. 1,” Proceedings of ICC ’93 - IEEE International Conference on Communications, Geneva, Switzerland, 1993, pp. 1064-1070 vol.2, doi: 10.1109/ICC.1993.397441.

[2] John G. Proakis, Masoud Salehi. “Digital Communications”, Fifth Edition,Chapter 8, McGraw-Hill.

[3] Todd K. Moon. “Error Correcting Codes”,Chapter 12, John Wiley & Sons.

[4] Alain Glavieux, “Channel Coding in Communication Networks: From Theory to Turbocodes”, Chapter 3, John Wiley & Son.

[5] Jing Sun and O. Y. Takeshita, ”Interleavers for turbo codes using permutation polynomials over integer rings,” in IEEE Transactions on Information Theory, vol. 51, no. 1, pp. 101-119, Jan. 2005, doi: 10.1109/TIT.2004.839478.

[6] R. Garzn-Bohrquez, C. Abdel Nour and C. Douillard, ”Protograph-Based Interleavers for Punctured Turbo Codes,” in IEEE Transactions on Communications, vol. 66, no. 5, pp. 1833-1844, May 2018, doi: 10.1109/TCOMM.2017.2783971.

[7] S. Lu, W. Hou and J. Cheng, Input-output weight distribution of terminated RSC codes with limited codelength, 2016 International Symposium on Information Theory and Its Applications (ISITA), Monterey, CA, USA, 2016, pp. 493-497.

[8] J. Deng, Y. Peng and H. Zhao, Distance spectrum calculation method for double binary turbo codes, 2017 International Conference on Recent Advances in Signal Processing, Telecommunications and Computing (SigTelCom), Da Nang, 2017, pp. 98-102, doi: 10.1109/SIGTELCOM.2017.7849803.