A Method to Obtain Complete Information about Low-Weight Codewords 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

For the *turbo code* (TC) consisting of *recursive systematic convolutional* (RSC) codes, the complete knowledge of the low-weight codewords of the component code is very important for interleaver design to achieve a good performance. In this paper, we present a method to obtain complete information about low-weight codewords for RSC codes. For a given RSC code, we first identify the codewords with weight-2 and weight-3 *parity-check components* (PCs) based on the characteristic of the feed-forward polynomial and similarly, we identify the codewords with low-weight *systematic components* (SCs) based on the feedback polynomial. The low-weight codewords are established from the identified codewords. To validate our proposed method, we obtain a union bound using the established low-weight codewords and compare it with that obtained via the transfer function and the *bit error rate* (BER) curve drawn from simulation results.

I. Introduction

The RSC code, developed by Punya Thitimajshima, was first introduced in the Claude Berrou's 1993 paper [1] as the *component code* (CC) for the then newly invented TC, which is one of the *forward-error correcting* (FEC) codes that comes very close to satisfying the Shannon limit for AWGN channels. 2 years after [1], [2] was published, giving more details on the RSC code. Subsequently, [3], which detailed the inner workings of parallel concatenated codes, proved why

RSC codes are better suited for use in the TC (better performance at any SNR for high code rate compared to other classes of convolutional codes) and in so doing, established them as the CC of choice for the TC.

Due to the success of the TC, it has been adopted by many industral standards [4], [5], [6] and the RSC code has been adopted as the CC for many other concatenated coding schemes. Examples include *recursive convolutional space-time codes* (ReC-STC) designed by adopting several parallel two-state RSC codes [7], a concatenated code obtained by combining a RSC code with a *Low-Density Parity-Check* (LDPC) code [8], *binary self-concatenated convolutional codes employing iterative decoding* (SECCC-ID) with RSC codes as the CC [9] and the *Reed-Solomon Convolutional Code* (RS-CC) concatenated code [10], [11], [12]. Despite the performance of these concatenated codes, the RSC code remains associated with the TC.

The TC is generally constructed by concatenating two RSC codes (usually of the same kind) parallely via an interleaver . A well designed TC realizes a large minimum distance via the interleaver, if it maps each bit sequence wit a low-weight PC in the first RSC code onto that with a high-weight PC in the second RSC code. Thus, the design of a good deterministic interleaver requires the complete knowledge of all the low-weight codeword component patterns of the employed RSC code and missing even one of these patterns may result in TC with subpar 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 [13] have to be used. Research into other methods for finding the distance spectrum have been carried out in recent years. In [14], an algorithm for evaluating the input-parity weight distribution of terminated RSC codes is presented, while in [15], 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 complete knowledge of the low-weight codewords. Because of this, many of the interleaver design methods end up completely ignoring certain important low-weight codewords. In [16] 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 method to obtain complete information about 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 also reveal the low-weight codeword patterns of an RSC code makes it an excellent interleaver design tool. We establish the low-weight codewords for the given RSC code by identifying codewords with either PCs or SCs of weight-2 and weight-3. Then, using the established low-weight codewords, we validate our proposed method by obtaining a union bound and comparing it to that obtained via the transfer function method and the BER curve obtained via simulation results.

The remainder of the research paper is organised as follows. Definitions used in this paper are introduced in Section II. In Section III, we discuss the characteristics of the low-weight RSC codewords and then present our method in Section IV. Validation of our proposed method for specific RSC codes as well as discussion of numerical results 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 $(\alpha,\ \beta)$, $(\alpha,\ \beta) \bmod \epsilon_0$ is shorthand for the operation $(\alpha \bmod \epsilon_0,\ \beta \bmod \epsilon_0)$. The sets of non-negative and positive integers are denoted by $\mathbb{Z}$ and $\mathbb{Z}^+$, respectively. For two integer sets $\mathbb{M}$ and $\mathbb{N}$, the tensor product that yields the set consisting of all pairs of $\mathbb{M}$ and $\mathbb{N}$ is denoted as $\mathbb{M} \otimes \mathbb{N}$ 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 $v_M \neq 0$ and $v_m,\ 0 \leq m \leq M$, are called the *coefficients*. If $v_M = 1,\ v(x)$ is called a *monic* polynomial. We say the total number of the non-zero coefficients of $v(x)$ is 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 $\mod p$, 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* $\mathrm{GF}(p)$.

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 represented by 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 polynomial 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 $b(x)$, states of the shift registers and 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 output $c(x)$ is 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. Therefore low-weight PCs occur if and only if

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

The bit sequences satisfying (9) are called *return-to-zero* (RTZ) input. 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 $w_H(h(x)) = 2, 3$.

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, 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 the root of $h(x)$ and we distinguish 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. 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 ${}_kC_j$.

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

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

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

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

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

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

B. The weight-3 PCs

Without loss of generality, the 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 (α, β) satisfying (21) 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 let $\mathbb{M} = \{\epsilon_0\ell + m\}_{\ell\in\mathbb{Z}}$ and $\mathbb{N} = \{\epsilon_0\ell + n\}_{\ell\in\mathbb{Z}}$. Then it is obvious that each pair $(\alpha,\ \beta) \in \mathbb{M} \otimes \mathbb{N}$ 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 $n \equiv \beta \mod \epsilon_0$.

C. Examples

In the following, we present some examples of the proposed method to determine weight-2 and weight-3 PCs for several feedfoward polynomials of form given in (12). For the case $K = 1$, Examples 1 and 2 are two instances where $f(x)$ is a primitive polynomial while an instance where $f(x)$ is prime but not a 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 $K = 2$. Some weight-2 and weight-3 $h(x)$ are compiled in Table II with their corresponding $a(x)$ for reference.

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 a root 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 I that there is a pair $(1, 2)$ satisfying $x^1 + x^2 \equiv 1 \mod f(x)$. Thus, let $\mathbb{M} = \{3\ell + 1\}_{\ell\in\mathbb{Z}}$ and $\mathbb{N} = \{3\ell + 2\}_{\ell\in\mathbb{Z}}$. Then, we have $x^{\alpha} + x^{\beta} \equiv 1 \mod f(x)$ for each $(\alpha, \beta) \in \mathbb{M} \otimes \mathbb{N}$.

TABLE I: 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	$1 + x$

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 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 III and observe that there are 7 (m, n) pairs which satisfy $x^m + x^n \equiv 1 \mod 15$. Thus, $(\alpha,\ \beta) \in \bigcup_{i=0}^{6} \mathbb{M}_i \otimes \mathbb{N}_i$ satisfies (20), where

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

2) $f(x)$ 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 III, $\epsilon_0 = 5 < 15$ 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 III 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.

3) $K = 1$ 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 $\epsilon_0 = 1$. Thus, each $\alpha \in \mathbb{Z}^+$ should satisfy

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

However, the following second order differential equation

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

implies α should be an even number. Therefore, 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 $\alpha = 4\ell$, $\ell \in \mathbb{Z}^+$, satisfies (25) simultaneously.

Since $\mathrm{GF}(2)$ 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 $f(x)$.

4) The case $K = 2$: For this case, we write the feedforward polynomial as $f(x) = f_0(x)f_1(x)$ and give two examples.

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 $f_1(x)$. Since $f_0(x)$ does not yields any weight-3 PCs as explained in the Example 4, there are no such PCs associated with $f(x)$.

With respect to the weight-2 PCs, from Example 4, $\epsilon_0 = 1$ 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

$$a(x) = \sum_{i=0}^{\ell-1} x^{7i}(1 + x^2 + x^3)$$

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 $h(x) = 1 + x^{21\ell}$, $\ell \in \mathbb{Z}^+$, 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 $\mathbb{M}$ and $\mathbb{N}$ in Example 1 as $\mathbb{M}^0$ and $\mathbb{N}^0$, respectively, and referring to Table III, let

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

Then, we have

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

and

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

Therefore, by taking the intersection, we can identify $(\alpha, \beta) \in (\mathbb{M}^0 \otimes \mathbb{N}^0) \bigcap \left(\bigcup_{i=0}^{2} \mathbb{M}_i^1 \otimes \mathbb{N}_i^1\right)$.

V. VALIDITY CONFIRMATION THROUGH UNION BOUND

In this section, in order to confirm the validity of our proposed method, we obtain a union bound using the low-weight codewords with weight-2 and weight-3 PC or SC and compare it with that obtained via the transfer function as well as simulation results.

A. A novel union bound

Let $\mathbb{A}_h(d)$ be the set of all $a(x)$ which yields weight-d PCs *i.e.*, $w_H(h(x)) = w_H(a(x)f(x)) = d$. Similarly, we also define $\mathbb{A}_b(d)$ and $\mathbb{A}_c(d)$ as the sets of all $a(x)$s which result in weight-d SCs and codewords, respectively.

TABLE II: $a(x)$ and $h(x)$ for various $f(x)$

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

TABLE III: 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}			$1+x^3$

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

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

Now, we replace the set $\mathbb{A}_c(d)$ by the following approximated set

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

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

To obtain $\mathbb{A}'_c(d)$, based on $f(x)$, we first generate the set consisting of $a(x)$s which yield the weight-2 and -3 PCs, *i.e.* $\mathbb{A}_h(2) \cup \mathbb{A}_h(3)$. Next, for each $a(x) \in \mathbb{A}_h(2) \cup \mathbb{A}_h(3)$, 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 $\mathbb{A}_b(2) \cup \mathbb{A}_b(3)$ obtained based on $g(x)$. Finally, we narrow down the corresponding codewords as $w_H(b(x)) + w_H(h(x)) \leq d_{\text{free+1}}$ for $a(x) \in \mathbb{A}_h(2) \cup \mathbb{A}_h(3) \cup \mathbb{A}_b(2) \cup \mathbb{A}_b(3)$.

As examples, in Table V,VI and VII, we listed the low-weight PCs and SCs found by our proposed method for the codes listed in Table IV with the corresponding example numbers where each polynomial appeared in.

TABLE IV: The generator polynomials

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

B. Numerical results

We obtained the approximated union bound by (29) for the codes listed in Table IV. Since the codewords with the weights larger than $d_{\text{free+1}}$ 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 obtained using transfer function in Figures 1-3. For reference purpose, the details of the PCs and SCs used for drawing the bound are listed in Tables V - VII with the extra codewords found by computer search (labelled as 'Not Found'). In these figures, we also evaluated 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 1,000 bits errors for obtain each plot point.

As shown in Table V, 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

TABLE V: SCs and PCs for Code I

$w_H(c(x))$		$a(x)$	$b(x)$	$h(x)$
5	Found	1	$1+x+x^2$	$1+x^2$
6		$1+x$	$1+x^3$	$1+x+x^2+x^3$
		$1+x^2$	$1+x+x^3+x^4$	$1+x^4$
	Found	$1+x+x^2$	$1+x^2+x^4$	$1+x+x^3+x^4$
		$1+x+x^3$	$1+x^4+x^5$	$1+x+x^2+x^5$
7	Found	$1+x^2+x^3$	$1+x+x^5$	$1+x^3+x^4+x^5$
		$1+x^2+x^4$	$1+x+x^3+x^5+x^6$	$1+x^6$
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 VI: SCs and PCs for Code II

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

hand, some of them 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 $d_{\text{free}+1}$ tracks the BER curve with sufficient accuracy, especially in the high E_b/N_0 region. Moreover, the bounds obtained by our method and the transfer function converge to the same value with E_b/N_0 increament and match the simulation results well.

TABLE VII: SCs and PCs for Code III

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

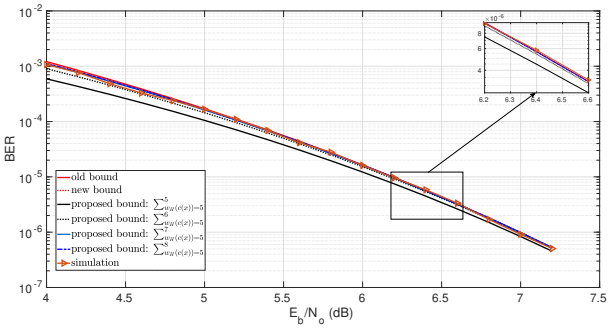

Fig. 1: Old Bound vs New Bound vs Simulation for 5/7 RSC Code

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 with weight higher than 7. Although Table VI indicates that there are 3 weight-8 codewords, and only one of them is found by our method, we can see from Fig. 2 that their contributions to the union bound are negligible and the BER curve can be well approximated using the codewords with weight 6 and 7 with a high accuracy at the high E_b/N_0 region.

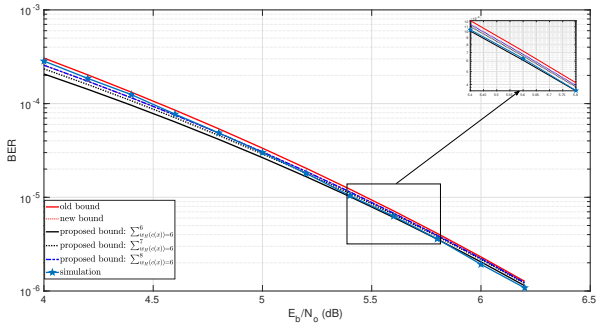

Fig. 2: Old Bound vs New Bound vs Simulation for 37/21 RSC Code

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

VI. Conclusion

In this paper, we proposed a method to determine the patterns of low-weight codewords of an RSC code. We established the low-weight codewords by identifying codewords with SCs or

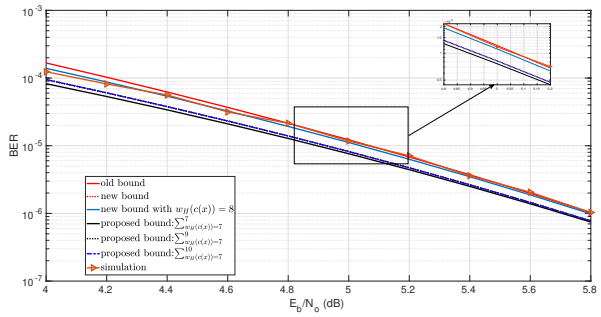

Fig. 3: Old Bound vs New Bound vs Simulation for 23/35 RSC Code

PCs of weight-2 and weight-3. We validated our proposed method by obtaining a union bound using the established low-weight codewords and compared it with that obtained via the transfer function method and the BER curve drawn from simulation results.

REFERENCES

[1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near shannon limit error-correcting coding and decoding: Turbo-codes. 1," in *Proc. IEEE Int. Conf. Commun. (ICC'93)*, vol. 2, Geneva, Switzerland, May 1993, pp. 1064–1070 vol.2.

[2] P. Thitimajshima, "Recursive systematic convolutional codes and application to parallel concatenation," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM'95)*, vol. 3, Singapore, Nov. 1995, pp. 2267–2272 vol.3.

[3] S. Benedetto and G. Montorsi, "Design of parallel concatenated convolutional codes," *IEEE Trans. Commun.*, vol. 44, no. 5, pp. 591–600, 1996.

[4] *IEEE Standard for Local and Metropolitan Area Networks. Part 16: Air Interface for Fixed Broadband Wireless Access Systems*, IEEE std 802.16-2004 Std., Rev. of IEEE 802.16-2001, Aug. 2004.

[5] *Digital Video Broadcasting(DVB); Second Generation DVB Interactive Satellite System (DVB-RCS2); Part2: Lower Layers for Satellite Standard*, ETSI EN 301 545-2 V1.2.1 (2014-04) Std., Rev. of ETSI EN 301 545-2 V1.1.1 (2012-01), Apr. 2014.

[6] *LTE Envolved Universal Terrestrial Radio Access (E-UTRA): Multiplexing and Channel Coding*, Third Generation Partnership Project (3GPP) Std., Jan. 2011.

[7] Y. Li, X. Guo, and X. Wang, "Design of recursive convolutional space-time codes with an arbitrary number of transmit antennas," *IEEE Commun. Lett.*, vol. 9, no. 7, pp. 637–639, 2005.

[8] S. Gounai, T. Ohtsuki, and T. Kaneko, "Performance of concatenated code with LDPC Code and RSC Code," in *Proc. IEEE Int. Conf. Commun. (ICC'06)*, vol. 3, 2006, pp. 1195–1199.

[9] M. F. U. Butt, R. A. Riaz, S. X. Ng, and L. Hanzo, "Near-capacity iteratively decoded binary self-concatenated code design using exit charts," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM'08)*, New Orleans, LA, USA, Nov.30-Dec.4 2008, pp. 1–5.

[10] C. Xu, "Soft decoding algorithm for rs-cc concatenated codes in wimax system," in *Proc. IEEE Veh. Tech. Conf. (VTC'07)*, Dublin, Ireland, Apr. 2007, pp. 740–742.

[11] K. Byun, S. Jung, D. J. Shin, and J. S. Um, "Performance comparison of RS-CC concatenated codes using NSC and RSC codes," in *Proc. IEEE Int. Conf. on Netw. Infrastructure and Digit. Content (ICNIDC'10)*, Beijing, China, Sep. 2010, pp. 992–994.

[12] A. Joshi and D. S. Saini, "Performance analysis of coded-OFDM with RS-CC and turbo codes in various fading environment," in *Proc. Int. Conf. on Inf. Tech. and Multimedia(ICIMU'11)*, Kuala Lumpur, Malaysia, Nov. 2011, pp. 1–6.

[13] T. K. Moon, *Error Correcting Codes*. Hoboken, NJ, USA: Wiley, 2005.

[14] S. Lu, W. Hou, and J. Cheng, "Input-output weight distribution of terminated RSC codes with limited codelength," in *Proc. IEEE Int. Symp. Infor. Theory and its Appl. (ISITA'16)*, Monterey, CA, USA, Oct.30-Nov.2 2016, pp. 493–497.

[15] J. Deng, Y. Peng, and H. Zhao, "Distance spectrum calculation method for double binary turbo codes," in *Proc. Int. Conf. on Recent Advances in Sig. Process., Telecommun. Comput.(SigTelCom'17)*, Da Nang, Vietnam, Jan. 2017, pp. 98–102.

[16] J. Sun and O. Y. Takeshita, "Interleavers for turbo codes using permutation polynomials over integer rings," *IEEE Trans. Inf. Theory*, vol. 51, no. 1, pp. 101–119, Jan. 2005.