“ Coset Interleaver Idea”

Kwame Ackah Bohulu

February 26, 2020

0.1 Introduction

We wish to design an interleaver which maximizes the minimum distance of a Turbo code constructed using a specific component code for a given interleaver length N. To acheive this objective, we need to investigate which input messages are likely to produce codewords with low distance and based on this knowledge design an interleaver is such a way as to prevent such input messages from being mapped. We refer to such input messages as error messages.

For the 5/7 RSC interleaver, we realize that weight-2 and weight-3 error messages are the most critical and any error message with higher weight than 3 is a combination of either or both.

0.2 Weight-2 Error Messages and Low Weight Codewords in Turbo Codes

It is common knowledge that the components of the Turbo encoder are two(usually identical) Recursive Systematic Convolutional (RSC) encoders and an interleaver. The input to the first RSC encoder (systematic bits) as well as outputs from these RSC encoders (parity check bits) are concatenated in parallel to form a Turbo codeword.

The inputs to the two RSC encoders are the systematic bits and an interleaved version of the systematic bits respectively. The weight of the Turbo codeword is the sum of the weight of the systematic bits and the parity bits from both RSC encoders. The Turbo codeword with the least weight determines the minimum Hamming distance of the turbo code and its known as the minimum weight codeword. The minimum distance of a turbo code determines its error correcting performance and the aim is to make it as large as possible.

Returning to the input of the RSC encoder, it is worth noting that there exists certain systematic bit sequences that produce low weight parity check bit sequences and by extension, low weight codewords. We will refer to such systematic bit sequences as weight-n error messages ($n = 2, 3, 4, ...$). If the input to both RSC encoders is such an error messages, then it is highly likely that the Turbo codeword generated will also have a low weight. To prevent this from happening, we need to somehow alter the structure of the weight-n error message at atleast one RSC encoders. Due to the design of the Turbo code not much can be done to alter the systematic bit sequences present at the first RSC encoder and the best we can do is to hope that no weight-n error message is input into it. However, since the input to the second RSC encoder is an interleaved version of the input to the first RSC encoder, we can change a weight-n error message into a different systematic bit sequence.

To design an interleaver which enables us to transform weight-n error message into good systematic bit sequences for a given component code, we need to know the structure of these weight-n error messages. In the design of interleavers for turbo codes over the years, weight-2 systematic bit sequences have been a reference point for the design and as such we will also begin from there as well.

0.2.1 Weight-2 Error Messages for 5/7 RSC Encoder

In general, any systematic bit sequence of the form $x^i + x^{i+m\tau}$ $\forall i,\ i = 0, 1, 2, ..., N - m\tau,\ m = 1, 2, 3$ is a weight-2 error message. N is the length of the systematic bit sequence (and by extension, the length of the interleaver), τ is the cyclic length of the RSC encoder ($\tau = 3$ for the 5/7 RSC encoder). It is worth noting that i and $i + m\tau$ are the positions in the input sequence where bit "1" occurs. Also, the smaller the value of m,the lower the weight of the parity check bit. More information can be gained for interleaver design by taking the following approach.

For a systematic bit sequence of length N let $i = \{0, 1, 2, ..., N-1\}$ be the set representing the bit positions indices of that input sequence and let $\Pi(i)$ be an interleaved version of the set i. We can group x into τ cosets. For the 5/7 RSC encoder, we will have 3 cosets, S_0, S_1 and S_2 where S_p contain elements of x such that $i \mod \tau = p$. Having done this, we can observe that weight-2 error messages occur when the bit position indices i and $i + m\tau$ are in the same coset. This means that, to transform weight-2 error messages into good systematic bit sequnces, either of the following conditions need to be met.

1. $\forall\ i$The bits at index i and $i + m\tau$ should be mapped to bits whose indices are in different cosets (most ideal situation)

2. $\forall\ i$ if the bits at index i and $i + m\tau$ end up being mapped to bits whose indices are in the same coset, they should be more than m positions apart(least ideal condition)

0.3 Interleaving Algorithm

To prevent error messages of the type described in the previous section, the following interleaving algorithm is proposed. Let $x = \{0, 1, 2, ...N-1\}$ Let N be the interleaver length and $N_c = N/\tau$ be the size of each coset. Let $j^{(n)}$ represent the index of the elements in each coset, where $n = \{0, 1, .., \tau - 1\}$ and $j^{(n)} = \{0, 1, ..., N_c - 1\}$. Also let $\mathbf{C}_n$ represent the various cosets modulo τ and $\mathbf{C}_n(j^{(n)})$ represent the jth element in coset $\mathbf{C}_n$ Finally,let $D1, D2$ be positive integers.$gcd(D1, N_c) = gcd(D2, N_c) = 1$

- Split the input bit positions into τ cosets such that $\forall i,\ \mathbf{C}_n^{(i)} = n \mod \tau$

- Pick τ elements elements from each coset. Let $\mathbf{m}^{(i)}$ represent the thruple of indexes of the chosen elements.Then, $\mathbf{m}^{(i)}$ is given by $\{D_2 i + D_1(0), D_2 i + D_1(1), D_2 i + D_1(2), ...\} \mod N_c,\ i = \{0, 1, 2, .., N_c - 1\}$

- The interleaved sequence $\pi(x)$ is then given by $\{\mathbf{C}_{i \mod \tau}(\mathbf{m}^{(i)})\ \forall i$

Example For the 5/7 RSC encoder, $\tau = 3$. Let $N = 27$ and $N_c = 9$ $\mathbf{C}_0 = \{0, 3, 6, 9, 12, 15, 18, 21, 24\}, \mathbf{C}_1 = \{1, 4, 7, 10, 13, 16, 19, 22, 25\}, \mathbf{C}_2 = \{2, 5, 8, 11, 14, 17, 20, 23, 26\}$

$\mathbf{m}^{(0)} = \{0, 5, 1\}, \mathbf{m}^{(1)} = \{7, 3, 8\}, \mathbf{m}^{(2)} = \{5, 1, 6\}, \mathbf{m}^{(8)} = \{2, 7, 3\}$

$\pi(x) = \{0, 15, 13, 22, 10, 25, 17, 5, 20, 9, 24, 12, 4, 19, 7, 26, 14, 2, 18, 6, 21, 13, 1, 16, 8, 23, 11\}$

0.4 Higher Weight error messages

For a given interleaver Length N it is possible to choose $D1, D2$ such that weight-2 and weight 3 error messages siscussed in prior sections are not mappped unto each other. If the interleaving is done properly, higher weight error messages will also be mapped to non-error messsages.

We investigate this interleaving method on weight-4 error messages. We rewrite the interleaved sequence in the form below

$$\begin{bmatrix} \mathbf{C}_0(\mathbf{m}^{(0)}) & \mathbf{C}_0(\mathbf{m}^{(3)}) & \mathbf{C}_0(\mathbf{m}^{(6)}) \\ \mathbf{C}_1(\mathbf{m}^{(1)}) & \mathbf{C}_1(\mathbf{m}^{(4)}) & \mathbf{C}_0(\mathbf{m}^{(7)}) \\ \mathbf{C}_2(\mathbf{m}^{(2)}) & \mathbf{C}_2(\mathbf{m}^{(5)}) & \mathbf{C}_0(\mathbf{m}^{(8)}) \end{bmatrix} \tag{0-1}$$

The difference $\mathbf{m}^{(i+1) \mod 3} - \mathbf{m}^{(i) \mod 3}$ is equal to $\{D_2(i+1) - D_2(i), D_2(i+1) - D_2(i), D_2(i+1) - D_2(i) \mod N_c\}$

This suggests that the elements picked from each coset are shifted versions of the elements in $\mathbf{C}_0$ and weight-4 error messages made up of a pair of weight-2 error messages of the form $\alpha^{n \mod \tau} + \alpha^{n+1 \mod \tau}$ or $\alpha^{n \mod \tau} + \alpha^{n+2 \mod \tau}$ map to inputs of the same structure and producing low-weight turbo codewords. And the multiplicity of such codewords increases an N increases.