Thesis

Kwame Ackah Bohulu

April 2, 2019

Chapter 1

Bit Error Probability for Turbo Codes Derivation

1.1 Maximum Likelihood Detection (MLD)

Assume that there exists an input space $\mathbb{R}^M$ with $M = 2^k$ elements. From this input space,a k-bit binary information sequence $\boldsymbol{x}_m = (x_{m,0}, x_{m,1}, ...x_{m,k-1})$ is mapped to an element $\boldsymbol{c}_m = (c_{m,0}, c_{m,1}, ..., c_{m,n-1})$ in the output space $\mathbb{R}^N$ with $N = 2^n$ elements. The M elements in $\mathbb{R}^M$ form a code and $\mathcal{C} \subset \mathbb{R}^n$ The BPSK modulated codeword $\boldsymbol{y}_m = (y_{m,0}, y_{m,1}, ..., y_{m,n-1})$ is transmitted over the AWGN channel and is received at the receiver as $\boldsymbol{r} = (r_0, r_1, r_{n-1})$

The task of the receiver is to obtain an estimate of $\boldsymbol{x}$, $\hat{\boldsymbol{x}_m}$ from $\boldsymbol{r}$. The probability of a correct decision given $\boldsymbol{r}$ $P[\text{correct decision}|\boldsymbol{r}] = P[\hat{\boldsymbol{x}_m} \text{ sent}|\boldsymbol{r}]$ and the probability of a correct decision $P[\text{correct decision}] = \int P[\hat{\boldsymbol{x}_m} \text{ sent}|\boldsymbol{r}]p(\boldsymbol{r})d\boldsymbol{r}$

For optimal accuracy the receiver must decide in favor of the $\boldsymbol{x}_m$ that maximizes $P[\boldsymbol{x}_m|\boldsymbol{r}]$ upon observing $\boldsymbol{r}$

$$\begin{aligned} \hat{\boldsymbol{x}}_m &= \arg\max_{1\leq m\leq m} P[\boldsymbol{x}_m|\boldsymbol{r}] \\ &\arg\max_{1\leq m\leq M} P[\boldsymbol{c}_m|\boldsymbol{r}] \end{aligned} \tag{1-1}$$

This decision rule is known as *maximum a posteriori (MAP) probability rule* and it may be simplified to

$$\hat{\boldsymbol{x}}_m = \arg\max_{1\leq m\leq M} \frac{P_{x_m} p(\boldsymbol{r}|\boldsymbol{c}_m)}{p(\boldsymbol{r})} = \arg\max_{1\leq m\leq N} P_{x_m} p(\boldsymbol{r}|\boldsymbol{c}_m) \tag{1-2}$$

Since $p(\boldsymbol{r}_m)$ is independent of $\boldsymbol{x}_m$. In the case where $P_{x_m} = 1/M$

$$\hat{\boldsymbol{x}}_m = \arg\max_{1\leq m\leq M} p(\boldsymbol{r}|\boldsymbol{c}_m) \tag{1-3}$$

(**??**) is known as the *Maximum Likelihood (ML) rule*. It is worth noting that what the receiver is essentially doing is dividing an output space $\mathbb{R}^N$ into M decision spaces $D_1, D_2, ..., D_M$ and if $\boldsymbol{r} \in D_m,\ \hat{\boldsymbol{x}_m} = \boldsymbol{x}_m$

For the MAP detector $D_m = \{r \in \mathbb{R}^N : P[\boldsymbol{x}_m|\boldsymbol{r}] > P[\boldsymbol{x}'_m|\boldsymbol{r}],\ \forall\ 1 \leq m \leq M,\ m' \neq M\}$

1.2 Error Probability

From the above discussion, we realize that an error occurs if $r \notin D_m$ when $\boldsymbol{c}_m$ is sent. The symbol error probability of a receiver is given by

$$P_e = \sum_{m=1}^{M} P_{\boldsymbol{x}_m} P[\boldsymbol{r} \in D_m | \boldsymbol{c}_m \text{ sent}] = \sum_{m=1}^{M} P_{\boldsymbol{x}_m} P_{e|\boldsymbol{c}_m} \tag{1-4}$$

Where

$$P_{e|m} = \sum_{1 \le m' \le M,\ m' \ne m} \int_{D_{m'}} p(\boldsymbol{r}|\boldsymbol{c}_m) d\boldsymbol{r}$$

is the error probability when the message $\boldsymbol{x}_m$ is sent and

$$P_e = \sum_{m=1}^{M} P_{\boldsymbol{x}_m} \sum_{1 \le m' \le M,\ m' \ne m} \int_{D_{m'}} p(\boldsymbol{r}|\boldsymbol{c}_m) d\boldsymbol{r} \tag{1-5}$$

(**??**) gives the *symbol error probability*

We define $D_{mm'} = \{p(\boldsymbol{r}|\boldsymbol{c}'_m) > p(\boldsymbol{r}|\boldsymbol{c}_m)\}$ and we see that $D_{m'} \le D_{mm'}$

Again, we define the *pairwise error probability* $P_{m \to m'}$ as

$$P_{m \to m'} = \int_{D_{mm'}} p(\boldsymbol{r}|\boldsymbol{c}_m) d\boldsymbol{r} \tag{1-6}$$

fixing (**??**) into (**??**) we get

$$\begin{aligned} P_e &\le \sum_{m=1}^{M} P_{\boldsymbol{x}_m} \sum_{1 \le m' \le M,\ m' \ne m} \int_{D_{mm'}} p(\boldsymbol{r}|\boldsymbol{c}_m) d\boldsymbol{r} \\ &\le \sum_{m=1}^{M} P_{\boldsymbol{x}_m} \sum_{1 \le m' \le M,\ m' \ne m} P_{m \to m'} \\ &\le \frac{1}{M} \sum_{m=1}^{M} \sum_{1 \le m' \le M,\ m' \ne m} P_{m \to m'} \end{aligned} \tag{1-7}$$

where $P_{x_m} = 1/M$ for the case where the messages are equiprobable

1.3 Symbol Error Probability for Linear Block Codes

Without loss of generality, we assume that the all zero codeword $\mathbf{0}$ From the above discussion, we realize that an error occurs if the receiver decides upon $\boldsymbol{c}_m \ne \mathbf{0}$ as the codeword that was transmitted and this event is defined by the *pairwise error probability* $P_{\mathbf{0} \to c_m}$. The symbol error probability of the linear block code is then

$$P_e = \sum_{c_m \in \mathcal{C},\ c_m \ne \mathbf{0}} P_{\mathbf{0} \to c_m} \tag{1-8}$$

Since codewords with the same weight have the same $P_{\mathbf{0}\to c_m}$ we have

$$P_e = \sum_{i=d_{\min}}^{n} A_i P_2(i) \tag{1-9}$$

Where A_i is the number of codewords if a given weight i and $P_2(i)$ is the PEP of codewords with weight i

1.3.1 Upper bound on Pairwise Error Probability (PEP) for AWGN channel

We attempt to find the PEP for the case of the AWGN channel

$$P_{\boldsymbol{c}_m \to \boldsymbol{c}'_m} \tag{1-10}$$

Let us assume that there are two possible codeword $\boldsymbol{c}_m$ and $\boldsymbol{c}_{m'}$ that can be transmitted over the AWGN channel with equal probability. The decision regions for these codewords are seperated by the perpendicular bisector of the line that connects these two codewords. Furthermore, we assume that $\boldsymbol{c}_m$ is sent and we want to find the $P_{\boldsymbol{c}_m \to \boldsymbol{c}'_m} = P[\boldsymbol{c}_{m'} detected | \boldsymbol{c}_m sent]$ This would mean that there is a point A from $\boldsymbol{c}_m$ which has a distance larger than $d_{mm'}/2$, $d_{mm'} = ||\boldsymbol{c}'_m - \boldsymbol{c}_m||$ since $\boldsymbol{c}_m$ is sent, then $\boldsymbol{n} = \boldsymbol{r} - \boldsymbol{c}_m$ and the scalar projection of $\boldsymbol{r} - \boldsymbol{c}_m$ on $\boldsymbol{c}'_m - \boldsymbol{c}_m$ is $\frac{\boldsymbol{n}\cdot(\boldsymbol{c}'_m - \boldsymbol{c}_m)}{d_{mm'}}$. With the above information, we can write the error probability as

$$\begin{aligned} P_{\boldsymbol{c}_m \to \boldsymbol{c}'_m} &= P\left[\frac{\boldsymbol{n}\cdot(\boldsymbol{c}'_m - \boldsymbol{c}_m)}{d_{mm'}} > \frac{d_{mm'}}{2}\right] \\ &= P\left[\boldsymbol{n}\cdot(\boldsymbol{c}'_m - \boldsymbol{c}_m) > \frac{d^2_{mm'}}{2}\right] \end{aligned} \tag{1-11}$$

Let $d = \boldsymbol{c}'_m - \boldsymbol{c}_m$. Since $\boldsymbol{n}$ is a Gaussian random variable $\boldsymbol{n}\cdot(\boldsymbol{c}'_m - \boldsymbol{c}_m)$ has zero mean. We wish too find the variance of $\boldsymbol{n}\cdot(\boldsymbol{c}'_m - \boldsymbol{c}_m)$ which is $E(x^2) - (E(x))^2 = E(x^2)$

$$\begin{aligned} E(x^2) &= E\left\{\sum_{i=1}^{N} n_i^2 \cdot d_i^2\right\} \\ &= E\left\{\sum_{i=1}^{N} n_i^2\right\}\left\{\sum_{i=1}^{N} d_i^2\right\} \\ &= \frac{N_0}{2} d^2_{mm'} \end{aligned} \tag{1-12}$$

And $P_{\boldsymbol{c}_m \to \boldsymbol{c}'_m} = Q\left(\sqrt{\frac{d^2_{mm'}}{2N_0}}\right)$

If we assume

1.4 Bit Error Probability for Linear Block codes

Let N be the block length of a code with 2^N codewords. From the previous section, we saw that for AWGN channels

$$P_{\mathbf{0}\to c_m} = Q\left(\sqrt{\frac{d^2_E(\boldsymbol{c}_m)}{2N_o}}\right) \tag{1-13}$$

For BPSK modulation $d_E^2(c_m) = 4E_bR_c\mathrm{w}(c_m)$ and we have

$$P_{\mathbf{0}\to c_m} = Q\left(\sqrt{\frac{2E_bR_c\mathrm{w}(\boldsymbol{c}_m)}{N_o}}\right) \tag{1-14}$$

the corresponding bit error probability when $\boldsymbol{c}_m$ is transmitted is given by

$$P_b(\mathbf{0}\to c_m) = \frac{w(\boldsymbol{x}_m)}{N}Q\left(\sqrt{\frac{2E_bR_c\mathrm{w}(\boldsymbol{c}_m)}{N_o}}\right) \tag{1-15}$$

Finally using the union bound, the average bit error probability is bounded by

$$P_b = \frac{1}{N}\sum_{m=1}^{2^N-1} w(\boldsymbol{x}_m)Q\left(\sqrt{\frac{2E_bR_c\mathrm{w}(\boldsymbol{c}_m)}{N_o}}\right) \tag{1-16}$$

Chapter 2

Summary of "Interleavers for Turbo Codes Using Permutation Polynomials Over Integer Rings" by Sun, Takeshita

2.1 Introduction

In this research paper, a new deterministic interleaver (The Quadratic permutation Polynomial Interleaver) is proposed. It is based on permutation polynomials over the ring of integers modulo $\boldsymbol{N}$, $\mathbb{Z}_N$

It has been observed that a subset of error events with input weight $2m$, m being a small positive integer, usually dominates the performance when the frame sizes arent very short. The criteria for selecting good interleavers is based on the effective free distance and for values of $m < 4$ a method for finding these error events is presented.

2.2 Permutation Polynomials over $\mathbb{Z}_N$

The following theorem shows how to identify a permutation polynomial over $\mathbb{Z}_N$, $N = 2^n$

Theorem 1. Let $P(x) = a_0 + a_1x + a_2x^2 + ... + a_mx^m$ be a polynomial with integer coefficients. $P(x)$ is a Permutation Polynomials over $\mathbb{Z}_N$ if and only if

1. a_1 is odd
2. $a_2 + a_4 + a_6 + ...$ is even
3. $a_3 + a_5 + a_7 + ...$ is even

For the more general case where $N = p^n$ we have the following theory

Theorem 2. $P(x)$ is a Permutation Polynomials over $\mathbb{Z}_N$ if and only if it is a permutation polynomial over $\boldsymbol{b}_Z(p)$ and $P'(x) \neq 0 \mod p \; \forall \; \mathbb{Z}_N$

For polynomials of degree two, the criteria for permutation polynomials is simplified by the following corollary

Corollary 1. A degree two polynomial of the form $P(x) = ax + bx^2$ is a permutation polynomial over $\mathbb{Z}_{p^n}$ iff $a \neq 0$ and $b = 0 \mod p$

2.3 Permutation Polynomial-Based Interleavers(PPI)

In general, if a polynomial $P(x)$ is a permutation polynomial over $\mathbb{Z}_N$, then an interleaver based on this permutation polynomial can be defined as

$$\pi_{\mathcal{P}_N} : x \to P(x), \; \forall x$$

A plot of the PPI reveals a period pattern which does not necessarily imply bad performance. This can be better explained considering input weight 2 error events.

input weight 2 error events (2W event)

Definition 1. An input weight 2 error event is defined as an error event with two information bits in error. for a Turbo Code(TC) this corresponds to one input weight 2 error event in each component code

It is widely know that this kind of error event determines the effectrive free distance (d_{eff}) and consequently, the performance of the TC.

A 2W event in the first component code can be represented by the pair $(x, x + t)$ and it is interleaved to $(\pi(x), \pi(x + t))$ in the second component code. $\Delta(x, t) = \pi(x + t) - \pi(x)$ mod N gives the distance between the interleaved points. if t is fixed, it is possible to plot a graph of $\Delta(x, t)$ against x. It is observed that all the points are uniformly located along a few equally separated horizontal lines. We are interested in points close to the line$\Delta(x, t) = 0$. If the coefficents a, b are well chosen, if is possible to have point far from the line $\Delta(x, t) = 0$

2.4 Permutation Polynomial Search

We wish to find the best PPI for a given component code. For a fixed frame size N and fixed polynomial degree, it comes down to choosing the coefficents of the permutation polynomial. Since the focus is on polynomials of degree 2. It comes down to selecting values for a and b .

The minimum distance of a subset of error events of a TC, namely the input weight $2m$ error events(2mW), is used as a criteria to select values for a and b. Though it doesnt always exist when RS convilutional codes are used, the existence of a tail biting trellis is used to simplify analysis. This helps ignore the boundary effects and makes finding errors easy.

2.4.1 Input Weight 2m Error (2mW) Events

A typical 2mW event is shown. It is made up of m 2W events in each component code and they are connected via an interleaver. The ith 2W event in the first component code begins at x_i

and has length t_i. In the second component code, the error event has length s_i. In our analysis we assume that t_i, s_i are all multiples of the cycle length which is the cycle at the output of a conponent encoder when the input is $[1, 0, 0, 0, 0, ...]$

An error pattern is represented by the length $2m$ vector $[t_1, t_2, ..., t_m, s_1, s_2, ..., s_m]$

For the 2mW error event shown in the figure,we have

$$P(x_2) - P(x_1) = s_1 \tag{2-1}$$

$$P(x_3) - P(x_1 + t_1) = s_2 \tag{2-2}$$

$$\begin{aligned} P(x_4) - P(x_2 + t_2) =& s_3 \\ & . \\ & . \\ & . \end{aligned} \tag{2-3}$$

$$P(x_{m-1}) - P(x_{m-3} + t_{m-3}) = s_{m-2} \tag{2-4}$$

$$P(x_m) - P(x_{m-2} + t_{m-2}) = s_{m-1} \tag{2-5}$$

$$P(x_m + t_m) - P(x_{m-1} + t_{m-1}) = s_m \tag{2-6}$$

For such a 2mW event to occur all m equations need to be satisfied. x_m takes values $0, 1, ..., N-1$ and t_i, s_i are multiples of τ

Given an error pattern it is possible to uniquely determine its Hamming distance assuming the error events dont overlap.

it is calculated by the equation below

$$6m + \left(\frac{\sum |t_i| + \sum |s_i|}{\tau} \right) w_o \tag{2-7}$$

2.5 Search for Good Interleavers using d_{eff}

Effective free distance of a TC is determined by its 2W events. For the permutation polynomial, we have

$$\Delta(x, t) = P(x + t) - P(x) = 2btx + bt^2 + at \tag{2-8}$$

The distance to zero can be represented by

$$\begin{aligned} s &= \pm \Delta(x, t) \mod p_N^{o_{c_1}} \\ &= \pm (bt^2 + at) \mod p_N^{o_{c_1}} \end{aligned} \tag{2-9}$$

where $c_1 = 2bt$

Given a, b, τ we define $\mathcal{L}_{a,b,\tau} = \min(|t| + |s|)$ where t, s are multiples of τ. For a given component code, $\mathcal{L}_{a,b,\tau}$ can be used to calculate d_{eff}. In a search for good interleavers, there is a need to limit the range of values for a, b. The following lemma gives info on how to

Lemma 1. For 2W event analysis, if we write $b = b_1 b_0 = b_1 \cdot p_N^{o_b}$, we can assume $b_1 = 1$

Lemma 2. For 2W event analysis,given $b = b_1 \cdot p_N^{o_b}$, we only need to consider a such that $1 \leq a < p_N^{o_b}$

Chapter 3

P_b considering distance between weight $2m$ error event points

For a linear block code with BPSK modulation, the probability of decoding $\boldsymbol{c}$ when the all zero information sequence $\mathbf{0}$ is transmitted is given by

$$P_r\{\mathbf{0} \to \boldsymbol{c}\} = Q\left(\sqrt{2R_c W_H(\boldsymbol{c})\gamma_b}\right) \tag{3-1}$$

where $\gamma_b = \frac{E_b}{N_o}$, $W_H(\boldsymbol{c})$ is the Hamming weight of $\boldsymbol{c}$ and R_c is the rate of the code. The corresponding bit error probability is

$$P_b(\mathbf{0} \to \boldsymbol{c}) = \frac{j}{N} Q\left(\sqrt{2R_c W_H(\boldsymbol{c})\gamma_b}\right) \tag{3-2}$$

Where j denotes the weight of the information sequence and N denotes the lenght of the information sequence.

From[SunTakeshita], we know that for input-weight $2m$ error (2mW) events it is possible to calculate the Hamming weight of the code using the equation below

$$\begin{aligned} W_H(\boldsymbol{c}) =& 6m + \left(\frac{\sum |t_i| + \sum |s_i|}{\tau}\right) w_o \\ =& 3j + \left(\frac{\sum |t_i| + \sum |s_i|}{\tau}\right) w_o \end{aligned} \tag{3-3}$$

Where

$j = 2m$

m is the number of 2mW events present in the codeword,

τ represents the cyclic shift of the component code,

w_o is the weight of the output sequence of the input $1 + D^{\tau}$,

t_i, s_i represent the seperation between the "1" bits in the first and second component codes respectively , $i = \{1, 2, ..., m\}$

Since $t_i,\ s_i$ must all be multiples of τ, we may write $t_i = q_i\tau,\ s_i = r_i\tau$. Where $q_i, r_i \in \mathbb{Z},\ q_i, r_i = \{1, 2, ..., \lfloor \frac{N}{\tau} \rfloor\}$. By a little change of subject we have,

$$W_H(\boldsymbol{c}) = 3j + \left(\sum |q_i| + \sum |r_i| \right) w_o \tag{3-4}$$

Substituting into (**??**) gives

$$\begin{aligned} P_b(\mathbf{0} \to \boldsymbol{c}) =& \frac{j}{N} Q\left(\sqrt{2R_c\gamma_b \left[3j + \left(\sum |q_i| + \sum |r_i| \right) w_o \right]} \right) \\ =& \frac{j}{N} Q\left(\sqrt{\Gamma \left[3j + \left(\sum |q_i| + \sum |r_i| \right) w_o \right]} \right) \\ =& \frac{2m}{N} Q\left(\sqrt{\Gamma \left[6m + \left(\sum |q_i| + \sum |r_i| \right) w_o \right]} \right) \end{aligned} \tag{3-5}$$

where $\Gamma = 2R_c\gamma_b$

Applying the union bound, the average error probability is given by

$$P_b \leq \frac{1}{N} \sum_{m=1}^{N/2} 2m \sum_{i=1}^{l} Q\left(\sqrt{\Gamma \left[6m + \left(\sum |q_i| + \sum |r_i| \right) w_o \right]} \right) \tag{3-6}$$

where

$$l = \sum_{z=\tau}^{N-1} \left\lfloor \frac{z}{\tau} \right\rfloor$$

is the size of the set of possible values for q_i, r_i

Chapter 4

"A Code-Mathed Interleaver Design for Turbo Codes" by Wen Feng, Jinhong Yuan and Branka S. Vucetic

4.1 Abstract

A code-matched interleaver design for turbo codes in which a particular interleaver is constructed to match the code weight distribution is proposed. The design method is based on the code distance spectrum. The low weight paths in the code trellis which give large contributions to the error probability in the signal-to-noise ratio region of interest for practical communication systems are eliminated so that they do not appear in the overall code trellis after interleaving. The proposed interleaver improves the code error performance at moderate to high signal-to-noise ratio and considerably increases the asymptotic slope of the error probability curves.

4.2 Introduction

interleavers are widely used with error control coding for channels which exibit bursty error characteristics and cancatenated codes, especially Turbo codes. Basic role of an interleaver is to construct a long random code and can be used to change the weight distribution of thee Turbo code. A number of Interleavers have been used in combination with turbo codes.

block interleaver Matrix of $N = r \times l$ where r is the number of rows and l is the number of colums. input data is written along row and read out along columns. A type of block interleaver that terminates both component encoders in the same state was also proposed, known as "simile" interleaver

Pseudorandom interleaver variation of the block interleaver where data is written sequentiallly and read out in a pseudo-random manner.The S-random interleaver is an improved version of this interleaver and can "spread" low-weight input patterns to generate higher weight codewords.

convolutional interleaver data is multiplexed into and out of a fixed number of shift registers

prime interleaver it is also based on block interleaving and can make turbo code generate codewords with good hamming distance