Fountain Code

MMRFIC Technology Pvt. Ltd., Bangalore

November 19, 2019

Ver.	Author(s)	Date	Comments
1.0	Sudhanshu Sharma	19/11/2019	created

Table 1: Version History

Contents

1 Introduction

This document contains the implementation of fountain codes. It explains encoding, decoding, channel information, modulation and demodulation for a given set of information.

2 Input Data

Input is 63 x 1 matrix which is in binary format (1's, 0's). It is 63 because of the columns in code matrix.

3 Encoder

Here we are using CodeMatrix for encoding and decoding, information about code matrix is presented at both the transmitter and receiver end. Size of Code matrix for this particular experiment was given as 441 x 21 (In octal format) which was then converted into 441 x 63 (Binary Format) as we are using Galois Field 2 which has only 0 and 1.Each row of code matrix will have only 3 non- zero elements.
Encoding is done by multiplying the input with the code matrix. Hence the size of code matrix is 441 x 63 so we have chosen input size as 63 x 1 which yield the Encoded output Bits as 441 x 1.

$$\begin{bmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,63} \\ p_{2,1} & p_{2,2} & \cdots & p_{2,63} \\ \vdots & \vdots & \ddots & \vdots \\ p_{441,1} & p_{441,2} & \cdots & p_{441,63} \end{bmatrix} \tag{1}$$

4 Modulation

Here we are doing BPSK in which the bit 0 is mapped to bit -1 and bit 1 is mapped to 1. After the modulation we have to pass the symbols to the channel

5 Channel Error

We are using two different channels -

1. BSC (Binary Symmetric Channel) : In this model, a transmitter wishes to send a bit (a zero or a one), and the receiver receives a bit. It is assumed that the bit is usually transmitted correctly, but that it will be "flipped" with a small probability (the "crossover probability").
   So here it's like - for block size 180 we had Error Probability of 1e-6, 1e-5, 1e-4, 1e-3, 1e-2. And then for block size 220, again Error Probability of 1e-6, 1e-5, 1e-4, 1e-3, 1e-2 and so on.

2. Additive White Gaussian Noise : Additive white Gaussian noise (AWGN) is a basic noise model used in Information theory to mimic the effect of many random processes that occur in nature.

Calculation of Sigma was done for all the SNR values from 1 to 10. Then added to the modulated signal.

$$\sigma = \sqrt{\frac{1}{10^{\frac{SNR(ErrorPro)}{10}}}x}$$

It was similar to BSC in the ways of implementation like -
We did AWGN channel Model with SNR of 1 to 10 .And with the block size of 180 to 300 in- steps of 40.

6 Demodulation

Demodulation is done using LLR (Log -Likelihood Ratio)

$$\rho = \log(\frac{P(y|b(w)=0,r)}{P(y|b(w)=1,r)})$$

$$= \log(\frac{\sum_{x0} P(y|w,r)}{\sum_{x1} P(y|w,r)})$$

$$P(y|w,r) = \frac{1}{\sqrt{2\pi\sigma^2}}.\exp(\frac{-(y-rw^2)}{2\sigma^2})$$

$$\rho = \frac{2ry}{\sigma^2}$$

Hence we can clearly see that it depends on the "ry" value if ry is positive the demodulated output is going to be positive and if its negative that the result is negative. Which is done using

$$Demoded = (1 * (ChannelOutput > 0));$$

7 Decoder

Decoding is done using Gaussian Elimination in which Demodulated bits are augmented with Code Matrix

$$\begin{bmatrix} CodeMatrix_{1,1} & CodeMatrix_{1,2} & \dots & CodeMatrix_{1,63}| & DemodulatedBits_{1,64} \\ CodeMatrix_{2,1} & CodeMatrix_{2,2} & \dots & CodeMatrix_{2,63}| & DemodulatedBits_{1,64} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ CodeMatrix_{441,1} & CodeMatrix_{441,2} & \dots & CodeMatrix_{441,63}| & DemodulatedBits_{441,64} \end{bmatrix}$$

Out of available 441 x 64 random rows are selected for different block size. As we only need K+E bits to do Gaussian Elimination where K - size of input and E - Extra bits to perform Gaussian Elimination

Gaussian Elimination Algorithm
M - Rows , N - Columns

1. Search for 1's column wise.
    - IF no 1s' are found, report - "lack of 1's to perform Gaussian Elimination". And give output as all zeros.
    - ELSE make the First encountered 1 as First row of the matrix
2. Then XOR with next 1's found on that same column to make it zero.
3. Performing this for all N will yield square matrix with all diagonal as 1's and all the element below diagonal as zeros.And are the non redundant element below the square (Upper triangular matrix)
4. By back substituting from nth row and keep solving the equation by equating it to the nth element.

8 Modification/cases

Initially, We were using all 441 rows to solve for Gaussian Elimination and then we start using random rows of particular block size to solve for Decoded output bits.
Then later we sorted the received symbols (after AWGN channel) in increasing order of the signal amplitude and picked the top "BlockSize" rows from there. Then used the BlockSize to decode the payload. And which result in better BER vs SNR curve.

9 Results

Figure 1 Result of BSC Channel (Error Probability vs BER) Figure 2 Result of BSC Channel (Error Probability vs BER)

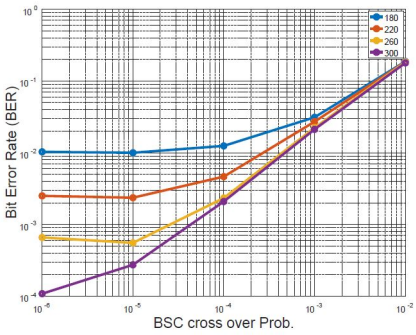

Figure 1: Result of BSC Channel (Error Probability vs BER)

10 Overview

$$Input \rightarrow Encoded \rightarrow Modulation \rightarrow Channel \rightarrow Demodulation \rightarrow Decoded \rightarrow Output$$

$$[CodeMatrix] * [Input] = [EncodedBits]$$

$$[EncodedBits] \rightarrow [ModulatedBits]$$

$$[ModulatedBits] \rightarrow [ChannelError]$$

$$[ChannelOutput] \rightarrow [DemodulatedBits]$$

$$[CodeMatrix|DemodulatedBits] \rightarrow [DecodedBits]$$

Compare Decoded Bits with Input Bits, If matches no error if not there is error and then BER is calculated

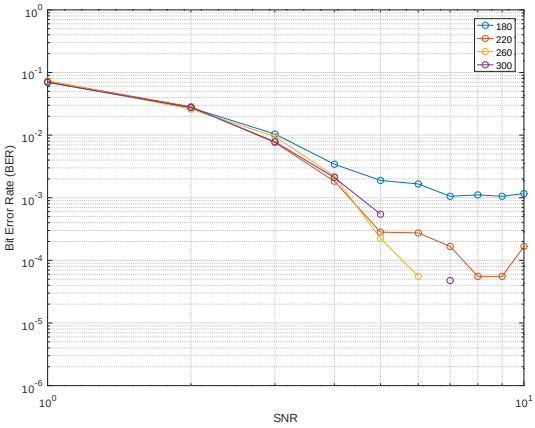

Figure 2: Result of AWGN Channel (SNR vs BER)