

A Novel Method for Obtaining the Pattern of 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

In this paper, we present a novel low-complexity method for determining the structure of the components of a low weight codeword for a given RSC code, which is useful in interleaver design for turbo codes. For selected RSC codes, we present a list of the low-weight codeword components up to a cut-off weight d_{\max} and compare the simulation results to the bounds obtained via our novel method and the transfer function method to validate our method.

I. INTRODUCTION

The *turbo code* (TC) [?], introduced by Claude Berrou in 1993 is one of the forward-error correcting 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 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 [?].

The simplest and most common construction of a TC is to concatenate two *recursive systematic convolutional* (RSC) codes (usually of the same kind) parallelly via an interleaver. One of the many reasons why the TC excels as a channel code is its ability to map low-weight parity-check sequences in the first component RSC code to high-weight parity-check sequences in the second component RSC code using an 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 [?] have to be used. 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 [?] 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 very dominant.

In this paper, we present a novel method that can be used to find the distance spectrum of an RSC code as well as the pattern of the low-weight codeword components. The complexity of our method is independent of the number of states of the RSC code and its ability to reveal low-weight codeword patterns of an RSC code makes it an excellent tool for use in interleaver

design.

In order to validate our method, we generate a partial distance spectrum for specific RSC codes and compare it to the lower bound obtained via the transfer function method. We also compare the bounds obtained using our novel method to simulation results. In both cases, it is observed that the values begin to converge as E_b/N_0 increases.

The remainder of the research paper is organised as follows. Notations and definitions used in the research paper are introduced in Section II. In Section III, we discuss the distance spectrum and union bound of RSC codes and present the theory behind our novel method for obtaining the distance spectrum. Moving on to Section IV, we use our novel method to determine low-weight parity check patterns. Comparison of bounds obtained using our novel method to that obtained using the transfer function as well as simulation results are presented in Section V and the paper concludes in Section ??.

A. Notations

For two positive integers α and β , the least common multiple of α and β is denoted as $\text{lcm}(\alpha, \beta)$ while the remainder α divided by β is denoted as $\alpha \bmod \beta$. For an integer pair (α, β) , $(\alpha, \beta) \bmod$ 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

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

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 the *monic* polynomial. We call the total number of non-zero coefficients the *Hamming weight* of $v(x)$, which is written 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 the Galois field, denoted as $\text{GF}(p)$. If the coefficients in (1) are elements of $\text{GF}(p)$, $v(x)$ is called *polynomial over* $\text{GF}(p)$.

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

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

and

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

respectively. A monic polynomial which cannot be represented by multiplication of some lower degree polynomials is called a *prime polynomial*. For two polynomials $v(x)$ and $w(x)$ over $\text{GF}(p)$, we assume $w(x) \neq 0$. Then there exists polynomials $q(x)$ and $r(x)$ over $\text{GF}(p)$ such that

$$v(x) = w(x)q(x) + r(x) \quad (4)$$

with $\deg(r(x)) < \deg(w(x))$. $r(x)$ in the expression (4), is denoted by

$$r(x) \equiv v(x) \mod w(x) \quad (5)$$

is called 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 $\text{GF}(p)$ with $\deg(v(x)) := M > 1$ and \mathcal{V} be the set of size p^M consisting of all polynomials over $\text{GF}(p)$ with degree less than M . Then, the *extension field of* $\text{GF}(p)$, denoted by $\text{GF}(p^M)$, is the set \mathcal{V} with addition and multiplication over $\text{GF}(p)$

where the multiplication is carried out modulo- $v(x)$ over $\text{GF}(p)$. Each non-zero elements in $\text{GF}(p^M)$ 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 $\text{GF}(p^M)$, there exist integers ϵ such that $x^\epsilon = 1$ and the least positive integer among them is called the *order* of x . The element with order $\epsilon = p^M - 1$ is called *primitive element*. For $\text{GF}(p^M)$ generated by a prime polynomial $v(x)$ with $\deg(v(x)) = M$, if x is a primitive element in $\text{GF}(p^M)$, then $v(x)$ is called *primitive polynomial*. Finally, the root of $v(x)$, is the non-zero element in $\varphi \in \text{GF}(p^M)$ such that $v(\varphi) = 0$. If $v(x)$ is a primitive polynomial, the order of φ is $\epsilon = p^M - 1$ while $\epsilon \nmid p^M - 1$ otherwise. Moreover, the elements φ^i , $0 \leq i \leq \epsilon - 1$, are all distinct 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 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

$$\begin{bmatrix} 1 & \frac{f(x)}{g(x)} \end{bmatrix}$$

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. Each output $c(x)$ is a mixture of the SC and PC as

$$c(x) = b(x^2) + xh(x^2) \quad (6)$$

where

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

From (6), it is trivial that

$$w_H(c(x)) = w_H(b(x)) + w_H(h(x)) \quad (8)$$

and hence, each low-weight codeword is a conjunction of two low-weight SC and PC.

Under the assumption of large frame sizes, the presence of $g^{-1}(x)$ in (7) may involve a particular sequence of bits that is repeated a large number of times, hence a high-weight PC. A low-weight PC occurs if and only if

$$b(x) \bmod g(x) \equiv 0 \quad (9)$$

The inputs $b(x)$ which meet the condition in (9) is called a *return-to-zero* (RTZ) input. Thus, every RTZ input can be factorized by

$$b(x) = a(x)g(x) \quad (10)$$

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

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

Finally, for a given RSC code, we can formulate our goal as, to find all $a(x)$ s which satisfy (10) and (11) simultaneously. 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

To determine the details of the patterns of the low-weight PCs, we assume $f(x)$ can be factorized into K prime polynomials as

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

where $\gamma_0, \gamma_1, \dots, \gamma_{K-1}$ are positive integers and we assume φ_k is a root of $f_k(x)$ of order ϵ_k .

Referring (11), we consider the solution of

$$h(x) \bmod f(x) \equiv 0 \quad (13)$$

We start from the simplest case $K = 1$, i.e., $f(x) = f_0^{\gamma_0}(x)$. Then, we can see from (11) that each root is also a root of $h(x)$. For the case $\gamma_0 = 1$, since all φ_0^i , $0 \leq i < \epsilon_0$, are distinct from each other, the equation

$$h(\varphi_0^i) = 0, \quad 0 \leq i < \epsilon_0 \quad (14)$$

is a necessary and sufficient condition of (13).

For $\gamma_0 > 1$, on the other hand, (14) is necessary but not sufficient for (13). For this case, although we may derive some solutions by differential equations

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

we can not determine the patterns completely, since the operations on coefficients of the polynomial are performed on the terms $\bmod p$. Thus, we need to remove possible ghost solutions at further confirmation step.

For the case $K > 1$, 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 PC patterns.

A. The patterns of the weight-2 PCs

Each weight-2 PC can be written as

$$h(x) = 1 + x^\alpha \quad (16)$$

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

$$(\varphi_0^i)^\alpha = 1, \quad 0 \leq i < \epsilon_0 \quad (17)$$

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

$$\alpha \bmod \epsilon_0 \equiv 0 \quad (18)$$

B. The patterns of the weight-3 PCs

Each weight-3 PC can be written as

$$h(x) = 1 + x^\alpha + x^\beta, \quad \alpha < \beta \quad (19)$$

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

$$\varphi_0^\alpha + \varphi_0^\beta = 1 \quad (20)$$

Such pairs can be found by referring to the table of the extended field for $\text{GF}(2^M)$. 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 $n \equiv \beta \pmod{\epsilon_0}$.

C. Examples

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

$f(x)$ is a primitive polynomial and since $x^1 = x$, $x^2 \equiv 1 + x$, and $x^3 \equiv 1 \pmod{f(x)}$, the order of the root φ_0 is $\epsilon_0 = 3$.

Weight-2 PCs: From (18), it is obvious that α should be a multiple of 3. The corresponding values for $a(x)$ and $h(x)$ are shown in Table I for the first four valid values of α . We may write

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

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

the weight-2 PCs in general form as $h(x) = 1 + x^{3\ell}$, $\ell > 1$, and the corresponding $a(x)$ is given by

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

Weight-3 PCs: The elements of $\text{GF}(2^2)$ are shown in Table II. We can see from this table that $(m, n) = (1, 2)$ and, consequently, let $\mathcal{M} = \{3\ell + 1\}_{\ell \geq 0}$ and $\mathcal{N} = \{3\ell + 2\}_{\ell \geq 0}$. Then, we have $(\alpha, \beta) \in \mathcal{M} \otimes \mathcal{N}$.

The corresponding values for $a(x)$ and $h(x)$ are shown in Table III below for the first four valid values of (α, β) and the weight-3 PCs can be written as $h(x) = 1 + x^{3\ell_0+1} + x^{3\ell_1+2}$, $\ell_0, \ell_1 \geq 0$ in general form.

TABLE II: Non-zero Elements of GF (2^2) generated by $f(x) = 1 + x + x^2$

power representation	actual value
$x^0 = x^3 = 1$	1
x	x
x^2	$1 + x$

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

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

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

For this polynomial, we can confirm the order of φ_0 is $\epsilon_0 = 5 < 15$. Thus, $f(x)$ is a prime but not primitive polynomial.

Weight-2 PCs: From (18), α should be a multiple of 5. The corresponding values for $a(x)$ and $h(x)$ are shown in Table IV with general forms for $\ell > 1$

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

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

Weight-3 PCs: We refer to Table V and confirm that there is no pair (m, n) that satisfies $x^m + x^n \equiv 1$. Thus, there are no weight-3 PCs for $f(x)$

TABLE V: Non-zero Elements of $\text{GF}(2^4)$ 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
$x^4 = x^9 = x^{14}$	$1 + x + x^2 + x^3$

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

We can write $f(x)$ as

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

and the order of the root $\varphi_0 = 1$ is $\epsilon_0 = 1$.

Weight-2 PCs: We obtain from (14) and (15)

$$(\varphi_0)^\alpha = 1 \tag{21}$$

and

$$\alpha(\varphi_0)^{(\alpha-1)} = 0 \tag{22}$$

respectively. Although (21) implies α in (18) can be any positive integer, we can see from (22) that α should be an even number. The corresponding values for $a(x)$ and $h(x)$ are shown in Table VI with general forms for $\ell > 1$.

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

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

Weight-3 PCs: With the same reason of Example 2, there are no weight-3 PCs associated with $f(x)$.

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

The given polynomial can be written as

$$f(x) = \prod_{k=0}^1 f_k(x)$$

where

$$f_0(x) = 1 + x + x^2, \quad f_1(x) = 1 + x + x^2 + x^3 + x^4$$

and we know from Example 1 and Example 2 that $\epsilon_0 = 3$ and $\epsilon_1 = 5$.

Weight-2 PCs: From (18), the valid values of α should be a multiple of ϵ_0 and ϵ_1 , hence, α should be a multiple of $\text{lcm}(\epsilon_0, \epsilon_1) = 15$. The corresponding values for $a(x)$ and $h(x)$ are shown in Table VIII with general forms for $\ell > 1$.

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

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

Weight-3 PCs: Since we have shown in Example 2 that $f_1(x)$ does not yield any weight-3 PC, there are no weight-3 PCs associated with $f(x)$.

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

The polynomial $f(x)$ can be written as

$$f(x) = \prod_{k=0}^1 f_k(x)$$

where

$$f_0(x) = 1 + x + x^2, \quad f_1(x) = 1 + x^2 + x^3$$

We know from Example 1 that $\epsilon_0 = 3$ and it can be confirmed that $\epsilon_1 = 7$.

Weight-2 PCs: The valid values of α in (18) should be a multiple of $\text{lcm}(\epsilon_0, \epsilon_1) = 21$. The corresponding values for $a(x)$ and $h(x)$ are shown in Table VIII with general forms for $\ell > 1$.

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

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

Weight-3 PCs: We rewrite \mathcal{M} and \mathcal{N} in Example 1 as \mathcal{M}^0 and \mathcal{N}^0 , respectively, and referring Table IX let

$$\begin{aligned}
\mathcal{M}_0^1 &:= \{7\ell + 1\}_{\ell \geq 0}, \quad \mathcal{N}_0^1 := \{7\ell + 5\}_{\ell \geq 0} \\
\mathcal{M}_1^1 &:= \{7\ell + 2\}_{\ell \geq 0}, \quad \mathcal{N}_1^1 := \{7\ell + 3\}_{\ell \geq 0} \\
\mathcal{M}_2^1 &:= \{7\ell + 4\}_{\ell \geq 0}, \quad \mathcal{N}_2^1 := \{7\ell + 6\}_{\ell \geq 0}
\end{aligned} \tag{23}$$

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 intersection, we have $(\alpha, \beta) \in (\mathcal{M}^0 \otimes \mathcal{N}^0) \cap (\bigcup_{i=0}^2 \mathcal{M}_i^1 \otimes \mathcal{N}_i^1)$.

TABLE IX: Non-zero Elements of $\text{GF}(2^3)$ 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	$x + x^2$

The corresponding values for $a(x)$ and $h(x)$ are shown in Table X below for the first three valid values of (α, β) .

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

$a(x)$	$h(x)$
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. THE VALIDITY CONFIRMATION THROUGH THE UNION BOUND

In this section, In order to confirm the validity of our proposed method, we obtain a union bound using the list of low-weight codewords and compare it with that by transfer function and simulations.

To obtain a union bound, let $\mathcal{A}_h(d)$ be the set of all $a(x)$ which yields weight- d parity-check component *i.e.*, $w_H(h(x)) = w_H(a(x)f(x)) = d$ for $a(x) \in \mathcal{A}_h(d)$. Similarly, let $\mathcal{A}_b(d)$ be the set of all $a(x)$ which yields weight- d systematic component *i.e.*, $w_H(b(x)) = w_H(a(x)g(x)) = d$, for $a(x) \in \mathcal{A}_b(d)$ while the set of all $a(x)$ which yields weight- d codeword *i.e.*, $w_H(c(x)) = w_H(a(x)f(x)) + w_H(a(x)g(x)) = d$, is denoted by $\mathcal{A}_c(d)$.

From (8), when $w_H(b(x)), w_H(h(x)) \geq 2$, we have

$$\mathcal{A}_c(d) = \bigcup_{\ell=2}^{d-2} \{\mathcal{A}_b(\ell) \cap \mathcal{A}_h(d-\ell)\} \quad (24)$$

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 approximated set $\mathcal{A}'_c(d)$, written as

$$\mathcal{A}_c(d) \approx \mathcal{A}'_c(d) = \left\{ \bigcup_{\ell=2}^{\ell+1} \{\mathcal{A}_b(\ell) \cap \mathcal{A}_h(d-\ell)\} \right\} \cup \left\{ \bigcup_{\ell=2}^{\ell+1} \{\mathcal{A}_b(d-\ell) \cap \mathcal{A}_h(\ell)\} \right\} \quad (25)$$

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 as

$$P_b \leq \frac{1}{k} \sum_{d=d_{\text{free}}}^{d_{\text{max}}} \sum_{a(x) \in \mathcal{A}'_c(d)} w_H(a(x)g(x)) Q\left(\sqrt{\frac{2dE_c}{N_0}}\right) \quad (26)$$

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

To verify the validity, assuming frame size $N = 64$, we compared the bounds in (26) with that obtained using the transfer function method as well as the simulation results for the 5/7, 37/21 and 23/35 RSC codes.

Example 6. The SC and PC components of 5/7 RSC are listed in Tables XI. For this code, since the feedforward connection has the polynomial representation $1 + x^2$, the low-weight PCs are derived from Example 3. On the other hand, for the feedback connection polynomial $1 + x + x^2$, the details of low-weight patterns are demonstrated in Example 1.

TABLE XI: SC and PC Components 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	$1 + x^2$	$1 + x + x^3 + x^4$	$1 + x^4$
	$1 + x$	$1 + x^3$	$1 + x + x^2 + x^3$
7	$1 + x^2 + x^4$	$1 + x + x^3 + x^5 + x^6$	$1 + x^6$
	$1 + x^2 + x^3$	$1 + x + x^5$	$1 + x^3 + x^4 + x^5$
	$1 + x + x^2$	$1 + x^2 + x^4$	$1 + x + x^3 + x^4$
	$1 + x + x^3$	$1 + x^4 + x^5$	$1 + x + x^2 + x^5$
8	$1 + x^2 + x^4 + x^6$	$1 + x + x^3 + x^5 + x^7 + x^8$	$1 + x^8$
	$1 + x + x^3 + x^4$	$1 + x^6$	$1 + x + x^2 + x^4 + x^5 + x^6$

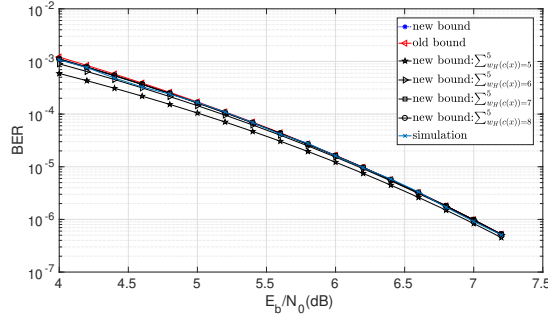


Fig. 1: Old Bound vs New Bound vs Simulation for 5/7 RSC Code

Fig. 1 shows the simulation results for the 5/7 RSC code as well as the lower bounds obtained using the transfer function as well as our novel method. We can observe that the accuracy of new (novel method) bound increases with the number of terms used in the approximation. Also, we can observe from this figure that there is some difference between the new (novel method) bound and the old (transfer function) bound, but they tend to converge as E_b/N_0 increases. This suggests that the approximation (25) used in our novel method is sufficient for the 5/7 RSC code.

Example 7. For 37/21 RSC code, since the polynomials of feedforward and feedback connections are $1 + x + x^2 + x^3 + x^4$ and $1 + x^4$, respectively. Thus, the CS and PC components listed in Table XII can be obtained by the method demonstrated in Examples 2 and 3, respectively.

TABLE XII: SC and PC Components for the 37/21 RSC code, $d_{\text{free}} = 6$

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

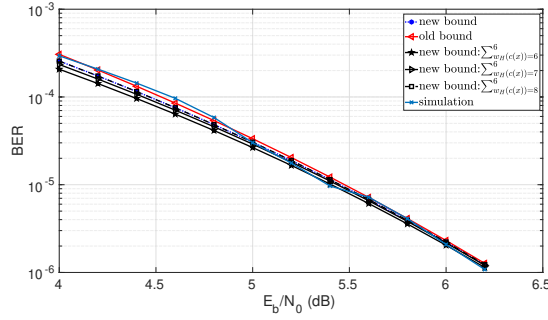


Fig. 2: Old Bound vs New Bound vs Simulation for 37/21 RSC Code

Fig. 2 shows the simulation results for the 37/21 RSC code as well as the lower bounds obtained using the transfer function as well as our novel method. The bounds as well as the simulation results for this RSC code are shown in Fig. 2 and by observation, we can draw conclusions similar to Example 6 with respect to the accuracy of the novel method .

Example 8. Fig. 3 shows the simulation results for the 23/35 RSC code as well as the lower bounds obtained using the transfer function as well as our novel method. The feedforward connection has the polynomial representation $1 + x + x^4$, which is similar in characteristic to the polynomial in Example 1. It can easily be confirmed that there exists weight-2 and weight-3 PCs. For the weight-2 PCs, the general form for $a(x)$ is

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

and since it yields codewords such that $w_H(c(x)) > d_{\text{max}}$, there are not included in our approximation of the lower bound, as can be observed from Table XII. The feedback connection

TABLE XIII: Partial Codeword Component Pattern Distance Spectrum for the 23/35 RSC code, $d_{\text{free}} = 7$

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

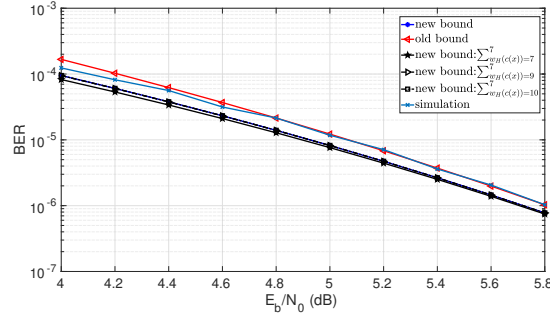


Fig. 3: Old Bound vs New Bound vs Simulation for 23/35 RSC Code

has the polynomial representation $1 + x^2 + x^3 + x^4$, which can be factorized into 2 irreducible polynomials and it can easily be confirmed that there exists no weight-3 SCs, since $1 + x$ is a factor. In Fig. 3, we observe that the old (transfer function) bounds and simulation results converge as the E_b/N_0 value increases. However, there is some difference between the new (novel method) bound and the old (transfer function) bound, even as E_b/N_0 increases. This suggests that the approximation used in our novel method is insufficient for this 23/35 RSC code and considering $w_H(h(x))$, $w_H(b(x)) = 4$ might yield a more accurate bound.