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

Kwame Ackah Bohulu

April 23, 2021

0.1 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.

0.2 Introduction

1. The *turbo code* (TC) [1], introduced by Claude Berrou
   - Excellent *forward-error correcting* (FEC) code
   - Applications include :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].
2. Construction of a TC
   - Two *recursive systematic convolutional* (RSC) codes concatenated via an interleaver
   - Good performance of TC due to interleaver.
3. Good deterministic interleaver design requirements?
   - complete knowledge of all the low-weight codeword component patterns in the RSC code
   - missing even one of these patterns can create sub-par interleaver and TC
4. Interleaver Design Tools?
   - The transfer function of an RSC code provides information about the distance spectrum.
   - No information with regards to the pattern of the low-weight codeword components.
   - Complexity increases with the number of states of RSC code.
   - No interleaver design tool reveals distance spectrum and the low-weight codeword component patterns.
   - As a result, many interleaver design methods completely ignoring certain important low-weight codewords, example [5].
5. Acheivements of this research
   - We propose a novel method for revealing the pattern of the low-weight codeword components.
   - Excellent interleaver design tool
   - Complexity independent of RSC code states

0.2.1 Notations

1. least common multiple of integers α and β : $\mathrm{lcm}(\alpha, \beta)$
2. remainder α divided by β, $\beta \neq 0$: $\alpha \mod \beta$
3. α is a divisor of β: $\alpha|\beta$
4. shorthand for the operation $(\alpha \text{ mod } \epsilon_0,\ \beta \text{ mod } \epsilon_0)$: $(\alpha,\ \beta)$ mod , $(\alpha,\ \beta)$ are integer pairs.
5. tensor product that yields the set consisting of all pairs of $\mathcal{M}$ and $\mathcal{N}$: $\mathcal{M} \otimes \mathcal{N}$, $\mathcal{M}$ and $\mathcal{N}$ are integer sets.

0.3 Preliminaries

1. 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 $v_m,\ 0 \leq m \leq M$, are called the *coefficients* and $v_M \neq 0$. If $v_M = 1,\ v(x)$ is called a *monic* polynomial.

2. We call the total number of the non-zero coefficients the *Hamming weight* of $v(x)$, denoted as $w_H(v(x))$.

3. 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 $\mod p$, 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* $\mathrm{GF}(p)$.

4. 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.

5. We say a monic polynomial is a *prime polynomial* if it cannot be obtained by the multiplication of some lower degree polynomials.

6. 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 $w(x)$.

7. 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 $\mathrm{GF}(p)$.

8. Each non-zero element in $\mathrm{GF}\left(p^M\right)$ can be represented by a power of x uniquely as $x^m,\ 0 \le m \le p^M - 1$.

9. 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*.

10. 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*.

11. 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 $\varphi^i,\ 0 \le i \le \epsilon - 1$, are all distinct from each other.

0.4 The characteristics of the low-weights codewords of RSC code

1. 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.

2. 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.

3. 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.

4. 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.

5. Low-weight PCs occur if and only if
$$b(x) \bmod g(x) \equiv 0 \tag{9}$$

6. 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

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

7. 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 $2 \leq w_H(h(x)) \leq 3$.

0.5 The patterns of the low-weight PCs

1. 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

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

2. 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 the cases $\gamma_0 = 1$ and $\gamma_0 > 1$.

3. 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.

4. Thus, for the case $\gamma_0 > 1$, we obtain extra conditions using differential equations as

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

where the derivation is calculated using the *Hasse derivative* defined as

$$\frac{d^j x^k}{dx^j} = \begin{cases} ({}_kC_j \mod 2)\, x^{k-j}, & k \ge j \\ 0, & \text{otherwise} \end{cases} \tag{16}$$

for the bionomial coefficient ${}_kC_j$.

5. 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.

0.5.1 The weight-2 PCs

Each weight-2 PC can be written as

$$h(x) = 1 + x^{\alpha} \tag{17}$$

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

$$(\varphi_0^i)^{\alpha} = 1, \quad 0 \le i < \epsilon_0 \tag{18}$$

On the other hand, the order of φ_0, ϵ_0 is the least integer satisfying $\varphi_0^{\epsilon_0} = 1$, thus, α should satisfy the condition

$$\alpha \equiv 0 \mod \epsilon_0 \quad \text{or} \quad \epsilon_0 | \alpha \tag{19}$$

0.5.2 The weight-3 PCs

Without loss of generality, theh weight-3 PCs can be written as

$$h(x) = 1 + x^{\alpha} + x^{\beta}, \ \alpha < \beta \tag{20}$$

and hence, (α, β) should satisfy the condition

$$\varphi_0^{\alpha} + \varphi_0^{\beta} = 1 \tag{21}$$

The pairs satisfying (21) can be found by referring to the table of the extended field for GF $\left(2^M\right)$. Let (m, n) be such a pair, and we let $\mathcal{M} = \{\epsilon_0 \ell + m\}$ and $\mathcal{N} = \{\epsilon_0 \ell + n\}$, $\ell \ge 0$. Then it is obvious that each pair $(\alpha,\ \beta) \in \mathcal{M} \otimes \mathcal{N}$ satisfies (21). For a fixed α, on the other hand, since $\alpha + i$, $0 \le i < \epsilon_0$, are distinct from each other, any integer β that satisfies (21) must be such that $n \equiv \beta \mod \epsilon_0$.

0.5.3 Examples

In the followings, we present some examples of the proposed method to determine weight-2 and -3 PCs for several feedfoward polynomials of form given in (12). For the case $K = 1$, Examples 1 and 2 are two instances that $f(x)$ is primitive polynomial while an instance of $f(x)$ is prime but primitive polynomial is given in Example 3. Example 4 demonstrate the case $\gamma_0 > 1$, and Examples 5 and 6 are two instances of the case $K = 2$. For these polynomials, we also show some detail low-weight patterns of $a(x)$ and $h(x)$ in Table 1.

$f(x)$ is a primitive polynomial

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 in the weight-2 PCs shown in (17) should be a multiple of 3 as $h(x) = 1 + x^{3\ell}$, $\ell \in \mathbb{Z}^+$, while the corresponding $a(x)$ can be expressed by

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

To determine the weight-3 PCs, we can see from Table 2 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 $\mathcal{N} = \{3\ell + 2\}_{\ell\geq 0}$, $x^\alpha + x^\beta \equiv 1 \mod f(x)$ for each pair $(\alpha, \beta) \in \mathcal{M} \otimes \mathcal{N}$.

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

Since $f(x)$ is a primitive polynomial with a root of order $\epsilon_0 = 2^M - 1 = 15$, the weight-2 PCs have the following general form

$$h(x) = 1 + x^{15\ell}$$

while the corresponding $a(x)$ can be expressed as

$$a(x) = \sum_{i=0}^{\ell} x^{15i} \left(1 + x + x^2 + x^3 + x^5 + x^7 + x^8 + x^{11}\right)$$

For the weight-3 PCs, we refer to Table 2 and observe that there are 7 (m, n) pairs which satisfy $x^m + x^n \equiv 1 \mod 15$. Thus, if we let

$$\begin{aligned}
&\mathcal{M}_0 := \{15\ell + 1\}_{\ell\geq 0},\ \mathcal{N}_0 := \{15\ell + 4\}_{\ell\geq 0} \\
&\mathcal{M}_1 := \{15\ell + 2\}_{\ell\geq 0},\ \mathcal{N}_1 := \{15\ell + 8\}_{\ell\geq 0} \\
&\mathcal{M}_2 := \{15\ell + 3\}_{\ell\geq 0},\ \mathcal{N}_2 := \{15\ell + 14\}_{\ell\geq 0} \\
&\mathcal{M}_3 := \{15\ell + 5\}_{\ell\geq 0},\ \mathcal{N}_3 := \{15\ell + 10\}_{\ell\geq 0} \\
&\mathcal{M}_4 := \{15\ell + 6\}_{\ell\geq 0},\ \mathcal{N}_4 := \{15\ell + 13\}_{\ell\geq 0} \\
&\mathcal{M}_5 := \{15\ell + 7\}_{\ell\geq 0},\ \mathcal{N}_5 := \{15\ell + 9\}_{\ell\geq 0} \\
&\mathcal{M}_6 := \{15\ell + 11\}_{\ell\geq 0},\ \mathcal{N}_6 := \{15\ell + 12\}_{\ell\geq 0}
\end{aligned} \tag{22}$$

each pair $(\alpha,\ \beta) \in \bigcup_{i=0}^{6} \mathcal{M}_i \otimes \mathcal{N}_i$ satisfies (20).

$f(x)$ is a prime but not primitive polynomial

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

Since $x \equiv x^5 \mod f(x)$ as shown in Table 2, $\epsilon_0 = 5 < 15$ and the weight-2 PCs can be expressed as $h(x) = 1 + x^{5\ell}$, $\ell \in \mathbb{Z}^+$. For weight-3 PCs, on the other hand, Table 2 indicates that there is no pair $(m,\ n)$ satisfying $x^m + x^n \equiv 1$, and hence, the given $f(x)$ does not yield any weight-3 PCs.

$K = 1$ and $\gamma_0 > 1$

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

If we rewrite the polynomials as $f(x) = (1 + x)^2$ and $f(x) = (1 + x)^4$, the order of the root φ_0 is $\epsilon_0 = 1$. Since GF(2) has single non-zero element, it does not provide a pair $(m,\ n)$ satisfying $x^m + x^n = 1$ and, consequently, there are no weight-3 PCs associated with $f(x)$.

Regarding the weight-2 PCs, since $\epsilon_0 = 1$, each $\alpha \in \mathbb{Z}^+$ satisfies

$$h(x) = 1 + x^\alpha = 0 \tag{23}$$

However, the following second order differential equation

$$\frac{dh(x)}{dx} = (\alpha \mod 2)x^{\alpha-1} = 0 \tag{24}$$

implies α should be even numbers. Thus, for the case $f(x) = 1 + x^2$, we write the PCs as $h(x) = 1 + x^{2\ell}$, $\ell \in \mathbb{Z}^+$.

For the case $f(x) = 1 + x^4$, from (15), we have

$$\begin{cases} \frac{d^2h(x)}{dx^2} = \left[\frac{\alpha(\alpha-1)}{2} \mod 2\right] x^{\alpha-2} = 0 \\ \frac{d^3h(x)}{dx^3} = \left[\frac{\alpha(\alpha-1)(\alpha-2)}{6} \mod 2\right] x^{\alpha-3} = 0 \end{cases} \tag{25}$$

and $\alpha \in 4\mathbb{Z}^+$ satisfies (25) simultaneously.

The case $K = 2$

For this case, we write the feedforward polynomial as $f(x) = f_0(x)f_1(x)$ and give two exmaples.

Example 5. $f(x) = (1 + x)(1 + x + x^3) = 1 + x^2 + x^3 + x^4$

Let $f_0(x) = 1 + x$ and $f_1(x) = 1 + x + x^3$. We know that the PCs associated with $f(x)$ are intersection of those with $f_0(x)$ and with $f_1(x)$. On the other hand, since $f_0(x)$ does not yields any weight-3 PCs as explained in the Example 4, there are no such PCs associated with $f(x)$.

In respect to the weight-2PCs, from $\epsilon_0 = 1$ and $\epsilon_1 = 7$, we have $\mathrm{lcm}(\epsilon_0, \epsilon_1) = 7$ and

$$h(x) = 1 + x^{7\ell}$$

with the corresponding $a(x)$ given by

$$a(x) = \sum x^{7i}(1 + x^2 + x^3)$$

Example 6. $f(x) = (1 + x + x^2)(1 + x^2 + x^3) = 1 + x + x^5$

For this case, it is not difficult to see that $\epsilon_0 = 3$ and $\epsilon_1 = 7$ for $f_0(x) = 1 + x + x^2$ and $f_1(x) = 1 + x^2 + x^3$, respectively. Thus, from $\mathrm{lcm}(\epsilon_0, \epsilon_1) = 21$, the weight-2 PCs have the general form of $h(x) = 1 + x^{21\ell}$ while the corresponding $a(x)$ can be expressed as

$$a(x) = \sum_{i=0}^{\ell-1} x^{21i}(1 + x^2 + x^3 + x^4 + x^6 + x^8 + x^4 + x^6 + x^8 + x^{11} + x^{12} + x^{16})$$

.

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 2, let

$$\begin{aligned} \mathcal{M}_0^1 &:= \{7\ell + 1\}_{\ell \geq 0},\ \mathcal{N}_0^1 := \{7\ell + 5\}_{\ell \geq 0} \\ \mathcal{M}_1^1 &:= \{7\ell + 2\}_{\ell \geq 0},\ \mathcal{N}_1^1 := \{7\ell + 3\}_{\ell \geq 0} \\ \mathcal{M}_2^1 &:= \{7\ell + 4\}_{\ell \geq 0},\ \mathcal{N}_2^1 := \{7\ell + 6\}_{\ell \geq 0} \end{aligned} \tag{26}$$

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 $(\alpha, \beta) \in (\mathcal{M}^0 \otimes \mathcal{N}^0) \bigcap \left(\bigcup_{i=0}^{2} \mathcal{M}_i^1 \otimes \mathcal{N}_i^1\right)$.

Table 1: $a(x)$ and $h(x)$ for various $f(x)$

$f(x)$	weight	$a(x)$	$h(x)$
$1+x+x^2$ (Ex. 1)	2	$1+x$	$1+x^3$
		$1+x+x^3+x^4$	$1+x^6$
		$1+x+x^3+x^4+x^6+x^7$	$1+x^9$
		$1+x+x^3+x^4+x^6+x^7+x^9+x^{10}$	$1+x^{12}$
	3	1	$1+x+x^2$
		$1+x+x^2$	$1+x^2+x^4$
		$1+x+x^3$	$1+x^4+x^5$
		$1+x^2+x^3$	$1+x+x^5$
$1+x+x^4$ (Ex. 2)	2	$1+x+x^2+x^3+x^5+x^7+x^8+x^{11}$	$1+x^{15}$
	3	1	$1+x+x^4$
		$1+x+x^4$	$1+x^2+x^8$
		$1+x+x^2+x^3+x^5$	$1+x^7+x^9$
		$1+x+x^2+x^3+x^6$	$1+x^5+x^{10}$
$1+x+x^2+x^3+x^4$ (Example 3)	2	$1+x$	$1+x^5$
		$1+x+x^5+x^6$	$1+x^{10}$
		$1+x+x^5+x^6+x^{10}+x^{11}$	$1+x^{15}$
		$1+x+x^5+x^6+x^{10}+x^{11}+x^{15}+x^{16}$	$1+x^{20}$
$1+x^2$ (Example 4)	2	1	$1+x^2$
		$1+x^2$	$1+x^4$
		$1+x^2+x^4$	$1+x^6$
		$1+x^2+x^4+x^6$	$1+x^8$
$1+x^4$ (Example 4)	2	1	$1+x^4$
		$1+x^4$	$1+x^8$
		$1+x^4+x^8$	$1+x^{12}$
		$1+x^4+x^8+x^{12}$	$1+x^{16}$
$1+x^2+x^3+x^4$ (Example 5)	2	$1+x+x^2$	$1+x^7$
		$1+x^2+x^3+x^7+x^9+x^{10}$	$1+x^{14}$
$1+x+x^5$ (Example 6)	2	$1+x^2+x^3+x^4+x^6+x^8+x^4+x^6+x^8+x^{11}+x^{12}+x^{16}$	$1+x^{21}$
	3	1	$1+x+x^5$
		$1+x+x^5$	$1+x^2+x^{10}$
		$1+x+x^2+x^3+x^4+x^6+x^8$	$1+x^{11}+x^{13}$

Table 2: Galois Field Elements for various prime polynomials $f(x)$

Power Representation	polynomial representation		
Generator polynomial	$1+x^2+x^3$	$1+x+x^2+x^3+x^4$	$1+x+x^4$
x^0	1	1	1
x	x	x	x
x^2	x^2	x^2	x^2
x^3	$1+x^2$	x^3	x^3
x^4	$1+x+x^2$	$1+x+x^2+x^3$	$1+x$
x^5	$1+x$		$x+x^2$
x^6	$x+x^2$		x^2+x^3
x^7			$1+x+x^3$
x^8			$1+x^2$
x^9			$x+x^3$
x^{10}			$1+x+x^2$
x^{11}			$x+x^2+x^3$
x^{12}			$1+x+x^2+x^3$
x^{13}			$1+x+x^3$
x^{14}			$1+x^3$

0.6 Validity Confirmation through the Union Bound

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

0.6.1 A novel 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 $a(x) \in \mathcal{A}_h(d)$. We also define $\mathcal{A}_b(d)$ and $\mathcal{A}_c(d)$ as the sets of all $a(x)$s which result weight-d SCs and codewords, respectively.

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

$$\mathcal{A}_c(d) = \bigcup_{\ell=2}^{d-2} \{\mathcal{A}_b(\ell) \cap \mathcal{A}_h(d-\ell)\} \tag{27}$$

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

$$\mathcal{A}_c(d) \approx \mathcal{A}'_c(d) = \left\{ \bigcup_{\ell=2}^{\ell+1} \{\mathcal{A}_b(\ell) \cap \mathcal{A}_h(d-\ell)\} \right\} \bigcup \left\{ \bigcup_{\ell=2}^{\ell+1} \{\mathcal{A}_b(d-\ell) \cap \mathcal{A}_h(\ell)\} \right\} \tag{28}$$

and obtain an approximated union bound as

$$P_b \leq \frac{1}{k} \sum_{d=d_{\text{free}}}^{d_{\text{free}+1}} \sum_{a(x) \in \mathcal{A}'_c(d)} w_H(a(x)g(x)) Q\left(\sqrt{\frac{2dE_c}{N_0}}\right) \tag{29}$$

Notice that since $\mathcal{A}_c(d)$ in (27) is replaced by $\mathcal{A}'_c(d)$, the contributions of the codewords with $\ell \approx d - \ell$ may be neglected in our approximation.

To obtain $\mathcal{A}'_c(d)$, based on $f(x)$, we first generate the set consisting of $a(x)$s which yield the weight-2 and -3 PCs, *i.e.* $\mathcal{A}_h(2) \cup \mathcal{A}_h(3)$. Next, for each $a(x) \in \mathcal{A}_h(2) \cup \mathcal{A}_h(3)$, we determine the corresponding SC $b(x) = a(x)g(x)$. Similarly, we determine the PC $h(x) = a(x)f(x)$ for each $a(x)$ in the set $\mathcal{A}_s(2) \cup \mathcal{A}_s(3)$ obtained based on $g(x)$. Finally, we narrow down the corresponding codewords as $w_H(b(x)) + w_H(h(x)) \leq d_{\text{free}+3}$ for $a(x) \in \mathcal{A}_h(2) \cup \mathcal{A}_h(3) \cup \mathcal{A}_s(2) \cup \mathcal{A}_s(3)$.

As examples, in Table 4,5 and 6, we listed the low-weight PCs and SCs founded by our proposed method for the codes listed in Table 3 with the corresponding example numbers where each polynomial appeared in.

Table 3: The generator polynomials

	$f(x)$	$g(x)$
Code I (5/7)	$1+x^2$ (Ex. 4)	$1+x+x^2$ (Ex. 1)
Code I (37/21)	$1+x+x^2+x^3+x^4$ (Ex. 3)	$1+x^4$ (Ex. 4)
Code III (23/35)	$1+x+x^4$ (Ex. 2)	$1+x^2+x^3+x^4$ (Ex. 5)

0.6.2 Numerical results

In order to verify the validity of the proposed method, for the codes listed in Table 3, we obtained the approximated union bound by (29). Since the codewords with the weights larger than $d_{\text{free+3}}$ are neglected in our approximation, we also obtained the union bounds obtained using the codewords with weights up to $d_{\text{free+i}}$, $0 \leq i \leq 3$, and compared them with that obtainded using transfer function in Figures 1-3. In these figures, we also evaluated *bit error rate* (BER) through computer simulations. To plot BER points, we assume each RSC code is BPSK modulated and transmitted over the AWGN channel with a frame with size of $N = 64$. At the receiver, the Viterbi algorithm is used to recover the transmitted bits and we accumulated more than 1000 bits errors for obtain each plot point.

For the 5/7 RSC code, the detail SCs and PCs found by computer searching are also listed in Table 4 and union bounds and simulation results are shown in Fig. 1.

As shown in Table 4, since the free distance of the 5/7 RSC code is 5, and the codewords consisting of weigts 2 and 3 SCs or PCs are taken into account in the proposed method, all codewords with weights up to 7 are picked up. For the codewords of weight-8, on the other hand, some of that consisting of the weight-4 SC and PC are omitted in our method. However, Fig. 1 indicates that the union bound obtained using the codewords with weights up to $d_{\text{free+1}}$ are sufficient to approximate the BER curve especially in the high-E_b/N_0 region. Moreover, the bounds otained by our mehotd and the transfer function converge to the same with E_b/N_0 increament bound and match the simulation results.

For the 37/21 RSC code, since the free distance of the code is 6, the counting omission in the proposed method occurs for the codewords higher than 7. Although Table 5 indicates that there are 2 weight-8 codewords more than 1 of them founded in our method, we can see from Fig. 2 that the contribution of them on the union bound are negligible and the BER curve can be approximated using the codewords with weight 6 and 7 with a high accuracy at the high E_b/N_0 region.

For the code III, the free distance is 7. Using the proposed method, we can identfy 2 codewords with weight-7 while 3 codewords with weight-8 can not be found as shown in Table 6. Thus, while we use the weight-7 codewords to approximate the BER curve as Fig. 3, there about 0.1 dB between the proposed method and simulation results.

Table 4: SCs and PCs for Code I

$w_H(c(x))$		$a(x)$	$b(x)$	$h(x)$
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	Found	$1+x+x^3+x^4$	$1+x^6$	$1+x+x^2+x^4+x^5+x^6$
		$1+x^2+x^4+x^6$	$1+x+x^3+x^5+x^7+x^8$	$1+x^8$
	Not Found	$1+x+x^2+x^3$	$1+x+x^3+x^5$	$1+x+x^4+x^5$
		$1+x+x^2+x^4$	$1+x^2+x^5+x^6$	$1+x+x^3+x^6$
		$1+x+x^3+x^5$	$1+x^4+x^6+x^7$	$1+x+x^2+x^7$
		$1+x^2+x^3+x^4$	$1+x+x^4+x^6$	$1+x^3+x^5+x^6$
		$1+x^2+x^3+x^5$	$1+x+x^6+x^7$	$1+x^3+x^4+x^7$
		$1+x^2+x^4+x^5$	$1+x+x^3+x^7$	$1+x+x^6+x^7$

Table 5: SCs and PCs for Code II

$w_H(c(x))$		$a(x)$	$b(x)$	$h(x)$
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	Found	$1+x+x^5+x^6$	$1+x+x^4+x^6+x^9+x^{10}$	$1+x^{10}$
	Not Found	$1+x^2$	$1+x^2+x^4+x^6$	$1+x+x^5+x^6$
		$1+x+x^4+x^6$	$1+x+x^8+x^9$	$1+x^4+x^5+x^9$
9	Not Found	$1+x+x^4$	$1+x+x^5+x^8$	$1+x^4+x^6+x^7+x^8$
		$1+x^2+x^4$	$1+x^2+x^6+x^8$	$1+x+x^4+x^7+x^8$
		$1+x^3+x^4$	$1+x^3+x^7+x^8$	$1+x+x^2+x^4+x^8$
		$1+x+x^5$	$1+x+x^4+x^9$	$1+x^6+x^7+x^8+x^9$
		$1+x^4+x^5$	$1+x^5+x^8+x^9$	$1+x+x^2+x^3+x^9$

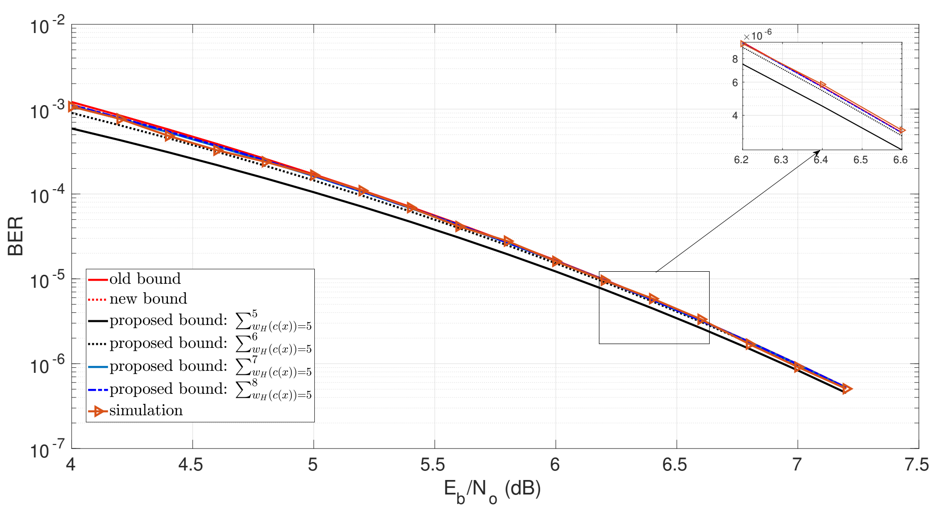

Figure 1: Old Bound vs New Bound vs Simulation for 5/7 RSC Code

Table 6: SCs and PCs for Code III

$w_H(c(x))$		$a(x)$	$b(x)$	$h(x)$
7		1	$1+x^2+x^3+x^4$	$1+x+x^4$
		$1+x^2+x^3$	$1+x^7$	$1+x+x^2+x^6+x^7$
8	Not Found	$1+x$	$1+x+x^2+x^5$	$1+x^2+x^4+x^5$
		$1+x+x^2+x^4$	$1+x+x^7+x^8$	$1+x^3+x^6+x^8$
		$1+x+x^2+x^4+x^6+x^7$	$1+x+x^6+x^{11}$	$1+x^3+x^{10}+x^{11}$
9	Found	$1+x+x^2+x^3+x^5$	$1+x+x^3+x^4+x^8+x^9$	$1+x^7+x^9$
		$1+x+x^2+x^3+x^5+x^7+x^8$	$1+x+x^3+x^4+x^7+x^{12}$	$1+x^{11}+x^{12}$
	Not Found	$1+x+x^2$	$1+x+x^4+x^6$	$1+x^3+x^4+x^5+x^6$
		$1+x+x^2+x^4+x^5$	$1+x+x^5+x^9$	$1+x^3+x^5+x^8+x^9$
10	Found	$1+x^2+x^3+x^7+x^9+x^{10}$	$1+x^{14}$	$1+x+x^2+x^6+x^8+x^9+x^{13}+x^{14}$
	Not Found	$1+x^2$	$1+x^3+x^5+x^6$	$1+x+x^2+x^3+x^4+x^6$
		$1+x+x^3$	$1+x+x^2+x^4+x^5+x^6+x^7$	$1+x+x^3+x^7$
		$1+x+x^2+x^3$	$1+x+x^3+x^4+x^5+x^7$	$1+x^5+x^6+x^7$
		$1+x^4$	$1+x^2+x^3+x^6+x^7+x^8$	$1+x+x^5+x^8$
		$1+x^2+x^3+x^5$	$1+x^5+x^8+x^9$	$1+x+x^2+x^5+x^7+x^9$
		$1+x^2+x^4+x^5$	$1+x^3+x^4+x^9$	$1+x+x^2+x^3+x^8+x^9$
		$1+x^2+x^3+x^6$	$1+x^6+x^7+x^8+x^9+x^{10}$	$1+x+x^2+x^{10}$
		$1+x+x^2+x^3+x^4+x^6$	$1+x+x^3+x^5+x^9+x^{10}$	$1+x^4+x^8+x^{10}$
		$1+x^3+x^5+x^6$	$1+x^2+x^4+x^{10}$	$1+x+x^3+x^5+x^9+x^{10}$
		$1+x+x^2+x^3+x^5+x^6$	$1+x+x^3+x^4+x^6+x^{10}$	$1+x^6+x^9+x^{10}$
		$1+x^2+x^3+x^7$	$1+x^9+x^{10}+x^{11}$	$1+x+x^2+x^6+x^8+x^{11}$
		$1+x^4+x^6+x^7$	$1+x^2+x^3+x^{11}$	$1+x+x^5+x^6+x^{10}+x^{11}$
		$1+x^2+x^3+x^5+x^7+x^8$	$1+x^5+x^7+x^{12}$	$1+x+x^2+x^5+x^{11}+x^{12}$
		$1+x^2+x^3+x^6+x^8+x^9$	$1+x^6+x^7+x^{13}$	$1+x+x^2+x^8+x^{12}+x^{13}$
		$1+x+x^2+x^3+x^4+x^6+x^8+x^9$	$1+x+x^3+x^5+x^8+x^{13}$	$1+x^4+x^{12}+x^{13}$

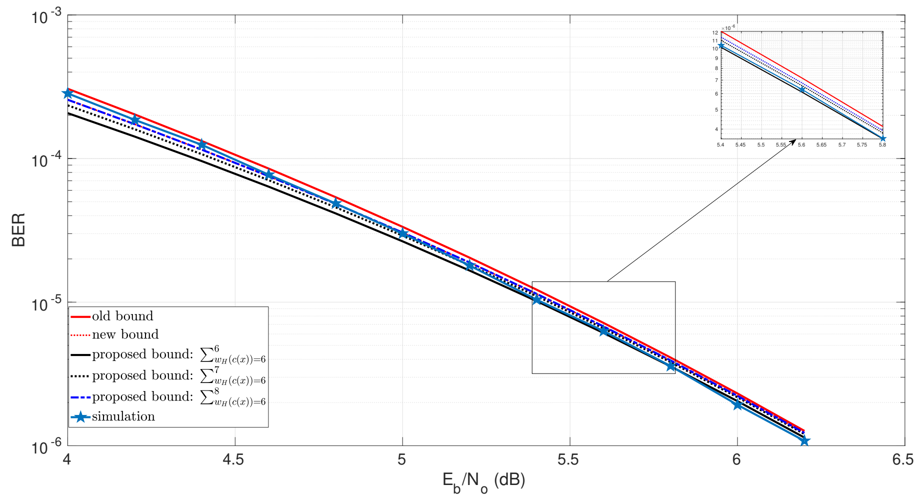

Figure 2: Old Bound vs New Bound vs Simulation for 37/21 RSC Code

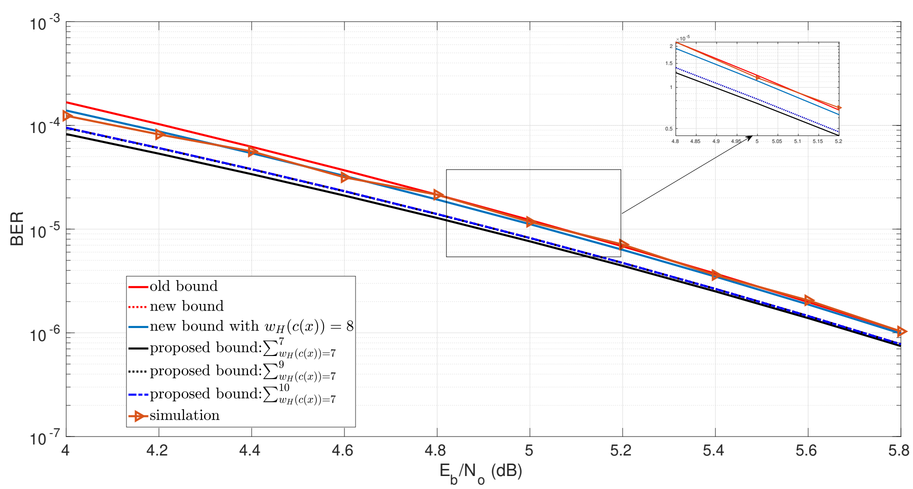

Figure 3: Old Bound vs New Bound vs Simulation for 23/35 RSC Code

0.7 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.

Bibliography

[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 Re-

cent Advances in Signal Processing, Telecommunications and Computing (SigTelCom), Da Nang, 2017, pp. 98-102, doi: 10.1109/SIGTELCOM.2017.7849803.