“ Progress So Far”

Kwame Ackah Bohulu

December 16, 2020

0.1 Preliminaries

This section outlines the definitions and notations that will be used. An example is also given at the end to better clarify the use of the notations.

0.1.1 Definitions and Notations

1. RTZ (Return-To-Zero) input :- A RTZ input is a binary input which causes a RSC encoder's final state to be return to zero after it has exited the zero state.
2. τ :- cycle length of the RSC encoder. For the 5/7 RSC encoder $\tau = 3$
3. N :- Interleaver length.
4. L :- Outer coset length where $L = N/\tau$
5. M :- Inner coset length where $M = L/\tau$
6. $\mathcal{N}$:- Integer set of $\{0, 1, \cdots, N-1\}$
7. $\mathbb{N}$: Indexed set of $\{0, 1, \cdots, N-1\}$ in the natural order.
8. $\mathcal{C}$:- Integer set of $\{0, 1, \cdots, L-1\}$
9. $\mathbb{C}$: Indexed set of $\{0, 1, \cdots, L-1\}$ in the natural order.
10. $\mathcal{M}$:- Integer set of $\{0, 1, \cdots, M-1\}$
11. $\mathbb{M}$: Indexed set of $\{0, 1, \cdots, M-1\}$ in the natural order.
12. $\mathcal{C}^t := \{c+t\}_{c\in\mathcal{C}}$ and $\mathbb{C}^t$ is the indexed set with the elements of $\mathcal{C}^t$ where $t = (0, 1, ..., \tau-1)$.
13. $\mathcal{C}^{tt'} := \{m+t\}_{m\in\mathcal{M}}$ and $\mathbb{C}^{tt'}$ is the indexed set with the elements of $\mathcal{C}^{tt'}$ where $tt' = (0, 1, ..., \tau-1)$.
14. Permutation matrix

$$\mathbf{\Pi} = \begin{bmatrix} \boldsymbol{\pi}^0 \\ \boldsymbol{\pi}^1 \\ \vdots \\ \boldsymbol{\pi}^{K-1} \end{bmatrix} = \left[\boldsymbol{\pi}_0, \boldsymbol{\pi}_1, \cdots, \boldsymbol{\pi}_{\tau-1}\right] = \left[\pi_t^{(k)}\right]_{k=0,\ t=0}^{K-1,\ \tau-1}$$

where $\pi_t^{(k)} \in \{0, 1, \tau-1\}$.
15. For the row vector $\boldsymbol{\pi}^{(k)}$, let $\mathscr{S}^e[\boldsymbol{\pi}^{(k)}]$ be the left-hand cycle shift of $\boldsymbol{\pi}^{(k)}$ and $\mathscr{S}^e[\boldsymbol{\pi}_t]$ be the up cycle shift of $\boldsymbol{\pi}_t$
16. We assume that the permutation matrix operation outputs the elements in $\mathbb{C}^t$ in the order which t appers in $\boldsymbol{\pi}^k$.
17. Our goal is to find the best $\mathbf{\Pi}$ and $\mathbb{C}^t$, $t = 0, 1, \cdots, \tau-1$.

0.1.2 Example

Lets assume we have a turbo code using the 5/7 RSC encoder as its component code ($\tau = 3$) and an interleaver length of $N = 27$. We have the following values

1. $C = 9$ and $M = 3$. Also $\mathcal{N} = \{0, 1, \cdots, 26\}$

2. $\mathcal{C}^0 = \{0, \tau, \cdots, (L-1)\tau\} = \{0, 3, \cdots, 24\}$, $\mathcal{C}^1 = \mathcal{C}^0 + 1$ and $\mathcal{C}^2 = \mathcal{C}^0 + 2$

3. $\mathcal{C}^{00} = \{0, \tau, \cdots, (M-1)\tau\} = \{0, 3, 6\}$, $\mathcal{C}^{01} = \mathcal{C}^{00} + 1$ and $\mathcal{C}^{02} = \mathcal{C}^{00} + 2$

4. Let $\mathbb{C}^0 = \{0, 3\tau, 6\tau, 1\tau, 4\tau, 7\tau, 2\tau, 5\tau, 8\tau\} = \{0, 9, 18, 3, 12, 21, 6, 15, 24\}$, $\mathbb{C}^1 = \{4, 13, 22, 7, 16, 25, 1, 10, 19\}$, $\{23, 8, 17, 26, 2, 11, 20, 5, 14\}$ and $\mathbf{\Pi} = \begin{bmatrix} 0 & 0 & 0 \\ 2 & 2 & 2 \\ 1 & 1 & 1 \end{bmatrix}$.

   then

$$\begin{aligned} \mathbb{N} =& \{c_0^0, c_1^0, c_2^0, c_0^2, c_1^2, c_2^2, c_0^1, c_1^1, c_2^1, \cdots, c_6^1, c_7^1, c_8^1\} \\ =& \{0, 9, 18, 23, 8, 17, 4, 13, 22, 3, 12, 21, 26, 2, 11, 7, 16, 25, 6, 15, 24, 20, 5, 14, 1, 10, 19\} \end{aligned} \tag{0-1}$$

$\mathbb{N}$ represents the interleaved sequence. From this example, we can see that the column index of i in $\pi^{(k)}$ represents the coset it belongs to before interleaving and the value $\pi_t^{(k)}$ specifies the coset after interleaving. Also notice that the rows of $\mathbf{\Pi}$ are taken cyclicly untill all elements of $\mathbb{C}^t$ are placed in $\mathbb{N}$.

0.2 RTZ Inputs

in this section we talk a little bit more about the types of RTZ inputs and introduce their polynomial and coset definitions. Finally we talk about how certain RTZ inputs may be dealt with after interleaving.

0.2.1 Types of RTZ inputs

Regardless of the component code used in turbo coding, the RTZ inputs can be grouped into two basic forms. These are *base RTZ inputs* and *compound RTZ inputs.* Base RTZ inputs are dependent on the component code and cannot be broken down into 2 or more RTZ inputs. Compound RTZ inputs as the name implies are formed from 2 or more base RTZ inputs and therefore can be broken down into base RTZ input form.

For the 5/7 component code, its base RTZ inputs are weight-2 RTZ inputs (W2RTZs) and weight-3 RTZ inputs (W3RTZs). Every RTZ input with a weight higher than 3 is a compound RTZ input. In general for RTZs with weight w greater than 3, if $w \bmod 2 = 0$, then the RTZ is made up of $w/2$ W2RTZs. On the other hand, if $w \bmod 2 = 1$, then the RTZ is made up of $\lfloor w/2 \rfloor - 1$ W2RTZs and 1 W3RTZ.

The impulse response of the RSC encoder is the output of the encoder when the input is $\boldsymbol{\rho} = (10000...)$. The impulse response can be used to calculate the weight of any input sequence of weight w in general. This is done by noting that any input sequence of weight w is just a summation of w $\boldsymbol{\rho}$'s, where the consequetive $w-1$ $\boldsymbol{\rho}$'s have leading zeros. For the 5/7 RSC encoder, the impulse response is given by

$$(1110110110...)$$

The permutation matrix that generates the set $\mathcal{N}$ is given by

$$\Pi' = \begin{bmatrix} 0 & 1 & 2 \end{bmatrix}$$

where Π was used repeatedly untill all elements in $\mathcal{C}^t$ are picked. From Π we can derive the defintions for W2RTZs and W3RTZs as well as ways to break up such RTZs

0.2.2 W2RTZs : Definitions and Breaking them

Given below is the definition of W2RTZs in terms of polynomials and cosets.

- polynomial: $P(x) = x^{h\tau+t}(1 + x^{\alpha\tau}) = x^t(x^{h\tau} + x^{(h+\alpha)\tau})$
- coset: the hth and $(h+\alpha)$th elements in $\mathcal{C}^t$

From the impulse response,let $\boldsymbol{\phi}_2 := (011)$. The above definition for W2RTZs holds because for any W2RTZ defined like the above, you have $\boldsymbol{\phi}_2 + \boldsymbol{\phi}_2 = (000)$ after the second 1 bit which is repeated infinitely leading to a low-weight codeword.

From the coset definition, it is easy to see that W2RTZs can be broken if after interleaving, the hth and $(h+\alpha)$th elements in $\mathcal{C}^t$ are mapped to different cosets.

0.2.3 W3RTZs : Definitions and Breaking them

Given below is the definition of W3RTZs in terms of polynomials and cosets.

- polynomial: $Q(x) = x^{h\tau+t}(1 + x^{\beta\tau+1} + x^{\gamma\tau+2}) = x^{h\tau+t} + x^{(h+\beta)\tau+t+1} + x^{(h+\gamma)\tau+t+2}$. Notice that $h \leq \beta$ is not a necessary condition.
- coset: the hth element in $\mathcal{C}^t$, $(h+\beta)$th element in $\mathcal{C}^{[t+1]_\tau}$, and $(h+\gamma)$th element in $\mathcal{C}^{[t+2]_\tau}$.

From the impulse response,let $\boldsymbol{\phi}_2 := (011)$, $\boldsymbol{\phi}'_2 := (110), \boldsymbol{\phi}''_2 := (101)$. The above definition for W2RTZs holds because for any W2RTZ defined like the above, you have $\boldsymbol{\phi}_2 + \boldsymbol{\phi}'_2 + \boldsymbol{\phi}''_2 = (000)$ after the third 1 bit which is repeated infinitely leading to a low-weight codeword.

Again, from the coset definition, we see that the easiest way to break up W3RTZs is to make sure that after interleaving, a minimum of 2 elements are mapped into the same coset.

0.2.4 W4RTZs : Definitions and Breaking them

Given below is the definition of W4RTZs in terms of polynomials and cosets.

- polynomial: $P(x) = x^{h\tau+t}(1 + x^{\alpha_1\tau}) + x^{h'\tau+t'}(1 + x^{\alpha_2\tau}) = x^t(x^{h\tau} + x^{(h+\alpha_1)\tau}) + x^{t'}(x^{h'\tau} + x^{(h'+\alpha_2)\tau})$
- coset: the hth and $(h+\alpha_1)$th elements in $\mathcal{C}^t$ and the h'th and $(h'+\alpha_2)$th elements in $\mathcal{C}^{t'}$

where $t,\ t' \in \{0,1,2\}, h \neq h'$

The above defintion holds because a W4RTZ is a combination of 2 W2RTZs

0.2.5 W5RTZs : Definitions and Breaking them

Given below is the definition of W5RTZs in terms of polynomials and cosets.

- polynomial: $P(x) = x^{h\tau+t}(1 + x^{\alpha\tau}) + x^{h'\tau+t'}(1 + x^{\beta\tau+1} + x^{\gamma\tau+2}) = x^t(x^{h\tau} + x^{(h+\alpha)\tau}) + x^{h'\tau+t'} + x^{(h'+\beta)\tau+t'+1} + x^{(h'+\gamma)\tau+t'+2}$
- coset: the hth and $(h+\alpha)$th elements in $\mathcal{C}^t$ and the h'th element in $\mathcal{C}^{t'}$, $(h'+\beta)$th element in $\mathcal{C}^{[t'+1]_\tau}$, and $(h'+\gamma)$th element in $\mathcal{C}^{[t'+2]_\tau}$

where $t,\ t' \in \{0,1,2\}, h \neq h'$

The above defintion holds because a W5RTZ is a combination of a W2RTZ and a W3RTZ

There is not much that can be done to break up W4RTZs or W5RTZs using just Π. However careful selection of Π conbined with coset design can be used to effectively break up these higher weight RTZs. Permutation matrix design will be focused solely on W2RTZs and W3RTZs

0.2.6 Calculating Codeword Weights For Turbo Codes

Low weight turbo codewords will occur only if an RTZ input is fed into both component encoders. We will give (or derive) the equations that can be used to calculate the codeword weight for turbo codes generated as a result of the above defined RTZs being fed into both component codes.

Hamming Weight for W2RTZ Turbo Codewords

We modify the equation given in [Sun Takeshita] to fit our notation. The Hamming weight $w_H^{(2)}$ for a turbo codeword generated by a W2RTZ is calculated using the equation below

$$\begin{aligned} w_H^{(2)} =& 6 + 2\Big(\frac{\alpha\tau}{\tau} + \frac{\alpha'\tau}{\tau}\Big) \\ =& 6 + 2(\alpha + \alpha') \end{aligned} \tag{0-2}$$

where α' is the value of α after interleaving

Hamming Weight for W3RTZ Turbo Codewords

The equation for the Hamming weight $w_H^{(3)}$ for a turbo codeword generated by a W3RTZ is given by

$$7 + 2(l + l') \tag{0-3}$$

Proof. The polynomial representation of a weight-3 RTZ input is given by

$$Q(x) = x^{h\tau+t}(1 + x^{\beta\tau+1} + x^{\gamma\tau+2})$$

With reference to the impulse response of the 5/7 RSC encoder,

Let
$\boldsymbol{\phi}_1 = (0\ 0\ 1),\ \boldsymbol{\phi}'_1 = (0\ 1\ 0),\ \boldsymbol{\phi}''_1 = (1\ 0\ 0),$
$\boldsymbol{\phi}_2 = (0\ 1\ 1),\ \boldsymbol{\phi}'_2 = (1\ 1\ 0),\ \boldsymbol{\phi}''_2 = (1\ 0\ 1),$
$\boldsymbol{\phi}_3 = (1\ 1\ 1).$

Now, we consider the weight of the vector derived by the sumation of the followings vectors.

$$\begin{array}{lll} (\mathbf{0}_{3(\gamma+h)} & \boldsymbol{\phi}_1\ \boldsymbol{\phi}'_2 & \cdots) \\ (\mathbf{0}_{3(\beta+h)} & \boldsymbol{\phi}_2\ \boldsymbol{\phi}''_2 & \cdots) \\ (\mathbf{0}_{3h} & \boldsymbol{\phi}_3\ \boldsymbol{\phi}_2 & \cdots) \end{array}$$

Without loss of generality, we can assume that all weight-3 RTZ inputs begin at the 0th position, ie $h = t = 0$. This is because the case where $h > 0$ or $t > 0$ is just a right-shifted version of the weight-3 RTZ. With this assumption, we we only need to consider cases where $h = 0,\ \gamma \geq h$. To simplify calculation, we have included an addition table for all the vectors which is shown in Table **??**

	$\boldsymbol{\phi}_1$	$\boldsymbol{\phi}'_1$	$\boldsymbol{\phi}''_1$	$\boldsymbol{\phi}_2$	$\boldsymbol{\phi}'_2$	$\boldsymbol{\phi}''_2$	$\boldsymbol{\phi}_3$
$\boldsymbol{\phi}_1$	$\mathbf{0}_3$	–	–	–	–	–	–
$\boldsymbol{\phi}'_1$	$\boldsymbol{\phi}_2$	$\mathbf{0}_3$	–	–	–	–	–
$\boldsymbol{\phi}''_1$	$\boldsymbol{\phi}''_2$	$\boldsymbol{\phi}'_2$	$\mathbf{0}_3$	–	–	–	–
$\boldsymbol{\phi}_2$	$\boldsymbol{\phi}'_1$	$\boldsymbol{\phi}_1$	$\boldsymbol{\phi}_3$	$\mathbf{0}_3$	–	–	–
$\boldsymbol{\phi}'_2$	$\boldsymbol{\phi}_3$	$\boldsymbol{\phi}''_1$	$\boldsymbol{\phi}'_1$	$\boldsymbol{\phi}''_2$	$\mathbf{0}_3$	–	–
$\boldsymbol{\phi}''_2$	$\boldsymbol{\phi}''_1$	$\boldsymbol{\phi}_3$	$\boldsymbol{\phi}_1$	$\boldsymbol{\phi}'_2$	$\boldsymbol{\phi}_2$	$\mathbf{0}_3$	–
$\boldsymbol{\phi}_3$	$\boldsymbol{\phi}'_2$	$\boldsymbol{\phi}''_2$	$\boldsymbol{\phi}_2$	$\boldsymbol{\phi}''_1$	$\boldsymbol{\phi}_1$	$\boldsymbol{\phi}'_1$	$\mathbf{0}_3$

Table 1: Truth Table

Furthermore, we consider 4 general cases for all possible values of i, j, k where $i \geq k$ These cases are $(=\ \ =)$, $(=\ \ <)$, $(<\ \ =)$ and $(<\ \ <)$

Case 0: $\gamma = \beta = h$

For this case, the vectors to sum will be

$$\begin{array}{c} (\boldsymbol{\phi}_1\ \boldsymbol{\phi}'_2\ \cdots) \\ (\boldsymbol{\phi}_2\ \boldsymbol{\phi}''_2\ \cdots) \\ (\boldsymbol{\phi}_3\ \boldsymbol{\phi}_2\ \cdots) \\ \hline (\boldsymbol{\phi}''_2\ \mathbf{0}_3\ \cdots) \end{array}$$

and the derived vector will be $(\boldsymbol{\phi}''_2\ \mathbf{0}_3\ \cdots)$ with a weight of $w_p = 2$

Case 1a: $\gamma = h < \beta$
vectors to sum:

$$\begin{array}{r} (\boldsymbol{\phi}_1\ \boldsymbol{\phi}'_2\ \boldsymbol{\phi}'_2\ \boldsymbol{\phi}'_2\ \boldsymbol{\phi}'_2\ \cdots) \\ (\mathbf{0}_3\ \cdots\ \mathbf{0}_3\ \boldsymbol{\phi}_2\ \boldsymbol{\phi}''_2\ \cdots) \\ +(\boldsymbol{\phi}_3\ \boldsymbol{\phi}_2\ \boldsymbol{\phi}_2\ \boldsymbol{\phi}_2\ \boldsymbol{\phi}_2\ \cdots) \\ \hline (\boldsymbol{\phi}'_2\ \boldsymbol{\phi}''_2\ \boldsymbol{\phi}''_2\ \boldsymbol{\phi}'_2\ \mathbf{0}_3\ \cdots) \end{array}$$

derived vector : $(\boldsymbol{\phi}'_2\ (\boldsymbol{\phi}''_2)_{\beta-h-1}\ \boldsymbol{\phi}'_2\ \mathbf{0}_3\ \cdots)$
Parity weight:

$$w_p = 2(\beta - h) + 2 = 2\beta + 2 \tag{0-4}$$

Case 1b: $\beta = h < \gamma$
vectors to sum:

$$\begin{array}{r} (\mathbf{0}_3\ \cdots\ \cdots\ \mathbf{0}_3\ \boldsymbol{\phi}_1\ \boldsymbol{\phi}'_2\ \cdots) \\ (\boldsymbol{\phi}_2\ \boldsymbol{\phi}''_2\ \cdots\ \boldsymbol{\phi}''_2\ \boldsymbol{\phi}''_2\boldsymbol{\phi}''_2\ \cdots) \\ +(\boldsymbol{\phi}_3\ \boldsymbol{\phi}_2\ \cdots\ \boldsymbol{\phi}_2\ \boldsymbol{\phi}_2\ \boldsymbol{\phi}_2\ \cdots) \\ \hline (\boldsymbol{\phi}''_1\ \boldsymbol{\phi}'_2\ \cdots\ \boldsymbol{\phi}'_2\ \boldsymbol{\phi}_3\ \mathbf{0}_3\ \cdots) \end{array}$$

derived vector : $(\boldsymbol{\phi}''_1\ (\boldsymbol{\phi}_2)_{\gamma-h-1}\ \boldsymbol{\phi}_3\ \mathbf{0}_3\ \cdots)$
Parity weight:

$$w_p = 2(\gamma - h) + 2 = 2\gamma + 2 \tag{0-5}$$

Case 2a: $h < \gamma = \beta$
vectors to sum:

$$\begin{array}{r}
(\mathbf{0}_3 \cdots \ \cdots \ \mathbf{0}_3 \ \phi_1 \ \phi_2' \ \cdots) \\
(\mathbf{0}_3 \cdots \ \cdots \ \mathbf{0}_3 \ \phi_2 \ \phi_2'' \ \cdots) \\
+(\phi_3 \ \phi_2 \ \cdots \ \phi_2 \ \phi_2 \ \phi_2 \ \cdots) \\
\hline
(\boldsymbol{\phi}_3 \ \boldsymbol{\phi}_2 \ \cdots \ \boldsymbol{\phi}_2 \ \boldsymbol{\phi}_1 \ \mathbf{0}_3 \ \cdots)
\end{array}$$

derived vector : $(\boldsymbol{\phi}_3 \ (\boldsymbol{\phi}_2)_{\gamma-h-1} \ \boldsymbol{\phi}_1 \ \mathbf{0}_3 \ \cdots)$
Parity weight:

$$w_p = 2(\gamma - h) + 2 = 2\gamma + 2 \tag{0-6}$$

Case 3a: $h < \gamma < \beta$
vectors to sum:

$$\begin{array}{r}
(\mathbf{0}_3 \cdots\cdots \ \mathbf{0}_3 \ \phi_1 \ \phi_2' \ \cdots \ \phi_2' \ \phi_2' \ \phi_2' \cdots) \\
(\mathbf{0}_3 \cdots \ \cdots \ \cdots \ \cdots \ \cdots \ \mathbf{0}_3 \ \phi_2 \ \phi_2'' \cdots) \\
+(\phi_3 \ \phi_2 \ \cdots \ \phi_2 \ \phi_2 \ \phi_2 \ \cdots \ \phi_2 \ \phi_2 \ \phi_2 \cdots) \\
\hline
(\boldsymbol{\phi}_3 \ \boldsymbol{\phi}_2 \cdots \ \boldsymbol{\phi}_2 \ \boldsymbol{\phi}_1' \ \boldsymbol{\phi}_2'' \cdots \ \boldsymbol{\phi}_2'' \ \boldsymbol{\phi}_2' \ \mathbf{0}_3 \cdots)
\end{array}$$

derived vector : $(\boldsymbol{\phi}_3 \ (\boldsymbol{\phi}_2)_{\gamma-h-1} \ \boldsymbol{\phi}_1' \ (\boldsymbol{\phi}_2'')_{\beta-\gamma-1} \ \boldsymbol{\phi}_2' \ \mathbf{0}_3 \ \cdots)$
Parity weight:

$$\begin{aligned}
w_p &= 2(\gamma - h) + 2 + 2(\beta - i) \\
&= 2(\beta - h) + 2 \\
&= 2\beta + 2
\end{aligned} \tag{0-7}$$

Case 3b: $h < \beta < \gamma$

$$\begin{array}{r}
(\mathbf{0}_3 \ \cdots \ \cdots \ \cdots \ \cdots \ \cdots \ \mathbf{0}_3 \ \phi_1 \ \phi_2' \cdots) \\
(\mathbf{0}_3 \ \cdots\cdots \ \mathbf{0}_3 \ \phi_2 \ \phi_2'' \ \cdots \ \phi_2'' \ \phi_2'' \ \phi_2'' \cdots) \\
+(\phi_3 \ \phi_2 \ \cdots \ \phi_2 \ \phi_2 \ \phi_2 \ \cdots \ \phi_2 \ \phi_2 \ \phi_2 \cdots) \\
\hline
(\boldsymbol{\phi}_3 \ \boldsymbol{\phi}_2 \cdots \ \boldsymbol{\phi}_2 \ \mathbf{0}_3 \ \boldsymbol{\phi}_2' \cdots \ \boldsymbol{\phi}_2' \ \boldsymbol{\phi}_3 \ \mathbf{0}_3 \cdots)
\end{array}$$

derived vector : $(\boldsymbol{\phi}_3 \ (\boldsymbol{\phi}_2)_{j-k-1} \ \mathbf{0}_3 \ (\boldsymbol{\phi}_2')_{i-j-1} \ \boldsymbol{\phi}_3 \ \mathbf{0}_3 \ \cdots)$
Parity weight:

$$\begin{aligned}
w_p &= 2(\beta - h) + 1 + 2(\gamma - \beta) + 1 \\
&= 2(\gamma - h) + 2 \\
&= 2\gamma + 2
\end{aligned} \tag{0-8}$$

From all the above cases we can conclude that the parity weight for a weight-3 RTZ sequence may be calculated as

$$w_p = 2l + 2 \tag{0-9}$$

where $l = \max\{\gamma, \beta\} - k = \max\{\gamma, \beta\}$ since $k = 0$

Assuming that after interleaving, another weight-3 RTZ input is produced. Let γ', β', h', l' and w'_p be similarly defined. Then the Hamming weight w_H of the turbo codeword produced can be calculated as

$$w_H = 7 + 2(l + l') \tag{0-10}$$

□

Hamming weight for W4RTZ Turbo Codewords

According to [SunTakeshita] the equation for the Hamming weight $w_H^{(4)}$ for a codeword generated by a W4RTZ is given by

$$w_H^{(4)} = 6m + 2\Big(\sum_{i=1}^{m} \alpha_i + \sum_{i=1}^{m} \alpha'_i\Big) \tag{0-11}$$

where $m = w/2 = 4/2 = 2$

The above equation is only accurate when in both component codes, the given W4RTZ is such that $h < h + \alpha_1 < h' < h' + \alpha_2$. When in both component codes, the given W4RTZ is such that $h < h' < h + \alpha_1 < h' + \alpha_2$ or $h < h' < h' + \alpha_2 < h + \alpha_1$

$$w_H^{(4)} = 4 + w_p + w'_p \tag{0-12}$$

where $w_p = \lfloor \frac{2f_1}{3} \rfloor + \lfloor \frac{2f_3}{3} \rfloor + \lfloor \frac{2(f_2-1)}{3} \rfloor + 2$ and
$w'_p = \lfloor \frac{2f'_1}{3} \rfloor + \lfloor \frac{2f'_3}{3} \rfloor + \lfloor \frac{2(f'_2-1)}{3} \rfloor + 2$

Proof. case: $h < h' < h + \alpha_1 < h' + \alpha_2$

let $f_1 = h' - h, f_2 = h + \alpha_1 - h', f_3 = h' + \alpha_2 - h + \alpha$ for f_1 and f_3 the bits do not overlap and the weight is maintained. The weights $(w_{f_1},\ w_{f_3})$ can be calculated using the equations below

$$w_{f_1} = \lfloor \frac{2f_1}{3} \rfloor + 1$$

and

$$w_{f_3} = \lfloor \frac{2f_3}{3} \rfloor + 1$$

For f_2, the bits do overlap and the overall weight is reduced The weight for f_2 (w_{f_2}) is calculated using the equation below $w_{f_2} = \lfloor \frac{2f_2-1}{3} \rfloor$

the total parity weight

$$\begin{aligned} w_p &= w_{f_1} + w_{f_2} + w_{f_2} \\ &= \lfloor \frac{2f_1}{3} \rfloor + \lfloor \frac{2f_3}{3} \rfloor + \lfloor \frac{2(f_2-1)}{3} \rfloor + 2 \end{aligned} \tag{0-13}$$

Assuming a similar W4RTZ occurs in the 2nd component code, w'_p will be calculated similarly to w_p and the total Hamming weight will be

$$w_H^{(4)} = 4 + w_p + w'_p \tag{0-14}$$

case: $h < h' < h' + \alpha_2 < h + \alpha_1$ let $f_1 = h' - h, f_2 = h + \alpha_2 - h', f_3 = h + \alpha_1 - (h + \alpha_2)$. For f_1 and f_3 the bits do not overlap and the weight is maintained. The weights $(w_{f_1},\ w_{f_3})$ can be calculated using the equations below

$$w_{f_1} = \lfloor \frac{2f_1}{3} \rfloor + 1$$

and

$$w_{f_3} = \lfloor \frac{2f_3}{3} \rfloor + 1$$

For f_2, the bits do overlap and the overall weight is reduced The weight for f_2 (w_{f_2}) is calculated using the equation below $w_{f_2} = \lfloor \frac{2f_2-1}{3} \rfloor$

the total parity weight

$$\begin{aligned} w_p &= w_{f_1} + w_{f_2} + w_{f_2} \\ &= \lfloor \frac{2f_1}{3} \rfloor + \lfloor \frac{2f_3}{3} \rfloor + \lfloor \frac{2(f_2-1)}{3} \rfloor + 2 \end{aligned} \tag{0-15}$$

Assuming a similar W4RTZ occurs in the 2nd component code, w'_p will be calculated similarly to w_p and the total Hamming weight will be

$$w_H^{(4)} = 4 + w_p + w'_p \tag{0-16}$$

□

0.3 Permutation Matrix Design

In this section, we outline the procedure for selecting a good permutation matrix $\mathbf{\Pi}$ with respect to W2RTZs and W3RTZs.

0.3.1 Permutation Matrix selection for W2RTZs

From the definition of Weight-2 RTZ inputs in the previous section, we know that the index of the "1" bits are in the same coset. Our aim is to make sure that the permutation matrix we select enbles the interleaver that we design to either break such weight-2 RTZ inputs or convert it into a large separation weight-2 RTZ. The condition to break weight-2 RTZs is given as

$$\pi_j^{(i)} \neq \pi_j^{(i')},\ |i - i'| \leq N_c \tag{0-17}$$

Since $\mathbf{\Pi}$ consisting of τ elements, the maximum length of column elements consisting of values different from each other is τ. Thus, the cut-off interleaver length for which (**??**) is satisfied is $N_c^2 = \tau^2 = 9$. For this interleaver length, we investigate 3 different compositions of permutation matrices that can be used to achieve this condition in in **??**

1. One cycle permutation: Each row is permutation of the sequence $(0, 1, 2)$. Setting the element at the first row and first column to 0, there are exactly 4 permutation matrices that exist for cut-off length N_c^2. Let

$$\boldsymbol{\psi} = \begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix},\ \boldsymbol{\psi}' = \begin{bmatrix} 0 \\ 2 \\ 1 \end{bmatrix}$$

   We then have

$$\begin{aligned}
[\boldsymbol{\psi}, \mathscr{S}^1[\boldsymbol{\psi}], \mathscr{S}^2[\boldsymbol{\psi}]] &= \begin{bmatrix} 0 & 1 & 2 \\ 1 & 2 & 0 \\ 2 & 0 & 1 \end{bmatrix} := \boldsymbol{\psi}(\boldsymbol{\psi}) \\
[\boldsymbol{\psi}', \mathscr{S}^1[\boldsymbol{\psi}'], \mathscr{S}^2[\boldsymbol{\psi}']] &= \begin{bmatrix} 0 & 1 & 2 \\ 2 & 0 & 1 \\ 1 & 2 & 0 \end{bmatrix} := \boldsymbol{\psi}(\boldsymbol{\psi}') \\
[\boldsymbol{\psi}, \mathscr{S}^2[\boldsymbol{\psi}], \mathscr{S}^1[\boldsymbol{\psi}]] &= \begin{bmatrix} 0 & 2 & 1 \\ 2 & 1 & 0 \\ 1 & 0 & 2 \end{bmatrix} := \boldsymbol{\psi}'(\boldsymbol{\psi}) \\
[\boldsymbol{\psi}', \mathscr{S}^2[\boldsymbol{\psi}'], \mathscr{S}^1[\boldsymbol{\psi}']] &= \begin{bmatrix} 0 & 2 & 1 \\ 1 & 0 & 2 \\ 2 & 1 & 0 \end{bmatrix} := \boldsymbol{\psi}'(\boldsymbol{\psi}')
\end{aligned} \tag{0-18}$$

2. Two cycle permutation: Two rows are permutation of the sequence $(0, 0, 1, 1, 2, 2)$.

   There are no permutation matrices that satisfying cut-off length N_c^2. This is because the sequence length is not divisible by N_c^2, there will always be 2 elements of the same value in each row of $\mathbf{\Pi}$

3. Three cycle permutation: Three rows are permutation of the sequence$(0, 0, 0, 1, 1, 1, 2, 2, 2)$.

   Example of the permutation matrices satisfying cut-off length N_c^2 are shown in **??**

Table ?? shows all unique coset interleaving arrays of length N_c that convert weight-2 RTZ inputs to non-RTZ inputs. They are labeled from A to X. A coset interleaving array is unique if a shift of the elements in the array does not produce another another coset interleaving array.

A	$\begin{bmatrix}0&0&0\\1&1&1\\2&2&2\end{bmatrix}$	B	$\begin{bmatrix}0&0&0\\1&1&2\\2&2&1\end{bmatrix}$	C	$\begin{bmatrix}0&0&0\\1&2&1\\2&1&2\end{bmatrix}$	D	$\begin{bmatrix}0&0&0\\1&2&2\\2&1&1\end{bmatrix}$
E	$\begin{bmatrix}0&0&0\\2&1&1\\1&2&2\end{bmatrix}$	F	$\begin{bmatrix}0&0&0\\2&1&2\\1&2&1\end{bmatrix}$	G	$\begin{bmatrix}0&0&0\\2&2&1\\1&1&2\end{bmatrix}$	H	$\begin{bmatrix}0&0&0\\2&2&2\\1&1&1\end{bmatrix}$
I	$\begin{bmatrix}0&0&1\\1&1&0\\2&2&2\end{bmatrix}$	J	$\begin{bmatrix}0&0&1\\1&2&0\\2&1&2\end{bmatrix}$	K	$\begin{bmatrix}0&0&1\\2&1&0\\1&2&2\end{bmatrix}$	L	$\begin{bmatrix}0&0&1\\2&2&0\\1&1&2\end{bmatrix}$
M	$\begin{bmatrix}0&0&2\\1&1&0\\2&2&1\end{bmatrix}$	N	$\begin{bmatrix}0&0&2\\1&2&0\\2&1&1\end{bmatrix}$	O	$\begin{bmatrix}0&0&2\\2&1&0\\1&2&1\end{bmatrix}$	P	$\begin{bmatrix}0&0&2\\2&2&0\\1&1&1\end{bmatrix}$
Q	$\begin{bmatrix}0&1&0\\1&0&1\\2&2&2\end{bmatrix}$	R	$\begin{bmatrix}0&1&0\\1&0&2\\2&2&1\end{bmatrix}$	S	$\begin{bmatrix}0&1&0\\1&2&1\\2&0&2\end{bmatrix}$	T	$\begin{bmatrix}0&1&0\\2&0&1\\1&2&2\end{bmatrix}$
U	$\begin{bmatrix}0&1&0\\2&0&2\\1&2&1\end{bmatrix}$	V	$\begin{bmatrix}0&1&0\\2&2&1\\1&0&2\end{bmatrix}$	W	$\begin{bmatrix}0&1&1\\1&2&0\\2&0&2\end{bmatrix}$	X	$\begin{bmatrix}0&2&0\\2&0&2\\1&1&1\end{bmatrix}$

Table 2: All unique coset interleaving arrays of length $N_c = 9$ for weight-2 RTZ inputs

The interleaver length used in turbo coding are way greater than N_c^2 and it is not possible to transform weight-2 RTZ inputs into non-RTZ inputs for all values of i. All is not lost however, since not all weight-2 RTZ inputs produce low-weight codewords. The formula for calculating the Hamming weight of the Turbo codeword produced by a weight-2 RTZ input occuring in both component codes ($w_H^{(2)}$) is given by [SunTakeshita]

$$\begin{aligned} w_H^{(2)} =& 2 + (2 + \frac{\Delta_c}{\tau})w_0 + (2 + \frac{\Delta_{c'}}{\tau})w_0 \\ =& 6 + \Big(\frac{\Delta_c + \Delta_{c'}}{\tau}\Big)w_0,\ w_0 = 2 \end{aligned} \tag{0-19}$$

For all the Π in Table ?? $\Delta_c = 9 = 3\tau$ and $\Delta_{c'} := (c^t_{(h+\alpha')} - c^t_{(h)})$.

0.3.2 Permutation Matrix selection for W3RTZs

As mentioned earlier, a W3RTZ is formed when the indices of the “1” bits each occur in different cosets. It goes without saying that the simplest way to convert a W3RTZ into a non-W3RTZ is to make sure that at least two of indices of the “1” bits occur within the same coset after interleaving.

The formula for calculating the hamming weight for a turbo code created by a W3RTZ ($w_H^{(3)}$)

$$\begin{aligned} w_H^{(3)} =& 3 + (2l + 2) + (2l' + 2) \\ =& 3 + w_p + w'_p,\ w_p = 2l + 2,\ w'_p = 2l' + 2 \\ =& 7 + 2(l + l') \end{aligned} \tag{0-20}$$

where w_p, w'_p refer to the pre-interleaving parity weight and the post-interleaving parity weight respectively.

In reality, W3RTZs are many and it is impossible to completely get rid of all of them, even within N_c. So in our selection of Permutation Matrices for W3RTZs, we make sure that the remaining W3RTZs have $w_p > 2$. Unique permutation matrices which meet this criteria are shown in Table **??** and they are labeled from A to L

A	$\begin{bmatrix}0&0&0\\1&1&1\\2&2&2\end{bmatrix}$	B	$\begin{bmatrix}0&0&0\\1&1&2\\1&2&2\end{bmatrix}$
C	$\begin{bmatrix}0&0&0\\1&1&2\\2&1&2\end{bmatrix}$	D	$\begin{bmatrix}0&0&0\\1&1&2\\2&2&1\end{bmatrix}$
E	$\begin{bmatrix}0&0&0\\2&2&1\\1&1&2\end{bmatrix}$	F	$\begin{bmatrix}0&0&0\\2&2&1\\1&2&1\end{bmatrix}$
G	$\begin{bmatrix}0&0&0\\2&2&1\\2&1&1\end{bmatrix}$	H	$\begin{bmatrix}0&0&1\\0&1&1\\2&2&2\end{bmatrix}$
I	$\begin{bmatrix}0&0&1\\1&1&2\\2&0&2\end{bmatrix}$	J	$\begin{bmatrix}0&0&2\\0&2&2\\1&1&1\end{bmatrix}$
K	$\begin{bmatrix}0&0&2\\2&2&1\\1&0&1\end{bmatrix}$	L	$\begin{bmatrix}0&1&0\\1&1&2\\2&0&2\end{bmatrix}$

Table 3: All unique permutation matrices of length $N_c = 9$ for weight-3 RTZ inputs

Depending on which permutation matrix is chosen from Table **??**, Equation **??** can be simplified. The value of w_p for the pre-interleaving weight-3 is dependent on the elements in $\mathcal{C}^t$

Let $(c^t_{(h)},\ c^{t+1}_{(h+\beta)},\ c^{t+2}_{(h+\gamma)})$ be the vector representing a weight-3 RTZ input

Without loss of generality, we can assume that $h = t = 0$. We then have

$$l = \max(\beta, \gamma) \tag{0-21}$$

And

$$w_p = 2(\max(\beta, \gamma)) + 2 \tag{0-22}$$

By deciding on the Π we can easily calculate all values of l and w_p. w'_p, β', γ' and l' are similarly defined and are dependent on the elements in $\mathbb{C}^t,\ t = 0, 1, .., \tau - 1$

As an example, Table **??** shows all the weight-3 RTZ inputs and the corresponding equations for calculating w_H

Finally, we need to choose a Permutation Matrix which is able to efectively deal with both W2RTZs and W3RTZs. This part is simple as the only thing that we need to do is to select the permutation matrices that appear in both Table **??** and Table **??**. This leaves us with

$$\begin{bmatrix}0&0&0\\1&1&1\\2&2&2\end{bmatrix}$$

RTZ index	l	w_p	w_H
(0 4 8)	2	6	$11 + 2(\max(\beta', \gamma'))$
(0 5 7)	2	6	$11 + 2(\max(\beta', \gamma'))$
(1 3 8)	2	6	$11 + 2(\max(\beta', \gamma'))$
(1 5 6)	1	4	$9 + 2(\max(\beta', \gamma'))$
(2 3 7)	1	4	$9 + 2(\max(\beta', \gamma'))$
(2 4 6)	1	4	$9 + 2(\max(\beta', \gamma'))$
(0 8 13)	4	10	$15 + 2(\max(\beta', \gamma'))$
(0 4 17)	5	12	$17 + 2(\max(\beta', \gamma'))$
(0 13 17)	5	12	$17 + 2(\max(\beta', \gamma'))$
(0 7 14)	4	6	$15 + 2(\max(\beta', \gamma'))$
(0 5 16)	5	6	$17 + 2(\max(\beta', \gamma'))$
(0 14 16)	5	6	$17 + 2(\max(\beta', \gamma'))$
(1 8 12)	3	8	$13 + 2(\max(\beta', \gamma'))$
(1 3 17)	5	12	$17 + 2(\max(\beta', \gamma'))$
(1 12 17)	5	12	$17 + 2(\max(\beta', \gamma'))$
(1 6 14)	4	10	$15 + 2(\max(\beta', \gamma'))$
(1 5 15)	4	10	$15 + 2(\max(\beta', \gamma'))$
(1 14 15)	4	10	$15 + 2(\max(\beta', \gamma'))$
(2 7 12)	3	8	$13 + 2(\max(\beta', \gamma'))$
(2 3 16)	4	10	$15 + 2(\max(\beta', \gamma'))$
(2 12 16)	4	10	$15 + 2(\max(\beta', \gamma'))$
(2 6 13)	3	8	$13 + 2(\max(\beta', \gamma'))$
(2 4 15)	4	10	$15 + 2(\max(\beta', \gamma'))$
(2 13 15)	4	10	$15 + 2(\max(\beta', \gamma'))$

Table 4: All unique permutation matrices of length $N_c = 9$ for weight-3 RTZ inputs

and

$$\begin{bmatrix} 0 & 0 & 0 \\ 2 & 2 & 2 \\ 1 & 1 & 1 \end{bmatrix}$$

Moving foward we will use the permutation matrix

$$\mathbf{\Pi}^{(0)} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 2 & 2 & 2 \end{bmatrix}$$

in all our considerations. It was chosen because in the design process, we only need to focus on only one of the cosets, say $\mathbb{C}^0$ and replicate the results for the remaining cosets.

0.3.3 Higher Weight RTZ inputs

Higher weight RTZ inputs are made up of some combination of W2RTZs and/or W3RTZs. Let the weight of the higher weight RTZ be represented by w. If $w \bmod 2 = 0$, then the RTZ is made up of $w/2$ W2RTZs. If $w \bmod 2 = 1$, then the RTZ is made up of $\lfloor w/2 \rfloor - 1$ W2RTZs and a single W3RTZ As the weight of the RTZ inputs increases beyond the largest base RTZ, the Permutation Matrices cannot be designed to effectively get rid of these Higher Weight RTZ inputs. However, with the the structure of the Permutation Matrix, we know exactly where these Higher Weight RTZ inputs occur. In addition to the hamming weight for W2RTZs and W3RTZs, we will consider the hamming weight for W4RTZs and W5RTZs in our design of good interleavers. From [SunTakeshita] we deduce that Hamming weight for a Turbo code produced as a result of an even weight RTZ is given by

$$w_H^{(\text{even})} = 6m + 2\Big(\frac{\sum_{i=1}^{m} \Delta_{c_i} + \sum_{i=1}^{m} \Delta_{c'_i}}{\tau}\Big) \tag{0-23}$$

where $m = w/2$ whiles that for an odd weight RTZ is given by

$$\begin{aligned} w_H^{(\text{odd})} &= 6m + 2\Big(\frac{\sum_{i=1}^{m} \Delta_{c_i} + \sum_{i=1}^{m} \Delta_{c'_i}}{\tau}\Big) + 7 + 2(l + l') \\ &= 7 + 6m + 2\Bigg(l + l' + \Big(\frac{\sum_{i=1}^{m} \Delta_{c_i} + \sum_{i=1}^{m} \Delta_{c'_i}}{\tau}\Big)\Bigg) \end{aligned} \tag{0-24}$$

where $m = \lfloor w/2 \rfloor - 1$ For W4RTZs we have

$$w_H^{(4)} = 12 + 2\Big(\frac{\Delta_{c1} + \Delta_{c2} + \Delta_{c'1} + \Delta_{c'2}}{\tau}\Big) \tag{0-25}$$

and for W5RTZs we have

$$w_H^{(5)} = 13 + 2\Bigg(l + l' + \Big(\frac{\Delta_{c_1} + \Delta_{c'_1}}{\tau}\Big)\Bigg) \tag{0-26}$$

These equations will be used as secondary checks to confirm the minimum distance of the turbo code with our designed interleaver

0.4 Coset Design

Once the permutation matrix is decided upon, we have the necessary constraints which will help us design $\mathbb{C}^t$ with respect to base RTZs (W2RTZs, W3RTZ) . We will make use of the Almost Linear Interleaver(ALI) is the design of $\mathbb{C}^0$. The interleaving equation for the ALI(L,D) Interleaver is given by

$$\pi(h) = D \cdot h + \left\lfloor \frac{h}{A} \right\rfloor \bmod L$$

where $A = L/C$ and $C = \gcd(D, L)$ Also, D is the period of the interleaver and $h = 0, 1, ..., L-1$. The value of a coset element at position h will be $3\pi(h) + t$. The resulting coset interleaver will be reffered to by the notation CI(N,D)

0.4.1 Coset Design for W2RTZs

$w_H^{(2)}$ is calculated using (**??**) It is convinient to write $\Delta_{c'}$ in terms of D

$$c^t_{(h'+\alpha')} = 3 * (D(h+\alpha) + \lfloor \frac{h+\alpha}{A} \rfloor \bmod L)$$

and

$$c^t_{(h')}) = 3 * (D(h) + \lfloor \frac{h}{A} \rfloor \bmod L)$$

where $A = L/C,\ C = \gcd(L, D)$. For the chosen permutation matrix $\mathbf{\Pi}^{(0)}$, we know that W2RTZs occur after 2 repititions and therefore we can set $\alpha = 3$ and without loss of generality, we can also set $h = 0$. A can take on 3 different values,$L,\ L/3,\ 3$.

if $A = L,\ L/3$ $\Delta_{c'}$ simplifies to

$$\begin{aligned}\Delta_{c'} =& 3\Bigg(\min \Big(D(\alpha) \bmod L,\ L - (D(\alpha) \bmod L)\Big)\Bigg) \\ =& 3\Bigg(\min \Big(3D \bmod L,\ L - (3D \bmod L)\Big)\Bigg)\end{aligned} \tag{0-27}$$

Feeding this into **??** we get

$$\begin{aligned}w_H^{(2)} =& 6 + 2\Bigg(3 + \frac{3\Big(\min \Big(3D \bmod L,\ L - (3D \bmod L)\Big)\Big)}{3}\Bigg) \\ =& 6 + 2\Big(3 + \min \Big(3D \bmod L,\ L - (3D \bmod L)\Big)\Big)\end{aligned}$$

For the case where $A = 3$

$$\begin{aligned}\Delta_{c'} =& 3\Bigg(\min \Big(D(\alpha) + 1 \bmod L,\ L - (D(\alpha) + 1 \bmod L)\Big)\Bigg) \\ =& 3\Bigg(\min \Big(3D + 1 \bmod L,\ L - (3D + 1 \bmod L)\Big)\Bigg)\end{aligned} \tag{0-28}$$

Substituting into (**??**) ,we have

$$\begin{aligned} w_H^{(2)} =&6+2\Big(3+\frac{3\Bigg(\min\Big(3D+1 \bmod L,\ L-(3D+1 \bmod L)\Big)\Bigg)}{3}\Big)\\ =&6+2\Big(3+\min\Big(3D+1 \bmod L,\ L-(3D+1 \bmod L)\Big)\Big) \end{aligned}$$

Combining all the equations gives us

$$w_H^{(2)} = \begin{cases} 6+2\Big(3+\min\Big(3D \bmod L,\ L-(3D \bmod L)\Big)\Big)\Big), & A=L/3 \text{ or } A=L \\ 6+2\Big(3+\min\Big(3D+1 \bmod L,\ L-(3D+1 \bmod L)\Big)\Big), & A=3 \end{cases} \tag{0-29}$$

0.4.2 Coset Design for W3RTZs

Unlike the W2RTZs, W3RTZs exist for the first and second repititions of $\mathbf{\Pi}^{(0)}$. These RTZs can potentially reduce the $d_{\min}$ value of the turbo code and coset design is carried out to prevent that.

The W3RTZs as well as the various equations to calculate the resultant codeword weight are given in Table **??**. Since the design for $\mathbb{C}^0$ is applied to the other cosets we should be able to calculate the minimum value for $w_H^{(3)}$ with respect to Table **??** whiles focusing on $\mathbb{C}^0$ only. The positions within 2 repititions of $\mathbf{\Pi}^{(0)}$ where a weight-3 inputs occur are given with respect to the complete interleaver and need to be scaled down to $\mathbb{C}^0$ Let $h,\ h+\beta,\ h+\gamma$ be the inputs representing where the weight-3 RTZ inputs occur due to $\mathbf{\Pi}^{(0)}$. Then the scaled down versions will be $h^{(0)},\ (h+\beta)^{(0)},\ (h+\gamma)^{(0)}$ and are calculated using the equation

$$f(x) = x \bmod 3 + 3\Big(\Big\lfloor \frac{x}{9} \Big\rfloor\Big) \tag{0-30}$$

We feed $h^{(0)},\ (h+\beta)^{(0)},\ (h+\gamma)^{(0)}$ into the ALI and we get

$$\boldsymbol{s} = (\pi(h^{(0)}),\ \pi((h+\beta)^{(0)}),\ \pi((h+\gamma)^{(0)})) - \min(\pi(h^{(0)}),\ \pi((h+\beta)^{(0)}),\ \pi((h+\gamma)^{(0)})) \tag{0-31}$$

and

$$l' = \max(\beta,\gamma) = \boldsymbol{s}_{\max} - \boldsymbol{s}_{\min} = \boldsymbol{s}_{\max} \tag{0-32}$$

When all l' values for the $W3RTZs$ in Table **??** are calculated, we take note of the ones that give the least value of $w_H^{(3)}$. Then for every shifted version of that $W3RTZ$ we calculate $w_H^{(3)}$ and select the minimum value from all calculated $w_H^{(3)}$ to get the true minimum value.

Below is a summary of the steps involved in determining the Hamming weight for all weight-3 RTZ associated with $\Pi^{(0)}$

1. convert $h,\ h+\beta,\ h+\gamma$ into $h^{(0)},\ (h+\beta)^{(0)},\ (h+\gamma)^{(0)}$ using (**??**)

2. input the indices into the ALI equation to obtain $susing$(**??**)

3. find the value of l' and Hamming weight using (**??**) and corresponding equation in Table **??**

4. repeat above steps for all shifted versions of the W3RTZ that produced the least value of $w_H^{(3)}$, where each shift is a multiple of τ^2

5. Finally, the least Hamming weight value associated with the shifted version of the W3RTZ input(s) is selected.

If we assume that the start position for all the cosets is the same, then the above calculations with respect to the weight-3 RTZ inputs are sufficient, however if we desire to apply a shift factor to all cosets but $\mathbb{C}^{(0)}$, a slight change in notation will be needed.

First off we adjust the notation for a Coset interleaver from $\mathrm{CI}(N, D)$ to $\mathrm{CI}(N, D, s_1, s_2)$ where $s_1,\ s_2$ indicate the start position for $\mathbb{C}^1,\ \mathbb{C}^2$ respectively. It is worth noting that the value of D used in the ALI(L,D) is the same for all the cosets, just that $\mathbb{C}^1,\ \mathbb{C}^2$ are shifted by $s_1,\ s_2$ respective positions to the right.

This means that values of $h^{(0)},\ (h+\beta)^{(0)},\ (h+\gamma)^{(0)}$ need to be adjusted by $0,\ s_1,\ s_2$ respectively. This statement is made assuming that $h^{(0)},\ (h+\beta)^{(0)},\ (h+\gamma)^{(0)}$ are in $\mathbb{C}^0, \mathbb{C}^1, \mathbb{C}^2$ respectively.

Below is a summary of the steps involved in determining the Hamming weight for all weight-3 RTZ associated with $\Pi^{(0)}$ when $s_1, s_2 > 0$

1. convert $h,\ h+\beta,\ h+\gamma$ into $h^{(0)},\ (h+\beta)^{(0)},\ (h+\gamma)^{(0)}$ using (**??**)

2. input the indices $h^{(0)},\ (h+\beta)^{(0)}+s_1,\ (h+\gamma)^{(0)}+s_2$ into the ALI equation to obtain $\boldsymbol{s}$ using (**??**)

3. find the value of l' and Hamming weight using (**??**) and corresponding equation in Table **??**

4. repeat above steps for all shifted versions of the W3RTZ that produced the least value of $w_H^{(3)}$, where each shift is a multiple of τ^2

5. Finally, the least Hamming weight value associated with the shifted version of the W3RTZ input(s) is selected.

0.5 Simulation Results and Discussion

Simulations for the coset interleaver CI(N, D, s_1, s_2) interleaver are done for $s_1, s_2 = 0$. The values for $D = \{28, 29, ..., 35\}$, $N = 261$ and

$$\Pi = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 2 & 2 & 2 \end{bmatrix}$$

The minimum weight for each interleaver with respect to W2RTZs, W3RTZs and W4RTZs are shown in Table(**??**) and the simulation results are shown in Figure() As expected, the interleaver

D	$w_H^{(2)}$	$w_H^{(3)}$	$w_H^{(4)}$
28	18	115	14
29	14	123	14
30	18	121	14
31	24	103	14
32	30	89	14
33	36	75	14
34	42	61	14
35	48	49	14

Table 5: Minimum Hamming weight for weight-2 and weight-3 RTZ using CI(N, D, s_1, s_2), where $s_1, s_2 = 0$

designed with CI$(261, 29)$ performs the worst, but the performance of the other interleavers does not perform as expected of the data from Table **??**. Further examination of the simulation results reveals that even though the other interleavers have high Hamming weight related to weight-2 and weight-3 RTZ inputs, the minimum distance of the code seem to be bound at by RTZ inputs with higher weights, specifically those of weight-4.

Since the start index for picking the elements for all the cosets are the same, this causes well separated pre-interleaving inputs to be bunched together post-interleaving. This is shown in the graph of the input output relation of any of the interleavers.

This can be remedied by making sure that the start position for the other cosets are different and this can be accomplished by shifting the elements of the cosets other than $\mathbb{C}^0$ a certain value to the left. Let a, b be the factor by which the $\mathbb{C}^1$ and $\mathbb{C}^2$ are shifted. We proceed to re-design CI$(23, 0, 0)$ by introducing $s_1 = 30$ and $s_2 = 7$. The simulation results for CI$(261, 23, 0, 0)$ and CI$(261, 23, 30, 7)$ are shown in Figure() as can be seen adjusting the start position of the other cosets, the error-correcting performance is greatly improved but the Hamming distance for the turbo code is again bounded by weight-4 RTZ inputs. A specific example is that the weight-4 input of the form $(1 + x^{\tau}) + (x^{155})(1 + x^{\tau})$ is transformed into another weight-4 RTZ input of the form$(1 + x^{\tau}) + (x^{244})(1 + x^{\tau})$. Then

$$w_H = w_m + w_p + w_{p'} = 4 + 8 + 8 = 20$$

. This means that with this with this design, the condition $\Delta_{c'1} \neq \Delta_{c1}$ and $\Delta_{c'2} \neq \Delta_{c2}$ was not met. In the next section, we attempt an an extra step to the coset redesign in order increase the value of $w_H^{(4)}$

0.6 Coset Design for W4RTZs

The coset interleaver with parameters D, s_1, s_2 carefully chosen is able to effectively deal with W2RTZs and W3RTZs. It is however unable to effectively deal with weight-4 RTZ inputs. The composition of all the cosets is the same and this brings about regularity characteristic in the overall interleaver. This regularity whiles desirable makes it difficult for W4RTZ to be effectively broken by the interleaver.

0.6.1 Dealing with weight-4 RTZ inputs

To deal with weight-4 RTZ inputs we need to introduce some kind of controlled irregularity into our design of the coset interleaver. This is acheived by performing coset design at the inner coset $\mathbb{C}^{tt'}$. Lets begin by focusing on $\mathbb{C}^{0t}$. The following notations will be used.

1. D_0 is the angular shift and is the same for all $\mathbb{C}^{tt'}$ and is used as a parameter in the ALI
2. $D_1,\ D_2$ represent to first element in $\mathbb{C}^{01}$ and $\mathbb{C}^{02}$ respectively.

For each element in $\mathbb{C}^{00}$ is calculated as

$$c_m^{00} = D_0 m + \lfloor \frac{m}{A} \rfloor \text{ mod } M$$

, where

$$A = M/C,\ C = gcd(M, D_0)$$

. Elements in $\mathbb{C}^{01}$ and $\mathbb{C}^{02}$ are calculated as follows

$$c_m^{01} = D_0 m + \lfloor \frac{m}{A} \rfloor + D_1 \text{ mod } M$$

and

$$c_m^{02} = D_0 m + \lfloor \frac{m}{A} \rfloor + D_2 \text{ mod } M$$

With respect to $\mathbb{C}^{1t'}$ and $\mathbb{C}^{2t'}$ we have

$$c_m^{10} = c_m^{01},\ c_m^{11} = c_m^{02},\ c_m^{12} = c_m^{00}$$

and

$$c_m^{20} = c_m^{02},\ c_m^{21} = c_m^{00},\ c_m^{22} = c_m^{01}$$

respectively. To obtain the corresponding values for $\mathbb{C}^t$, we perform the simple operation shown below

$$c_h^t = 3(3c_m^{tt'} + t') + t \tag{0-33}$$

where

$$m = \left\lfloor \frac{h}{\tau} \right\rfloor,\ t' = \text{mod}(m, \tau), t = 0, 1, ..., \tau - 1,\ h = 0, 1, ..., C - 1$$

After the index sets $\mathbb{C}^t$ have been determined, s_1, s_2 may be used to further adjust the start point of $\mathbb{C}^1$ and $\mathbb{C}^2$ respectively. The above process introduces some chaos into $\mathbb{C}^t$ which will be useful in dealing with weight-4 RTZ inputs. We show how to calculate the respective Hamming weights with respect to the changed design.

0.6.2 Weight-2 RTZ inputs Hamming Weight

w_H for weight-2 RTZ inputs is easily determined. For all $\mathbb{C}^{tt'}$ the separation between each element is the same and for simplicity sake we will use $\mathbb{C}^{00}$ to rewrite **??**

we know that

$$c_m^{00} = D_0 m + \lfloor \frac{m}{A} \rfloor \bmod M$$

From $\mathbf{\Pi}^{(0)}$ we can deduce that W2RTZs in both component codes corresponds to the m and $m+1$ elements being picked and the difference between the mth and $(m+1)$th element is either $D \bmod M$ or $M - (D \bmod M)$. In terms of the hth and $(h+\alpha)$th element in $\mathbb{C}^0, \Delta_{c'} = 3 \cdot \min\left(D \bmod M, M - (D \bmod M)\right)$

Upon substitution, our Hamming weight equation becomes

$$\begin{aligned} w_H^{(2)} =& 6 + 2\Big(3 + \frac{3(3 \cdot \min\left(D \bmod M, M - (D \bmod M)\right))}{3}\Big) \\ =& 6 + 2\Big(3 + 3 \cdot \min\left(D \bmod M, M - (D \bmod M)\right)\Big) \\ =& 6 + 6(1 + \min\left(D \bmod M, M - (D \bmod M)\right)) \end{aligned} \tag{0-34}$$

0.6.3 Weight-3 RTZ inputs Hamming Weight

From the choice of $\mathbf{\Pi}^{(0)}$, we are able to know the W3RTZs which potentially cause low-weight turbo codes. Let us represent the pre-interleaving weight-3 RTZ inputs by the vector $\boldsymbol{n} = (n_0, n_1, n_2),\ \boldsymbol{n} \in \mathcal{N}$. Also t, l, m, t' in (**??**) are calculated using the following equations

$$\begin{aligned} t_i &= n_i \bmod 3 \\ h_i &= \left\lfloor \frac{n_i}{3} \right\rfloor \\ m_i &= \left\lfloor \frac{h_i}{3} \right\rfloor \\ t'_i &= h_i \bmod 3 \end{aligned} \tag{0-35}$$

In calculating the corresponding hamming weight for the weight-3 RTZ we only need the $3c_m^{tt'} + t'$ portion of **??** and the corresponding vector $\boldsymbol{y}$ as a result of feeding $\boldsymbol{n}$ into it is given by

$$= \Big((3c_{m_0}^{t_0 t'_0} + t'_0,\ 3c_{m_1}^{t_1 t'_1} + t'_1,\ 3c_{m_2}^{t_2 t'_2} + t'_2) - \min(3c_{m_0}^{t_0 t'_0} + t'_0,\ 3c_{m_1}^{t_1 t'_1} + t'_1,\ 3c_{m_2}^{t_2 t'_2} + t'_2)\Big)$$

and $l' = \max(\beta, \gamma) = y_{\max}$

0.6.4 Weight-4 RTZ Inputs Hamming Weight

$\mathbb{N}$