TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

PULCHOWK CAMPUS

THESIS NO: 070/MSI/617

Complexity Reduction of Reliability Based Decoding of Linear Block Codes

by

Sulav Adhikari

A THESIS

SUBMITTED TO THE DEPARTMENT OF COMPUTER AND ELECTRONICS ENGINEERING IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE DEGREE OF MASTER OF SCIENCE IN INFORMATION AND COMMUNICATION ENGINEERING

DEPARTMENT OF ElECTRONICS AND COMPUTER ENGINEERING

LALITPUR, NEPAL

FEBRUARY, 2016

Complexity Reduction of Reliability Based Decoding of Linear Block Codes

by

Sulav Adhikari

070/MSI/617

A thesis submitted in partial fulfilment of the requirements for the degree of

Master of Science in Information and Communication Engineering

under the supervision of

Prof. Dr. Subarna Shakya

Department of Electronics and Computer Engineering,

Pulchowk Campus, Institute of Engineering

Tribhuvan University

Lalitpur, Nepal

February, 2016

COPYRIGHT

TRIBHUVAN UNIVERSITY
INSTITUTE OF ENGINEERING
PULCHOWK CAMPUS
DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING

The undersigned certify that they have read, and recommended to the Institute of Engineering for acceptance, a thesis report entitled "Complexity Reduction of Reliability Based Decoding of Linear Block Codes", submitted by Sulav Adhikari in partial fulfillment of the requirement for the degree of "Master of Science in Information and Communication Engineering".

Defense Date: 7th February, 2016

Examination Committee

Dr. Dibakar Raj Pant ……..……
Chairperson,
Head of Department

Dr. Subarna Shakya ……..……
Professor, Supervisor
Member

Dr. Surendra Shrestha ……..……
Program coordinator

Dr. Gajendra Sharma
External Examiner ……..……
Associate Professor
Department of Computer Science and Engineering
Kathmandu University
Dhulikhel, Kavre

DEPARTMENTAL ACCEPTANCE

The thesis entitled “Complexity Reduction of Reliability Based Decoding of Linear Block Codes” submitted by Sulav Adhikari in partial fulfilment of the requirement for the award of the degree of “Master of Science in Information and Communication Engineering” has been accepted as a bonafide record of work independently carried out by him in the department.

_____

Dr. Dibakar Raj Pant
Head of the Department
Department of Electronics and Computer Engineering
Pulchowk Campus, Tribhuvan University, Nepal

ACKNOWLEDGEMENT

I would like to express sincere gratitude to Professor Dr. Subarna Shakya for his constant support, enthusiasm and keen interest in this work. His advices on technical matters are invaluable, and his guidance is very critical for the successful of my thesis.

I have taken effort in this thesis. However, it would not have possible without the kind support and help of many individuals and Institute of Engineering Pulchowk Campus. I would like to extend my sincere thanks to all of them.

I would like to thank Professor Dr. Shashidar Ram Joshi, not only for his comments but also for his enjoyable and excellent lectures which have brought me a strong background in the field of information theory and coding.

I am very much thankful to the Department of Electronics and Computer Engineering, Institute of Engineering for accepting my thesis on "Complexity Reduction of Reliability Based Decoding of Linear Block Codes".

Furthermore, I would like to acknowledge with much appreciation the crucial role of our Master's degree program coordinator, Dr. Surendra Shrestha.

Finally, I'd like to express my heartfelt thanks to my family and my friends who have always encouraged and supported me.

ABSTRACT

The Reliability Based Algorithm for linear block codes is investigated. The decoding complexity is increased when the length of information bits of linear block codes is increased. A Simplified statistical approach to evaluate the error performance bound of Reliability Based Algorithm of Linear Block Codes is investigated. First, one novel statistic is proposed which depicts the number of error contain in the ordered received noisy codewords. Also, another new statistic is proposed which compare the hamming distance between the permuted received word and the reprocessing codeword only on the parity check section. Then, the Probability Density Function (PDF) and the Cumulative Distribution Function (CDF) for each statistics are derived. Also, the new Probabilistic Threshold Test in combination with basic order-I reprocessing is proposed which compare the decision statistic evaluated only on the parity check section for all reprocessing codeword corresponding to TEP of any permissible order. Finally, with proposed approach, reduction of the number of Test Error Patterns (TEPs) is obtained to calculate the required codeword which reduces the decoding complexity.

Keywords: Reliability Based Algorithm, Linear Block codes, Probabilistic Threshold Test

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

BPSK	Binary Phase Shift Keying
CDF	Cumulative Distribution Function
GMD	Generalized Minimum Distance
GPTT	Generalized Probabilistic Threshold Test
MLD	Maximum Likelihood Detection
MRB	Most Reliable Basis
RBA	Reliability Based Algorithm
PDF	Probability Density Function
PTT	Probabilistic Threshold Test
TEP	Test Error Patterns

CHAPTER ONE: INTRODUCTION

1.1 Background

Coding theory is deals with reliability of communication over noisy channel. The main purpose of coding theory is to improve the reliability of digital communication by detecting and correcting the error in the received bit during the digital transmission. In the digital communication, transmission is not 100% reliable in practice because of the presence of noise which distorts the information bit. There are two techniques which is used to reduce the error in digital transmission 1) Automatic Repeat request (ARQ) 2) Forward error correction (FEC).

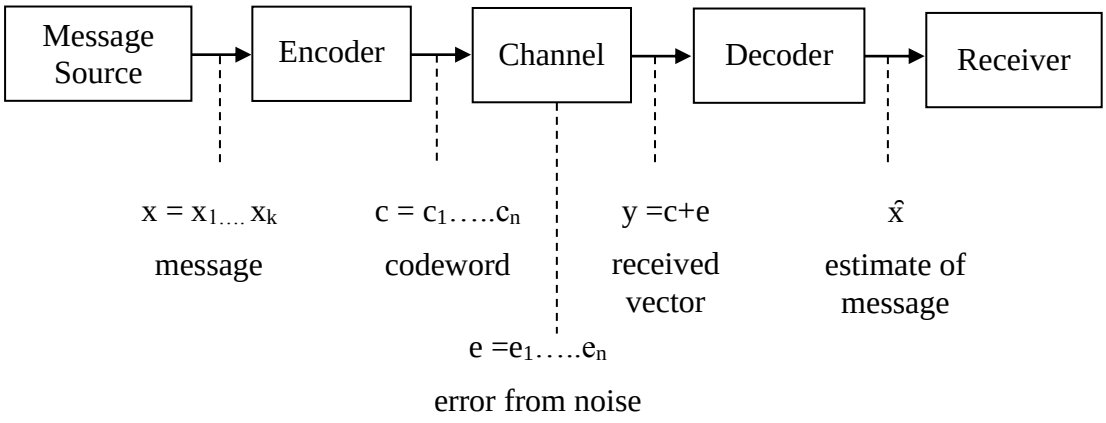

Figure 1.1: Communication channel

ARQ is the process which detects the presence of error in the received data and if errors are found then receiver notifies the presence of error to the transmitter. Finally, transmitter resends the data until they are correctly received. In forward error correction, receiver not only detects the error but also corrects the error, so there is no need of retransmission of data. Sometimes retransmission of any data is impossible or not feasible so that forward error correction is better solution. A special type of code is used for forward error correction which is called as error correcting codes or error control codes. The techniques in error correcting codes is add redundancy to original message in such a way that receiver have ability to detect the error and correct it [1].

A communication channel is illustrated in figure 1.1. x represents message at the source. If there is no modification in message x and transmitted over a communication channel, the message is distorted due to the presence of noise and receiver cannot

recover the original message. The key concept of channel encoding is adding redundancy to the message to be sent. Channel encoder add the redundancy bit in original message bit and form the codeword **c**. Noise is in the form of error vector **e** and distorts the message and produce the output y be decoded where errors are remove by help of redundancy bit [1].

Error correcting codes can be divided into two classes on the basis of redundancy: Block code and Convolutional code. Block coding encodes and decodes data on a block by block basis, treating each block of information bits independently from others. Encoding and decoding operation of convolutional code depend not only on the current data but also on the previous data.

The use of redundancy act as overhead or cost in terms of channel bandwidth, or transmission power in the digital transmission. Coding rate R give the quantitative measure of redundancy and is defined as the ratio of message length to the codeword length.

$$R=K/N \quad \text{..... (1.1)}$$

The capability of error correction is increased when the redundancy is increased but coding rate is decreased. A sequence of digital symbols is obtained after the error control encoding. Digital modulation is needed to convert symbols into analog symbols for transmit over transmission channel. Binary phase shift keying (BPSK) is one of the modulation technique. BPSK assigns to the carrier 180 phase shifts when the bit is 0 and π.

Channel coding is used in digital communication system to protect the digital information from noise and interference. Channel coding is mostly accomplished by selectively introducing redundant bits into transmitted information stream. These additional bits will allow detection and correction of bit errors in the received data stream and provide more reliable information transmission [3].

1.2 Linear Block Code

The output of an information source consist of binary digits “0” or “1”. In block code, the sequence of binary digits segmented into message blocks of fixed length, say k.

Each message block is mapped into codeword by using redundancy bit. For k bits, 2^k are the possible codewords. Linear block is represented by (n, k) where n is the length of codeword and k is the length of message. Codeword consists of n-k bits are the parity check bits or redundancy check bits.

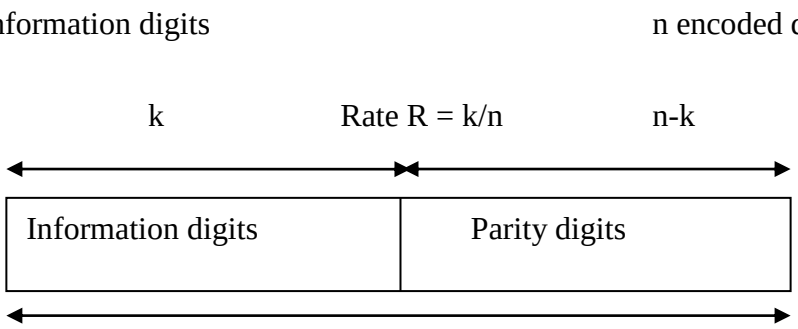

Figure 1.2: Codeword of linear block code

The linear block code is possess a systematic structure of codeword which is called as systematic linear block code which is shown in figure 1.2. Codeword is divided into two part. The first parts of codeword is message and the second part is redundant checking part. The message part consists of k information digits and the redundant checking part consist of n-k parity check digits which are linear sum of information digits.

Linear block code can be describes by generator matrix and parity check matrix. Generator matrix is used in the encoding operation at the transmitter.

Let us consider the Generator matrix is denoted by

$$\boldsymbol{G} = \begin{bmatrix} g_0 \\ g_1 \\ \cdot \\ \cdot \\ \cdot \\ g_{k-1} \end{bmatrix} = \begin{bmatrix} g_{00} & g_{10} & g_{02} & \cdot\ \cdot\ \cdot & g_{0,n-1} \\ g_{21} & g_{11} & g_{12} & \cdot\ \cdot\ \cdot & g_{1,n-1} \\ \cdot & \cdot & \cdot & \cdot\ \cdot\ \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot\ \cdot\ \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot\ \cdot\ \cdot & \cdot \\ g_{k-1,0} & g_{k-1,1} & g_{k-1,2} & \cdot\ \cdot\ \cdot & g_{k-1.n-1} \end{bmatrix}.$$

Where $\boldsymbol{g_i}$ = ($g_0, g_1, \ldots, g_{k-1}$) for $0 \leq i \leq k$ and $\boldsymbol{u} = (u_0, u_1, \ldots u_{k-1})$ be the message to be encoded.

$\boldsymbol{v} = \boldsymbol{u}.\boldsymbol{G}$ ……..... (1.2)

$$= (u_0, u_1, \dots, u_{k-1}) \begin{bmatrix} g_0 \\ g_1 \\ \vdots \\ g_{k-1} \end{bmatrix} \quad \text{..... (1.3)}$$

$$= u_0 g_0 + u_1 g_1 + \cdots + u_{k-1} g_{k-1} \quad \text{..... (1.4)}$$

The rows of $\boldsymbol{G}$ generate the (n, k) linear code $\boldsymbol{c}$ so that matrix $\boldsymbol{G}$ is called the generator matrix.

A linear systematic (n, k) code is completely specified by a $k \times n$ matrix of $\boldsymbol{G}$ of the following form

$$G = \begin{bmatrix} g_0 \\ g_1 \\ \vdots \\ g_{k-1} \end{bmatrix} = \begin{bmatrix} p_{00} & p_{01} & \cdots & p_{0,n-k-1} & 1 & 0 & 0 & \cdots & 0 \\ p_{10} & p_{11} & \cdots & p_{1,n-k-1} & 0 & 1 & 0 & \cdots & 0 \\ \cdot & \cdot & \cdots & \cdot & 0 & 0 & 1 & \cdots & 0 \\ \cdot & \cdot & \cdots & \cdot & \cdot & \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot & \cdot & \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot & \cdot & \cdot & \cdot & \cdots & \cdot \\ p_{k-1,0} & p_{k-1,1} & \cdots & p_{k-1,n-k-1} & 0 & 0 & 0 & \cdots & 1 \end{bmatrix}$$

Where p_{ij} = 0 or 1 and let I_k denoted the $k \times k$ identity matrix. Then$G = [p\ I_k]$. Also for any $k \times n$ matrix G with k linearly independent rows there exist a $(n-k) \times n$ matrix. The matrix $\boldsymbol{H}$ is called parity check matrix of the code. The code generated by$\boldsymbol{G}$ is codeword if and only if $\boldsymbol{v}.\boldsymbol{H}^{\boldsymbol{T}} = \boldsymbol{0}$ **.**

$$\boldsymbol{H} = [\boldsymbol{I}_{\boldsymbol{n-k}}\, \boldsymbol{p}^{\boldsymbol{T}}] \quad \text{..... (1.4)}$$

$$\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & p_{00} & p_{10} & \cdots & p_{k-1,0} \\ 0 & 1 & 0 & \cdots & 0 & p_{01} & p_{11} & \cdots & p_{k-1,1} \\ 0 & 0 & 1 & \cdots & 0 & p_{02} & p_{12} & \cdots & p_{k-1,2} \\ \cdot & \cdot & \cdot & \cdots & \cdot & \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdot & \cdots & \cdot & \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdot & \cdots & \cdot & \cdot & \cdot & \cdots & \cdot \\ 0 & 0 & 0 & \cdots & 1 & p_{0,n-k-1} & p_{1,n-k-1} & \cdots & p_{k-1,n-k-1} \end{bmatrix}$$

1.3 Hard and Soft Decision Decoding

The received codeword is compared with all codewords in hard decision decoding and the codeword which gives the minimum hamming distance is selected. The hamming (7 4 3) code have three redundancy bit which is add with four information bits to make a codeword. There are 16 possible codewords. The transmitted codeword is modulated by BPSK signal. Modulated signal is transmitted through channel. Due to the presence of noise, let the received symbol is (-0.8 -1.1 +0. 3 +0.5 +1.8 +0.2 -0.1 -0.4). Hard decision decoder decodes the symbol into bit by taking a reference (Threshold detector). The positive symbol taken as 1 and the negative symbol taken as 0. The output codeword after the hard decision decoding is "00111100". The codeword is declared as required codeword which gives the minimum hamming distance [3].

Soft decision decoding can calculate the Euclidean distance between the received symbols and all possible codewords in the form of modulated symbols in Euclidean space and calculate the each Euclidean distance. The codeword is selected which gives the minimum Euclidean distance. Thus the soft decision decoding improves the decision making process by supplying additional reliability information (calculated Euclidean distance). Soft decision decoder uses all of the information in the process of decision whereas the hard decision decoder does not fully utilize the information available in the received signal [3].

1.4 Problem Definition

The Reliability based algorithm gives an excellent trade-off between the performance and complexity for the decoding of linear block codes of length even greater than 128. However, when it comes to achieving the Maximum Likelihood performance, the number of TEP to be tested is increase exponentially with the increase in length of information bits. Thus, if the length of information bits is increased, it is clear that number of TEP is quite large for practical implementation and increased in complexity of soft decision decoding of linear block code. According the Reliability Based Decoding, for k/n≥0.5, $d_{min}/4$ reprocessing order is sufficient to obtain the required codeword, but at each reprocessing order, the number of TEP to be checked is increased exponentially.

1.5 Research Questions

- What happens if the length of the information bits is increased in the soft decision decoding of linear block codes based on ordered statistics?
- How to reduce the complexity of Reliability Based Decoding of linear block code at each reprocessing order.

1.6 Objectives

The main objective of this work is reducing the decoding complexity of soft decision decoding of linear systematic block code based on most reliable basis.

CHAPTER TWO: LITERATURE REVIEW

Channel coding is an important study which attempts to minimize data loss, due to errors introduced in transmission due to imperfect channels, by adding redundancy to the data during data transmission. Among the two important class of error control coding: block and convolutional, linear block codes come under a block category with a linearity property which is mainly used for Forward Error Correction (FEC) [4]. For a linear block code defined with $C(n, k, d_{min})$: n a codeword length, k information word length and d_{min} the minimum hamming distance to find the Maximum Likelihood (ML) best codeword at minimum euclidean distance, exhaustive decoding comparing the noisy codeword against all codeword (2^k) is obviously impossible for non-trivial codes. Many decoding algorithms have been proposed including highly efficient viterbi decoding [4]. Although all the linear block codes possess the trellis structure which is the backbone of viterbi decoding, the number of states becomes too large to practically implement for long codes. Forney presents a new iterative a new distance measure which enable likelihood information to be used in algebraic minimum distance decoding techniques [5]. Forney uses an algebraic decoder to generate a list of codeword candidates is determined from the reliability measures of the symbols within each received block. Each candidate codeword is test and most likely candidate is chosen as decoded codeword. Chase introduced an algorithm where a fixed number of the error patterns are systematically searched [6]. Tested position can be chosen according to the reliabilities less than a predetermined threshold.

The error performance depends on the choice of threshold and maximum number of computation depends on choice of threshold as well as signal-to-noise ratio (SNR) for the given set of tested position. The complexity of this algorithm increased exponentially with dimension of code.

There has been interest in algorithm which perform soft decision decoding of linear block code on the basis of "most reliable basis". These decoding algorithms are called as MRIP-reprocessing decoding algorithm. GMD and Chase type algorithm take only a partial ordering of the reliability values of the received symbol for identification of the LRPs of received sequence. In the decoding process, MRIPs required complete ordering of the received sequence as well as determination of K MRIPs. The first k

bits are obtained according to the MRB and permuting the column of generator matrix based on this ordering. Soft decision decoding for binary linear code is the method to obtain required error performance progressively in a number of stages.

Soft decision decoding based on ordered statistics have long been investigated extensively [7]-[22], since *Fossorier and Lin,* in their original contribution presented a novel scheme for soft decision decoding of linear block codes based on ordered statistic of the received noisy samples [8]. An efficient near optimum ML decoding algorithm for a binary linear block code has been proposed. Algorithm was basically implement into two stages, A) determining the Most Reliable Independent (MRI) bits from Most Reliable Basis (MRB) of the code and B) order $I-$ reprocessing on MRI using most likely Test Error Patterns (TEPs). Order $I-$ reprocessing is designed to improve the hard decision decoded codeword progressively until either practically optimum or a desired error performance is achieved.

The approach of ML resource test based on the cost function calculated from the soft valued samples of the permuted received sequence is introduced as a stopping criterion after each stage $j, 0 \leq j \leq I$ of order $-I$ reprocessing which indeed proved excellent in reducing the average number of computation. The major weak point of this algorithm is no instant stopping criterion can be achieved between that order$-j$ and order $-(j-1)$ reprocessing stages. Combination $\binom{k}{l}$ increases for increasing i which means most of the candidate codewords are processed at phase$-I$ of order$-I$ reprocessing. This is the reason that, the stopping criteria presented in [8] for each stage$-I$ still proves inefficient in reducing the average number of computation to an optimum value. Number of techniques with minimized complexities is TEP list optimization and reprocessing strategy optimization. In the context OSD order$-I$ reprocessing the arrangement of TEPs in the list is quite important. The probability of finding the correct codeword corresponding to any TEP clearly depends on the *priori* likelihood could be hamming weight [7]. Improved a *priori* likelihood function like punctured correlation discrepancy [12], a *priori* weight based on the mean bit reliability [22] and the error probability in TEP position [11] proving to be efficient in reducing the worst case computation to a certain level. A method based on some redundant information provided to the decoder which is used to reduce the TEP list size in addition to OSD [14]. The upper union bound of maximum likelihood decoding for linear block code is

presented [23]. Order statistics decoding also used in decoding of LDPC and convolution codes where a reliability measure of received symbol has been used to reduce search space and find the most likely codewords [24].

A next approach which is based on the probabilistic Threshold Test (PTT) is one of the very few which actually proved to reduce the average number of computation in an order$-I$ reprocessing [11]. It is assumed that under sufficiently high Signal-to-Noise Ratio (SNR), the order-0 reprocessing codeword gives the correct codeword with high probability, thus it suggests to compare the Hamming distance between the hard decided permuted received vector and order-0 reprocessing codeword with certain threshold to decide to go for reprocessing. The reduction in computational cost using this approach is quite impressive but not sufficient.

The first contribution of the thesis present some new statistics of the ordered vector components in addition to [7], [8]. Simplified expression for the Probability Density Function (pdf) and Cumulative Distribution Function (cdf) for each statistics is derived. The properties of these new statistics is incorporated to derive the simplified error performance bound for OSD. The next contribution is extension to the concept of PTT [11] named as Generalized Probabilistic Threshold Test (GPTT) in combination with basic order$-I$ reprocessing is proposed which has the property of instant stopping criterion.

CHAPTER THREE: METHODOLOGY

3.1 Reliability Based Algorithm

Reliability based Algorithm is a decoding process which decodes the received symbol according to their reliability. A symbols which have more probability is placed to the first and symbols which have least probability is placed to the last. Hamming code (7 4 3) where n=7, k=4 and d_{min}=3, is used for error control over the channel under the binary phase shift keying (BPSK) signaling. A bipolar version of codeword is used for transmission. The step of Reliability based algorithm for linear block code are

Step 1: Reordering the received symbols according to their decreasing reliability.

The received vector **r** composed of soft values is

$$\mathbf{r}= \begin{bmatrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 \end{bmatrix}$$

Permute **r** into **r**' ordered for decreasing reliability

$$\mathbf{r}'= \begin{bmatrix} x_a & x_b & x_c & x_d & x_e & x_f & x_g \end{bmatrix}$$

Where $x_i \in \mathbf{r}$ and $[x_a] > [x_b] > [x_c] > [x_d] > [x_e] > [x_f] > [x_g]$

$i = a, b, c, d, e, f, g$

Step 2: Obtain a systematic matrix corresponding to a permuted vector.

A systematic generating matrix $\mathbf{G}^*_{\mathbf{kn}}$ corresponding to permuted vector **r**' is obtained.

$\mathbf{G_{kn}} \longrightarrow$ Column permutation reduce to systematic form $\longrightarrow \mathbf{G}^*_{\mathbf{kn}}$

Step 3: Hard decoding of the permuted received word

The permuted received word **r**' is given by

$$\mathbf{r}' = \begin{bmatrix} x_a & x_b & x_c & x_d & x_e & x_f & x_g \end{bmatrix}$$

The hard decoding of the permuted received word is denoted as $\mathbf{R'}$

$$\mathbf{R'} = \begin{bmatrix} X_a & X_b & X_c & X_d & X_e & X_f & X_g \end{bmatrix}$$

Taking first k bits of $\mathbf{R'}$ in account to get an input vector $\mathbf{V^*}$ as.

$$\mathbf{V^*} = \begin{bmatrix} X_a & X_b & X_c & X_d \end{bmatrix}$$

Step 4: Calculate the first best codeword by encoding first k bits of hard decoding of permuted vector and systematic generating matrix corresponding to permuted vector

Now, we encode $\mathbf{V^*}$ by $\mathbf{G^*_{kn}}$ to obtain the first best codeword $\mathbf{C^*}$ as

$$C^* = \mathbf{V^*}\mathbf{G^*_{kn}} = \begin{bmatrix} C_1^* & C_2^* & C_3^* & C_4^* & C_5^* & C_6^* & C_7^* \end{bmatrix} \quad \text{(3.1)}$$

Step 5: Consider all the Test Error Patterns (TEPs) and calculate the each distance and update the codeword which gives minimum distance.

Next, compute the euclidean distance D of current best codeword $\mathbf{C^*}$ from permuted received word.

First choose the order O_r of the algorithm and consider all Test Error Patterns (TEP) of length k and weight $< O_r$. For each considered TEP

- Sum to $\mathbf{V^*}$
- encode by $\mathbf{G^*_{kn}}$
- compute distance D

Where distance D compare with current best distance D^*, update new best codeword $\mathbf{C^*}$ and new best distance D^*. Finally all TEPs have been considered, best codeword $\mathbf{C^*}$ is accepted.

The main objective of the algorithm is to find a codeword $\mathbf{C}$ which is nearest to the received vector **r** in a Euclidean space. This is clearly equivalent to find a codeword

$\mathbf{C}^*$ (Permuted version) which is nearest from the permuted vector $\mathbf{r}'$ in Euclidean space. Now, the Euclidean distance between $\mathbf{C}^*$ and $\mathbf{r}'$ is given by,

$$D = \sum_{i=1}^{n} (c_i^* - x_i)^2 \quad \text{………………………………………………………….. (3.2)}$$

$$= \sum_{i=1}^{n} (c_i^{*2} + x_i^2) - 2\sum_{i=1}^{n} c_i^* x_i \quad \text{……………………………………………...…. (3.3)}$$

The distance D can be minimized by using from the equation 3.3 i.e.

$$mass = \sum_{i=1}^{n} c_i^* x_i \quad \text{……………………..………………………….……..……… (3.4)}$$

The focus is on minimizing the distance D for the given received vector $\mathbf{r}'$ by maximizing mass $\sum_{i=1}^{n} c_i^* x_i$.

3.2 Generalized Probabilistic Threshold Test (GPTT)

The ordered statistics decoding gives an excellent trade-off between the performance and complexity for the decoding of linear block codes. For near optimum decoding performance extended (128, 64, 22) BCH code still requires a tremendous number of TEP reprocessing even for I=4 which is less than $d_{min}/4$. The stopping criterion is necessary which is able to limit the number of computation to a small number. Generalized Probabilistic Threshold Test which reduces the number of computation is presented. Given a permuted received sequence r' and its binary version R'. Redefine a test statistic E_T which measure the number of differences in the tail (parity check section) of permuted binary received vector R' when compared with the tail of the reprocessing c^*.

OSD decoding considered the first k Most Reliable Information (MRI) bits of R' to form a candidate information vector V^*. The candidate information is processed by adding a Test Error Pattern (TEP) and encoded by the matrix G_{kn}^* to obtain a reprocessing codeword c^*.

Reformulate the scenario with binary hypothesis testing, we obtain

Under Null Hypothesis H_0, $E_T=E_3$ ……..…. (3.5)

and under Alternate Hypothesis H_1, $E_T=E_2$ …… (3.6)

Consider a test where the test statistic E_T is compared against a predefined threshold T in order to decide if c^*a true permuted codeword is or not. Based on the test if $E_T > T$, decision in favor of H_0 is done, thus assumed a true permuted codeword is not yet found and the reprocessing algorithm goes on with the update of TEP. Otherwise, if $E_T < T$, decision in favor of H_1 is done, thus assumed a true permuted codeword is found and the reprocessing algorithm stops. It is clear that a predefined threshold measures the extent of TEP reduction.

The decision statistic E_T is compared which is evaluated only on the parity check section for all reprocessing codeword corresponding to TEP of any permissible order for which is called Generalized Probabilistic Threshold Test. Steps for OSD with $I-$order reprocessing based on GPTT defined stopping criterion is summarized here.

(Starting from the permuted received vector and permuted generator matrix)

1. Input
   a. Permuted received vector r'
   b. Permuted generator matrix G'_{kn}
2. Initialization
   a. Initial TEP $\overline{\boldsymbol{p}} = \mathbf{0}$
   b. Best minimum Euclidean distance D_E^{min} to a all zero codeword.
   c. Best codeword c_b (corresponding to D_E^{min} to all zero codeword)
   d. Set order I of the reprocessing algorithm
3. Hard decode the permuted received sequence r' to form a permuted binary received vector $\bar{y}'$
4. Take first k information bits of $\bar{y}'$ to form candidate information vector V^* and start reprocessing.
5. Add a TEP to a candidate information vector V^* to form a reprocessing information vector $V' = V^* + \overline{\boldsymbol{p}}$

6. Encode the reprocessing information vector by a permuted generator matrix G^*_{kn} generating a reprocessing codeword c^*.
7. Evaluate test parameters
    a. Calculate the E_T based on differences in the tail of c^* and $\bar{y}'$
    b. Note the Euclidean distance D_E between the permuted received sequence and the soft (BPSK modulated) valued version of the reprocessing codeword c^*
8. If $D_E < D_E^{min}$, update D_E^{min} by D_E and c_b by c^*.
9. If $E_T < T$, update c_b byc^*,stop reprocessing and go step 11, else go to step-10.
10. Update TEP. If maximum number of TEP test supported by maximum order I is reached, stop reprocessing and go to step-11, else go to step-4.
11. Declare the best codeword c_b as the true permuted codeword.

CHAPTER FOUR: RELATED THEORIES

4.1 TEP Reduction Analysis for Hamming (7, 4, 3) code

TEP reduction analysis for reducing the complexity of Reliability based decoding of linear systematic block code based on ordered statistics. Reliability based algorithm doesn't use the reliability information of the remaining parity check symbols of the permuted received vector. TEP reduction algorithm is based on the reliability information obtained from the tail of the permuted received sequence. In the TEP reduction analysis, the reliability information of the last n-k soft values to confine the minimum Euclidean distance in a very small set of TEP is utilized.

Step 1: Count the number bits different from last n-k bits of hard decoded received word $R^{'}$ and last n-k bits of first best codeword C^{*}. Let the count is denoted by N.

Step 2: Choose TEPs for testing based on the following observation.

a. If N = 0 or 1, No need to check new best distance in other TEPs
b. If N = 2 , Consider only single order TEPs
c. If N = 3, Consider first and second order TEPs only

Step 3: For each considered TEP sum to V^{*}

Step 4: Each V^{*} is encoded by $G^{*}_{\text{kn to}}$ find the codeword

Step 5: Compute the Euclidean distance for each TEPs

4.2 Explanation of the TEP reduction algorithm

Observation 1:

$X^{'}_{i} == C^{*}_{i,}$ the product term $x_i c_i^*$ is always positive.

The arbitrary product term $x_i c_i^*$ from the expression of the "mass" is taken. For BPSK modulated symbols, it is obvious to say that,

$X_i' = 1 => x_i$ is positive

$C_i^* = 1 => c_i$ is positive

and the product $x_i c_i^*$ is positive.

Similarly, for the reverse case.

$X_i' = 0 => x_i$ is Negative

$C_i^* = 0 => c_i$ is Negative

Thus, the product $x_i c_i^*$ is positive.

$R' == C^*$, which shows mass is the sum of absolute values of the element of permuted received vector.

Observation 2:

V^* is equal to the first k bits of $R'$and G_{kn} is systematic,

First k bits of C^* == V^*

First 4 bits of R' == First four bits of C^* which suggests,

First mass = $|x_a| + |x_b| + |x_c| + |x_d| + x_5 c_5^* + x_6 c_6^* + x_7 c_7^*$ ……..…… (4.1)

Observation 3:

Mass Decreases if $R_i' \neq C_i^*$

The arbitrary product term $x_i c_i^*$ from the expression of the "mass". For BPSK modulated symbols, it can be say that,

$X_i' = 0 => x_i$ is negative

$C_i^* = 1 => c_i$ is positive

and the product $x_i c_i^*$ is positive.

Similarly, for the reverse case.

$X_i' = 1 => x_i$ is positive

$C_i^* = 0 => c_i$ is Negative

Thus, the product $x_i c_i^*$ is negative.

The product is negative means, it is going to get subtracted reducing the total "mass".

The analysis depicts the first information. First *k* bits of first best codeword C^* and the first *k* bits of R' are equal. Thus, in calculation of first "mass", absolute of firs *k* soft values of r' are always additive.

Observation 4: (No TEP test condition)

$R' = C_i^*$, i.e. $\forall i$, if $R' = C_i^*$, then

$$mass1 = |x_a| + |x_b| + |x_c| + |x_d| + |x_e| + |x_f| + |x_g| \quad \text{..... (4.2)}$$

Now, consider $C_5^* C_6^* C_7^*$ differ with $X_e X_f X_g$ in any one position, the possible first masses in these situation are,

$$mass2 = |x_a| + |x_b| + |x_c| + |x_d| + |x_e| + |x_f| - |x_g| \quad \text{..... (4.3)}$$

$$mass3 = |x_a| + |x_b| + |x_c| + |x_d| + |x_e| - |x_f| + |x_g| \quad \text{..... (4.4)}$$

$$mass4 = |x_a| + |x_b| + |x_c| + |x_d| - |x_e| + |x_f| + |x_g| \quad \text{..... (4.5)}$$

$mass1 > mass2 > mass3 > mass4 >$ any other combination is a known fact since elements of r' are ordered in decreasing reliability.

The analysis depicts the second information. For no bit difference or single bit difference in the last n-k bits of C^* and R', first mass belongs (mass1, mass2, mass3, mass4) for sure which is already the maximum mass. It suggests that no other TEP can maximize the masses$(\text{mass1}, \text{mass2}, \text{mass3}, \text{mass4})$.

Observation 5: (Single order TEP test condition)

The permuted received word r' is defined as,

$r' = [x_a x_b x_c x_d x_e x_f x_g]$

Hard decoded the permuted received word R'

$R' = [X_a X_b X_c X_d X_e X_f X_g]$

Input vector V^*

$V^* = [X_a X_b X_c X_d]$

For best codeword C^*

$C^* = V^* G^*_{kn} = [C^*_1 C^*_2 C^*_3 C^*_4 C^*_5 C^*_6 C^*_7]$……..... (4.6)

Now, consider $C^*_5 C^*_6 C^*_7$ differ with $X_e X_f X_g$ in any two position, the possible first masses in these situation are,

$mass5 = |x_a| + |x_b| + |x_c| + |x_d| + |x_e| - |x_f| - |x_g|$……..……. (4.7)

$mass6 = |x_a| + |x_b| + |x_c| + |x_d| - |x_e| + |x_f| - |x_g|$……..... (4.8)

$mass7 = |x_a| + |x_b| + |x_c| + |x_d| - |x_e| - |x_f| + |x_g|$……..…. (4.9)

The analysis of equation 4.1-4.9 say that $mass1 > mass2 > mass3 > mass3 > mass4 > mass5 > mass6 > mass7$ but still cannot make a conclusion that this is the only possible ordering of the mass because there are possibilities such that one bit change in V^* increase the mass compared to $mass5, mass6, mass7$.

A TEP '1000' which changes above input vector $V^*_1 = [\bar{X}_a X_b X_c X_d]$ now, since the permuted generator matrix is also a systematic matrix, the first k bits of the codeword $C^*_1 = V^*_1 G^*_{kn} = [\bar{X}_a X_b X_c X_d C^*_5 C^*_6 C^*_7]$

Check the last $(n-k)$ bits of codeword C^*_1, it can find that, each of those bits are the linear combination of any three input vector bits. It means any one bit change in the input vector can change the at least two or three of the last $(n-k)$ bits of codeword C^*_1. Thus, possible masses for this condition can be noted as,

$mass8 = -|x_a| + |x_b| + |x_c| + |x_d| + |x_e| + |x_f| + |x_g|$..... (4.10)

$mass9 = -|x_a| + |x_b| + |x_c| + |x_d| + |x_e| + |x_f| - |x_g|$..... (4.11)

$mass10 = -|x_a| + |x_b| + |x_c| + |x_d| + |x_e| - |x_f| + |x_g|$..... (4.12)

$mass11 = -|x_a| + |x_b| + |x_c| + |x_d| - |x_e| + |x_f| + |x_g|$..... (4.13)

$mass12 = -|x_a| + |x_b| + |x_c| + |x_d| + |x_e| - |x_f| - |x_g|$..... (4.14)

$mass13 = -|x_a| + |x_b| + |x_c| + |x_d| - |x_e| + |x_f| - |x_g|$..... (4.15)

$mass14 = -|x_a| + |x_b| + |x_c| + |x_d| - |x_e| - |x_f| + |x_g|$..... (4.16)

All the possibilities from equation 4.1- 4.16, only one mass, i.e. $mass8$ might be (not always) greater than $mass5, mass6, mass7$. If this is true then it locates $mass_{max}$ corresponding to above TEP. If this is not true, then the next TEP of order 1 can satisfy thus finding $mass_{max}$ in next TEP of order 1.

The difference of number between C^* and R' in any two position gives the information that no TEP of order greater than 1 is going to give $mass_{max}$, because maximum masses with second or greater order TEP like,

$mass15 = |x_a| + |x_b| - |x_c| - |x_d| + |x_e| + |x_f| + |x_g|$..... (4.17)

$mass16 = |x_a| - |x_b| + |x_c| - |x_d| + |x_e| + |x_f| + |x_g|$..... (4.18)

$mass17 = |x_a| - |x_b| - |x_c| + |x_d| + |x_e| + |x_f| + |x_g|$..... (4.19)

$mass18 = -|x_a| + |x_b| + |x_c| - |x_d| + |x_e| + |x_f| + |x_g|$..... (4.20)

$mass19 = -|x_a| + |x_b| - |x_c| + |x_d| + |x_e| + |x_f| + |x_g|$..... (4.21)

$mass20 = -|x_a| - |x_b| + |x_c| + |x_d| + |x_e| + |x_f| + |x_g|$..... (4.22)

can never be greater masses $mass1, mass2, mass3, mass4, mass5, mass6$ and $mass7$. When the C^*differ with R' in any two position, the single order TEPs test is sufficient in order to find $mass_{max}$ which proves the second steps, point 'b' of the TEP reduction analysis.

Observation 5: (single and double order TEP test condition)

The permuted received word r'.is defined as

$r' = [x_a x_b x_c x_d x_e x_f x_g]$

Hard decoded the permuted received word R^l

$R' = [X_a X_b X_c X_d X_e X_f X_g]$

Input vector V^*

$V^* = [X_a X_b X_c X_d]$

For best codeword C^*

$C^* = V^* G^*_{kn} = [C^*_1 C^*_2 C^*_3 C^*_4 C^*_5 C^*_6 C^*_7]$

Now, consider $C^*_5 C^*_6 C^*_7$ differ with $X_e X_f X_g$ in all three position, the possible first masses in these situation is,

$mass21 = |x_a| + |x_b| + |x_c| + |x_d| - |x_e| - |x_f| - |x_g|$…….... (4.23)

The possibilities of finding $mass_{max}$ by changing at most any two bits of V^* searching is given by,

$mass22 = |x_a| + |x_b| - |x_c| - |x_d| + |x_e| + |x_f| + |x_g|$…..... (4.24)

$mass23 = |x_a| - |x_b| + |x_c| - |x_d| + |x_e| + |x_f| + |x_g|$……..……..... (4.25)

$mass24 = |x_a| - |x_b| - |x_c| + |x_d| + |x_e| + |x_f| + |x_g|$…..... (4.26)

$mass25 = -|x_a| + |x_b| + |x_c| - |x_d| + |x_e| + |x_f| + |x_g|$..... (4.27)

$mass26 = -|x_a| + |x_b| - |x_c| + |x_d| + |x_e| + |x_f| + |x_g|$..... (4.28)

$mass27 = -|x_a| - |x_b| + |x_c| + |x_d| + |x_e| + |x_f| + |x_g|$..... (4.29)

Which might be greater than $mass21$.

The analysis of equation 4.23 - 4.29 gives the information that, C^* differ with R' in any 3 position, no TEP of order greater than 2 is going to give $mass_{max}$, because maximum masses with third or greater order TEP like,

$mass28 = |x_a| - |x_b| - |x_c| - |x_d| + |x_e| + |x_f| + |x_g|$..... (4.30)

$mass27 = -|x_a| - |x_b| + |x_c| - |x_d| + |x_e| + |x_f| + |x_g|$..... (4.31)

$mass30 = -|x_a| - |x_b| - |x_c| + |x_d| + |x_e| + |x_f| + |x_g|$..... (4.32)

can never be greater than $mass21$ which clarifies that in case where the C^* differ with R' in any 3 position, it is sufficient to test $mass_{max}$ with TEPs of order less than 3, which proves the second step, point 'c' of the TEP reduction analysis..

4.3 MRB Decoding and Conventional Reprocessing

The linear block code $C(n, k)$ with systematic generator matrix G_{kn}, at the transmitter side a k −bit information vector.is given by,

$\bar{\mathbf{v}} = (v_1, v_2, \ldots, v_3)$..... (4.33)

is mapped into codeword.

$\bar{\mathbf{c}} = V.G_{kn} = (c_1, c_2, \ldots, c_n)$..... (4.34)

Under Binary Phase Shift Keying (BPSK), the codeword is mapped into a real-valued vector

$\bar{\mathbf{s}} = (s_1, s_2, \ldots, s_n)$ ..... (4.35)

Where

$c_i = 0 \rightarrow s_i = -1$

$c_i = 1 \rightarrow s_i = +1$

The vector $\bar{\mathbf{s}}$ is transmitted over an Additive White Gaussian Noise (AWGN) channel. At receiver side, the received vector can be obtained as,

$\bar{r} = (r_1, r_2, \ldots, r_n)$ …….. (4.36)

Where: $r_i = s_i + w_i$, w_i is White Gaussian noise sample with mean zero and variance σ^2. RBA start by reordering the received vector by decreasing magnitude. The first symbol characterized by a high probability, i.e. a large probability of being correct. Given $\bar{r}$, by reordering its components for decreasing magnitude $\lambda_i = |r_i|$, a vector is given by,

$\bar{r}' = (r_1^*, r_2^*, \ldots, r_n^*)$…….. (4.37)

Such that $|r_i^*| > |r_{i+1}^*|$ for $1 \leq i \leq n$.

The generator matrix G_{kn} is also permuted using the permutation rule, to give the new permuted generating matrix. The permuted generating matrix is then processed by using elementary row operation to obtain a systematic form G_{kn}^*. Given $\bar{r^*}$,a symbol-by-symbol hard decision is used to obtain the binary vector:

$\bar{y}' = (y_1', y_2', \ldots, y_n')$ …… (4.38)

Where

$r_i' < 0 \rightarrow y_i' = 0$

$r_i' \geq 0 \rightarrow y_i' = 1$

Given $\bar{y}'$, first k bits to form the candidate information vector:

$\bar{v}^* = (v_1^*, v_2^*, \ldots, v_n^*)$ …….... (4.39)

OSD algorithm consider a set of patterns

$s = \{\overline{\mathbf{p}} = (p_1, p_2, \ldots\ldots, p_k)$ ..... (4.40)

With hamming weight $0 \leq w_H(\overline{\mathbf{p}}) \leq I$, where I is called the order of the algorithm. Each pattern is added to the candidate information vector, which is encoded by the matrix $\boldsymbol{G}_{\boldsymbol{kn}}^{*}$ to obtain a reprocessing codeword. When all patterns have been considered, the codeword $\bar{\boldsymbol{c}}^{*}$ at minimum Euclidean distance from the permuted received vector $\bar{\boldsymbol{r}}'$ is chosen as the received codeword. Since the first k bits have high reliability, most of them are correct. If $\bar{v}'$ contains no errors, the weight-zero pattern generates the maximum-likelihood best codeword, and so on.

4.4 Ordered vector component statistics

Given a linear block code $C(n, k)$ and consider all-zero codeword transmitted codeword:

$\bar{c} = (0, \ldots\ldots, 0 \ldots\ldots, 0)$,

Which is after BPSK mapping, corresponds to the transmitted vector:

$\bar{s} = (-1, \ldots\ldots, -1, \ldots\ldots, -1)$

At the output of AWGN channel, the received vector $\bar{r} = (r_{1,} \ldots\ldots, r_i, \ldots\ldots, r_n)$, with:

$r_i = -1 + w_i$, ... (4.41)

Where w_i is Gaussian random variable with zero mean and variance σ^2 is observed.

Each component r_i has a Probability Density Function (pdf) given by:

$f_r(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x+1)^2}{2\sigma^2}}$ ..... (4.42)

It is considered that the magnitude of the components of $\bar{\boldsymbol{r}}$ written as $\lambda_i = |r_i|$, the pdf of λ_i is given by,

$$f_r(x) = \begin{cases} 0, x < 0 \\ \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x+1)^2}{2\sigma^2}} + \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-1)^2}{2\sigma^2}}, x \geq 0 \end{cases} \quad \text{..... (4.43)}$$

While it's Cumulative Distribution Function (cdf) is given by,

$$f_r(x) = \begin{cases} 0, x < 0 \\ \frac{1}{2}\text{erf}\frac{(x+1)^2}{2\sigma^2} + \frac{1}{2}\text{erf}\frac{(x-1)^2}{2\sigma^2}, x \geq 0 \end{cases} \quad \text{..... (4.44)}$$

The vector $\bar{r}$ observed by transmitting the all-zero codeword over an AWGN channel under BPSK modulation, the ordered vector $\bar{r}'$ obtained by ordering $\bar{r}$ in decreasing magnitude.

The pdf of the $i-th$ component r_i' of $\bar{r}'$ is given by:

$$\forall x: f_{r_i'}(x) = \frac{n!}{(i-1)!(n-i)!}(1 - F_\lambda(|x|))^{i-1}(F_\lambda(|x|)^{n-i} f_r(x) \text{..... (4.45)}$$

The pdf of the $i-th$ magnitude of component $\lambda_i' = |r_i'|$ of $\bar{r}'$is given by,

$$\forall x: f_{\lambda_i'}(x) = \frac{n!}{(i-1)!(n-i)!}(1 - F_\lambda(x))^{i-1}(F_\lambda(x)^{n-i} f_\lambda(x) \text{..... (4.46)}$$

4.5 Error information Based on Permuted Codeword

Introduce three quantities, computed starting from the ordered vector$\bar{r}'$, which are important in OSD algorithm. Given $\bar{r}'$, a symbol-by-symbol hard decision is used to obtain a binary received vector.

$$\bar{y}' = (y_1', \ldots, y_2', \ldots, y_n') \text{..... (4.47)}$$

For further reprocessing, first k Most Reliable Information (MRI) bits of $\bar{y}'$ are considered to form a candidate information vector $\bar{v}'$. The candidate information is processed by adding a Test Error Pattern (TEP) and encoded by the matrix G_{kn}^* to obtain a reprocessing codeword. If c' is the permuted version of the true codeword which is called true permuted codeword, there are two possibilities connected to codeword c^*which can be formulated by simple binary Hypothesis Testing. Under Null Hypothesis H_0, a codeword which is not equal to the true permuted codeword is obtained, i.e.

$c^*|_{H_0} \neq c'$…….. ……... (4.48)

and under Alternate HypothesisH_1, true permuted codeword c' is obtained, i.e.

$c^*|_{H_1} = c'$……..... (4.49)

There are different basis for comparing these two hypothesis for given code of size (n, k, d_{min}). The most common and effective method is the basis of Euclidean distance which compares the Euclidean distance between the permuted received vector $r'$and the modulated soft valued vector corresponding to the codeword c^*. The basis of Euclidean distance is combined with the basis of hamming distance between the permuted binary received vector $\bar{y}'$ and the reprocessing codeword c^*.

Define two Random Variables (RVs) measuring the Hamming distance between the permuted binary received vector $\bar{y}'$ and the reprocessing codewordc^*.

1) RV E_1 = It considers the Hamming Distance between Permuted Binary received vector $\bar{\boldsymbol{y}}'$ and the reprocessing codeword $\bar{\boldsymbol{c}}^*$ in the first $\boldsymbol{L}$ positions, $1 \leq l \leq n$, under a predefined scenario supported by H_1.
2) RV E_2: It considers the Hamming distance between permuted binary received vector $\bar{y}'$ and the reprocessing codeword c^* in the parity check section under a predefined scenario supported by H_1.
3) RV E_3: It considers the Hamming distance between permuted binary received vector $\bar{y}'$ and the parity check section under a predefined scenario supported by H_0 .

4.6 Analytical computation of Probability Density Function of E_1 and E_2 and E_3

The pdf of random variable E_2 is given by

$$f_{E_2}(E_2 = j) = \int_0^{+\infty} \binom{n-k}{j} p^j (1-p)^{n-k-j} f_{\lambda_k'}(x)dx \quad \text{(4.50)}$$

Where

$$p = \frac{\left[\text{erf}\left(\frac{x+1}{\sigma\sqrt{2}}\right) - \text{erf}\left(\frac{1}{\sigma\sqrt{2}}\right)\right]}{\left[\text{erf}\left(\frac{x+1}{\sigma\sqrt{2}}\right) - \text{erf}\left(\frac{-x+1}{\sigma\sqrt{2}}\right)\right]} \quad \text{(4.51)}$$

First of all, the distribution of the samples in the tail of the permuted received vector which fall in the range of $-x$ to x as shown in figure 4.1.

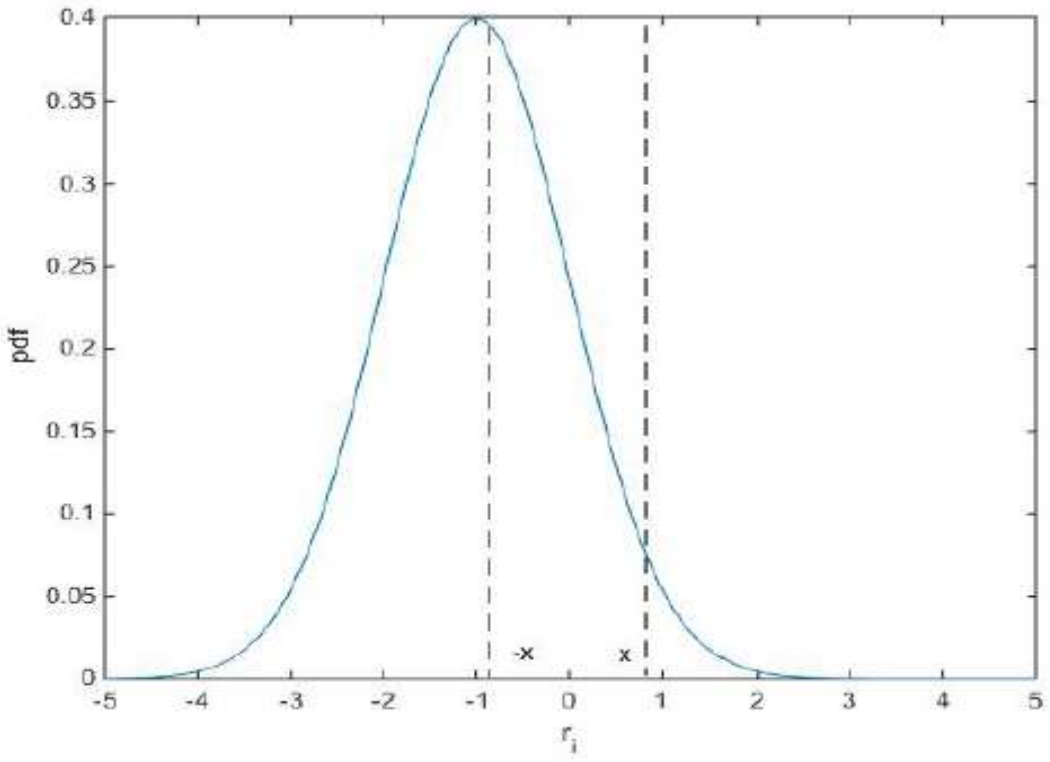

Figure 4.1: PDF of r_i

Fix a value$x$, and suppose the magnitude of the $(k)-th$ component of the reordered vector $\bar{\boldsymbol{r}}'$ is $\boldsymbol{\lambda_k'}$. Then, the vector $\bar{\boldsymbol{r}}$ contains exactly n-k components with $|r_i| \leq x$. As can be observed in Figure 4.1, for each of these components, the probability of having an error is

$$= \frac{F_{R_i}(x) - F_{R_i}(0)}{F_{R_i}(x) - F_{R_i}(-x)} \quad \text{(4.52)}$$

$$= \frac{[\text{erf}\left(\frac{x+1}{\sigma\sqrt{2}}\right) - \text{erf}\left(\frac{1}{\sigma\sqrt{2}}\right)]}{[\text{erf}\left(\frac{x+1}{\sigma\sqrt{2}}\right) - \text{erf}\left(\frac{-x+1}{\sigma\sqrt{2}}\right)]} \quad \text{(4.53)}$$

The components of received vector $\bar{\boldsymbol{r}}$ which are not reordered thus statistically independent, the probability of having j errors among these k components is given by:

$$P(E_2 = j|_{\{\lambda_k' = x, L = n-k\}} = \binom{n-k}{j} p^j (1-p)^{n-k-j} \quad \text{(4.55)}$$

The above result is obtained under the condition $\boldsymbol{\lambda'_k = x}$. By integrating over all x values by using $f_{\lambda'_k}(x)$, the final result obtained as (4.49)

Similar concept of the expression of random variable E_1, the probability density function of the variable $E_1|_{L=l}$ is given by

$$f_{E_1(L=l)}(E_1 = j) = \int_0^{+\infty} \binom{l}{j} p^j (1-p)^{l-j} f_{\lambda'_{l+1}}(x)dx \quad \text{(4.56)}$$

Where $p = \frac{\left[\text{erf}\,c\left(\frac{x+1}{\sigma\sqrt{2}}\right)\right]}{\left[1+\text{erf}\,c\left(\frac{x+1}{\sigma\sqrt{2}}\right)-\text{erf}\left(\frac{-x+1}{\sigma\sqrt{2}}\right)\right]}$ ..... (4.57)

Fix a value$x$, the magnitude of the $(l+1)$-th component of reordered vector $\bar{\boldsymbol{r}}'$ is x. Then, vector $\bar{\boldsymbol{r}}$ contains exactly l components with $|r_i| \geq x$ where the probability of having an error is

$$\frac{\left[\text{erf}\,c\left(\frac{x+1}{\sigma\sqrt{2}}\right)\right]}{\left[1+\text{erf}\,c\left(\frac{x+1}{\sigma\sqrt{2}}\right)-\text{erf}\left(\frac{-x+1}{\sigma\sqrt{2}}\right)\right]} \quad \text{(4.58)}$$

The probability of having j errors among these k components is given by,

$$P\left(E_1 = j|_{\lambda'_{l+1}=x, L=l}\right) = \binom{l}{j} p^j (1-p)^{l-j} \quad \text{(4.59)}$$

The equation 4.57 is obtained under the condition $\lambda'_{l+1} = x$. By integrating over all x values by using $f_{\lambda'_{l+1}}(x)$ and obtained the final pdf of E_1 as in equation 4.54 and its cdf at some value $E_1 = j$ is obtained at in equation 4.55 by summing the normalized pdf of E_1 for all E_1: $0 \leq E_1 \leq j$.

Define D_w the weight distribution of a linear block $C(n, k, d_{min})$ under consideration. Now define E_3 in term of hamming distance and eventually show its relation with the weight distribution of the considered linear block code.

$$E_3 = d_H^p(\bar{y}', \bar{c}^*)|_{H_0} \quad \text{(4.60)}$$

Where d_H^p is the hamming distance operator applied only on parity check section of the vector under consideration. Equivalently, $\bar{y}'$ can be written as a sum of the true

permuted codeword $\bar{c}'$ and the error vector $\bar{e}'$. Also $\bar{c}^*$ can be written as sum of true permuted codeword $\bar{c}'$ and a codeword corresponding to a particular TEP defined by $\bar{c}_T = \bar{\boldsymbol{p}}.\boldsymbol{G}_{\boldsymbol{kn}}^*$.

$E_3 = d_H^p(\bar{c}' + \bar{e}', \bar{c}' + \bar{c}_T)$…… (4.61)

$= d_H^p(\bar{0}, \bar{e}' + \bar{c}_T)$…… ..(4.62)

Where $\bar{0}$ is an all-zero codeword.

By the definition of Null hypothesis, for all cases, $\bar{c}^T|_{H_0} \neq \bar{0}$. Now, the scenario can be explained well for two different cases of error vector $\bar{e}'$.

a. When $\bar{e}' = \bar{0}$ which occurs at very high SNR. Given a TEP of order i , the distribution of E_3 can be written in terms of weight distribution as,

   $E_3 = d_H^p(\bar{0}, \bar{c}_T)|_{H_0, \bar{e}'=\bar{0}} \equiv D_w - i$ …….. (4.63)

   TEP of all orders are assumed but exclude the effect of TEP order. Thus, in this scenario the distribution of E_3 is equivalent to the weight distribution of the code.

b. When $\bar{e}' \neq \bar{0}$ which occurs at nominal SNR. Theoretically E_3 is dependent on all three components, i.e. TEP order, the error vector and the weight distribution. However, codewords can be considered to have good properties in terms of distribution of 1's and 0's within the codeword, with which we can assume that summing a error vector does not change distribution of hamming weight of a codeword. Thus, when considering TEP of all orders, the distribution of E_3 is the equivalent the weight distribution of the code.

   Finally, it can be write as,

   $E_3 = D_w$ ……. (4.64)

4.7 OSD Error Performance Based on Distribution of E_1

A different look at the error performance of order-I OSD is presented based on the distribution of E_1. Define $Pe_{OSD-I}(\bar{\boldsymbol{c}})$ the code error performance of the order-I OSD

and $P_e(I)$ the probability that the correct codeword is not among the candidate code-words supported by the order-I OSD. The upper bound on the order-I OSD performance can be written as an inequality as:

$$Pe_{OSD-I}(\bar{\boldsymbol{c}}) \leq P_{eML}(\bar{\boldsymbol{c}}) + P_e(I) \quad \text{(4.65)}$$

Where $P_{eML}(\bar{\boldsymbol{c}})$ is the MLD code error rate $P_e(I)$ can be simply evaluated as the probability of having more than I errors in the first- K ordered received symbols in $\bar{\boldsymbol{r}}'$given a identity permutation function ρ_2. i.e.,

$$P_e(I) = 1 - F_{E_1}(I) \quad \text{(4.66)}$$

The permutation function ρ_2 may or may not be identity. Consider d is the number of dependent columns before k^{th} independent one and $p(d)$ is the probability associated with d. The maximum number of dependent columns that can be found for a given generating matrix is given by

$$d_{max} = n - k - d_H - 1 \quad \text{(4.67)}$$

Thus, $P_e(I)$ can be expressed under all cases of ρ_2

$$P_e(I) = \sum_{d=0}^{d_{max}} P\begin{pmatrix} More\ than\ I\ errors\ occured \\ in\ 1^{st}\ k+d\ positions\ of\ \bar{r}' \end{pmatrix} P(d) \quad \text{(4.68)}$$

$$= \sum_{d=0}^{d_{max}} P(d)(1 - F_{E_1(k+d)}(I) \quad \text{(4.69)}$$

Where $F_{E_1(k+d)}$ is the cdf of E_1 at $L = k + d$.

4.8 Performance and complexity trade-off based on GPTT

The test performance can be evaluated by the well-known ROC parameters where probability of False Alarm and Probability of detection defined respectively as follows

$$P_F = pr(E_T < T|_{H_0}) \quad \text{(4.70)}$$

$$P_D = pr(E_T < T|_{H_1}) \quad \text{(4.71)}$$

4.9 Error performance of OSD with GPTT Based on stopping criterion

The order-k reprocessing with the threshold equal to zero achieves the maximum likelihood decoding and requires 2^k computations. Threshold equal to zero corresponds to a probability of false alarm equal to zero which means GPTT with threshold equal to zero does not limit the number of computation at all. Thus while considering order-k reprocessing with GPTT (T>0≡ P_F>0). Some of the unnecessary computation are spared with the price of certain probability of error. The probability of error due to the GPTT false alarm is upper bounded by the probability that GPTT gives a false alarm, given the codeword is within the codewords supported by the order - I OSD.

$$P_{e-GPTT} = P_F \times \begin{pmatrix} Probability\ of\ having\ I \\ or\ less\ errors\ in\ MRI \end{pmatrix} \quad (4.72)$$

$$= P_F \times F_{E_1(k)}(I) \quad (4.73)$$

4.10 Computational Savings

Let $S_1(k) = \{1,2, \ldots\ldots, k\}$ be the index set for the first k positions of $\bar{\boldsymbol{y}}'$. For $1 \leq i \leq k$, let $S_2(i) = \{j_1, j_2, \ldots\ldots j_i\}$ be the proper subset of i elements of S_1 . If the elements in S_2 denotes the error positions in the information vector, then the TEP probability can be related to the probability of error in the positions indicated by the index set S_2 as,

$$P_{TEP}(\bar{\boldsymbol{p}}) = P_e(S_2) \quad (4.74)$$

The order of TEP is arranged based on the decreasing order of the $P_{TEP}(\bar{\boldsymbol{p}})$ which optimize the reprocessing algorithm, . Denote T_I the total number of TEP supported by the I −order reprocessing which is given by

$$T_I = \sum_{t_I=1}^{T_I} \binom{k}{i} \quad (4.75)$$

The number of TEP's to be reprocessed for particular codeword varies depending the E_b/N_0. In ordered TEP reprocessing, the number of TEPs to be reprocessed for locating the correct codeword is clearly dependent on the number of errors in MRI bits. If the number of errors in the MRI bits is less than or equal to I, then using a

GPTT stopping criterion, the algorithm may or may not require test all the TEPs supported by the order-I reprocessing otherwise if the number of errors in the MRI bits are greater than I, all the TEPs has to be reprocessed in order to declare a received codeword unless a GPTT False alarm occur. The scenario can be divided in two conditions as

4.10.1 Number of errors in MRI is less than or equal to I

The probability of having I or less error in MRI can be easily evaluated using the cdf of E_1 keeping $L = K$. Number of TEP's to be reprocessed depends upon the GPTT outcomes i.e. Normal detection and Missed detection.

Normal detection: Consider a GPTT test truly detects a true permuted codeword is located corresponding to a TEP at position t_I . If GPTT correctly detects the average number of TEP's (T_N) required to located the true permuted codeword is given by,

$$T_N = \sum_{t_I=1}^{T_I} P_{TEP}\left(\bar{P}_{t_I}\right).t_I \quad \text{(4.76)}$$

Missed detection: Under a missed detection, it is straight forward to say that normally the reprocessing algorithm has to test all the TEP's to locate the true permuted codeword. Thus, for a given miss, average number of TEP's (T_M) to be reprocessed is given by,

$$T_M = T_I \quad \text{(4.77)}$$

4.10.2 Number of errors in MRI is more than I: The GPTT cannot detect the true codeword unless a False alarm is triggered, thus all the TEP has to be reprocessed. The probability of having more than I errors can be evaluated as a complementary cdf of E_1 keeping $L = k$. If T_G is the average number of TEPs to be reprocessed in this case, it is given by,

$$T_G = T_I \quad \text{(4.78)}$$

In total, the average number of TEPs computations required to decide for a true permuted codeword can be obtained by summing T_N , T_M , T_G after weighting each of them by their respective probabilities, then the average number of TEP's (N_A) to be reprocessed to locate the true permuted codeword is given by,

$$N_A = F_{E_1}(I)\left[P_D\left(\sum_{t_I=1}^{T_I} P_{TEP}\left(\bar{P}_{t_I}\right)t_I\right) + P_M T_I\right] + (1\text{-}F_{E_1}(I))T_I \quad \text{..... (4.79)}$$

CHAPTER FIVE: RESULTS, ANALYSIS AND COMPARISION

5.1 Results

Table 5.1: Generating matrix, transmitted bit and codeword

```
k =

     4


n =

     7


G =

     1     0     0     0     1     1     0
     0     1     0     0     1     0     1
     0     0     1     0     0     1     1
     0     0     0     1     1     1     1

tran_dat =

     1     0     1     1


Ct =

     1     0     1     1     0     1     0


St =

     1    -1     1     1    -1     1    -1
```

Table 5.2: Ordered of received symbols

```
Sr =

    1.2189   -0.9799    0.0609    0.1140   -1.3502   -0.1093   -1.9877


rum =

    1.2189    0.9799    0.0609    0.1140    1.3502    0.1093    1.9877


ord =

    1.9877    1.3502    1.2189    0.9799    0.1140    0.1093    0.0609


perm =

     7     5     1     2     4     6     3
```

Table 5.3: Systematic matrix

```
mat2 =

     1     0     0     0     0     1     1
     0     1     1     0     0     1     0
     0     0     1     1     0     0     1
     0     0     0     1     1     1     0


mat2 =

     1     0     0     0     0     1     1
     0     1     0     1     0     1     1
     0     0     1     1     0     0     1
     0     0     0     1     1     1     0


mat2 =

     1     0     0     0     0     1     1
     0     1     0     0     1     0     1
     0     0     1     1     0     0     1
     0     0     0     1     1     1     0


mat2 =

     1     0     0     0     0     1     1
     0     1     0     0     1     0     1
     0     0     1     0     1     1     1
     0     0     0     1     1     1     0
```

Table 5.4: First best codeword

```
S2_r =

   -1.9877   -1.3502    1.2189   -0.9799    0.1140   -0.1093    0.0609


v_2 =

     0     0     1     0     1     0     1


vv_2 =

     0     0     1     0


cc_2 =

     0     0     1     0     1     1     1
```

Table 5.5: Distance for all the test error patterns

```
      0     0     1     0


cc_2 =

      0     0     1     0     1     1     1


vr2 =

      1     0     1     1


M =

    5.6023
    1.7237
    2.5521
    3.0333
    3.6331
   -1.0829
   -1.0389
   -0.6827
    0.6827
    1.0389
    1.0829
   -3.6331
   -3.0333
   -2.5521
   -1.7237
```

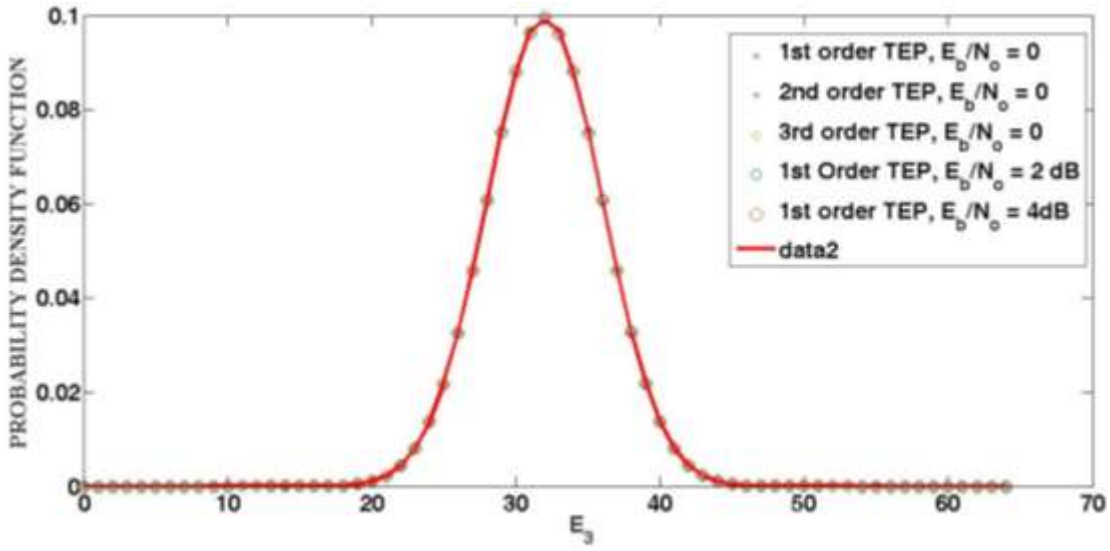

Figure 5.1: Probability density function of E_3

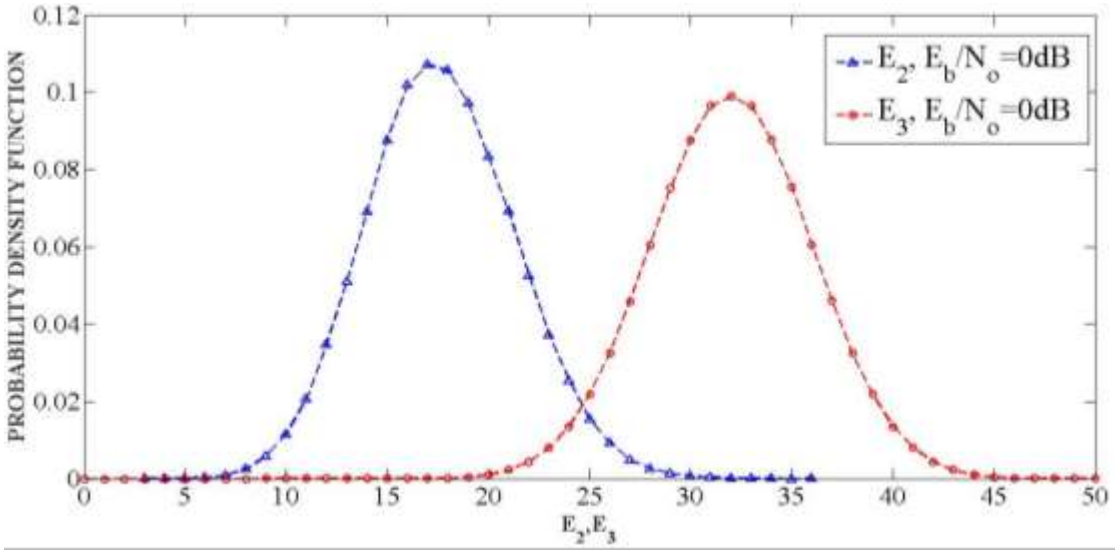

Figure 5.2: Probability density function of E_2 and E_3 at 0dB

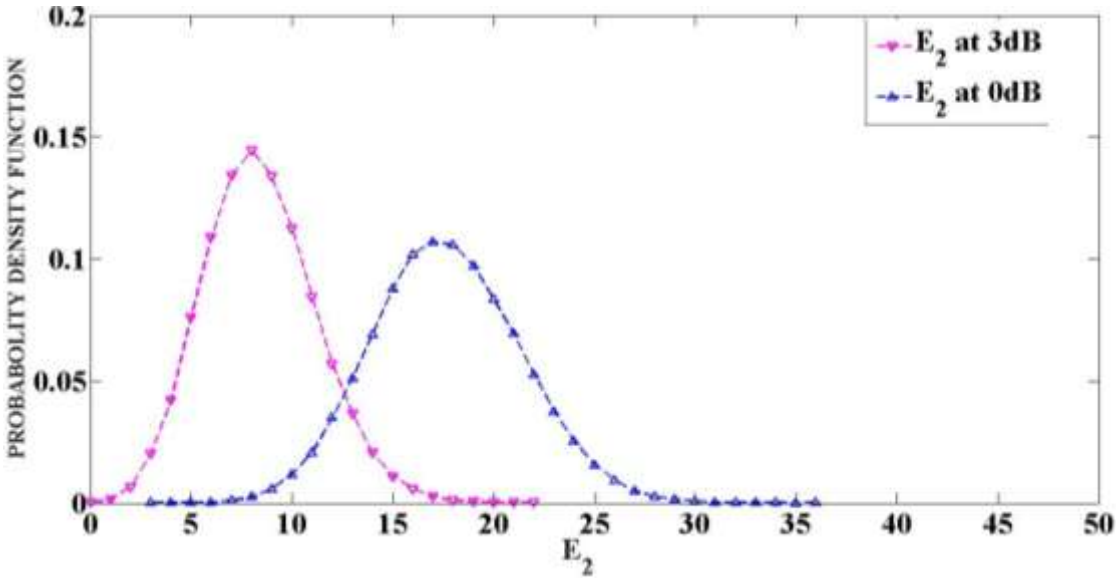

Figure 5.3: Probability density function of E_2 at different SNR

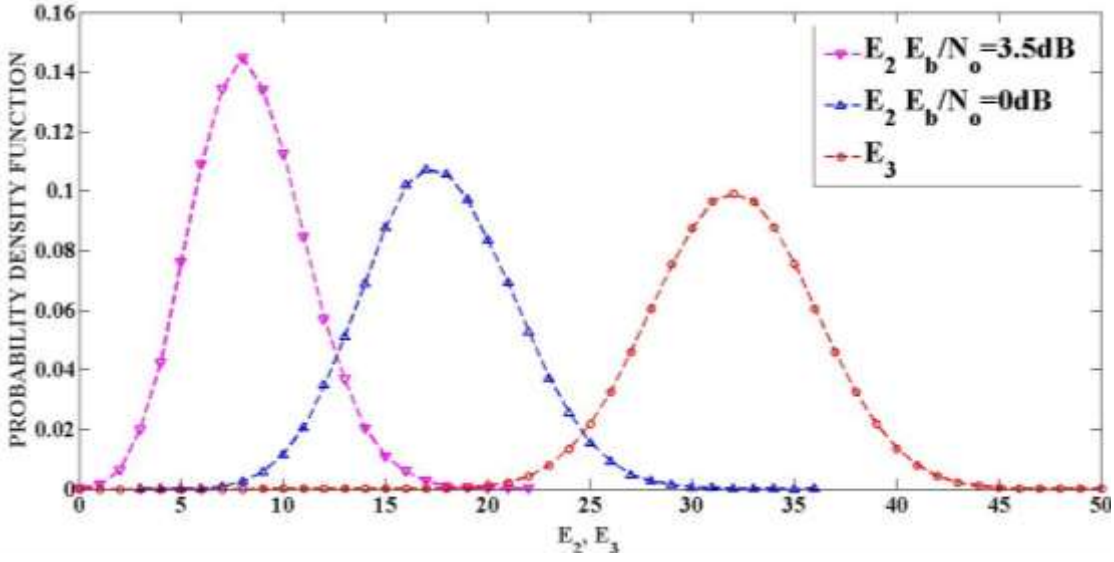

Figure 5.4: Pdf of E_2 at different SNR compared with Pdf of E_3

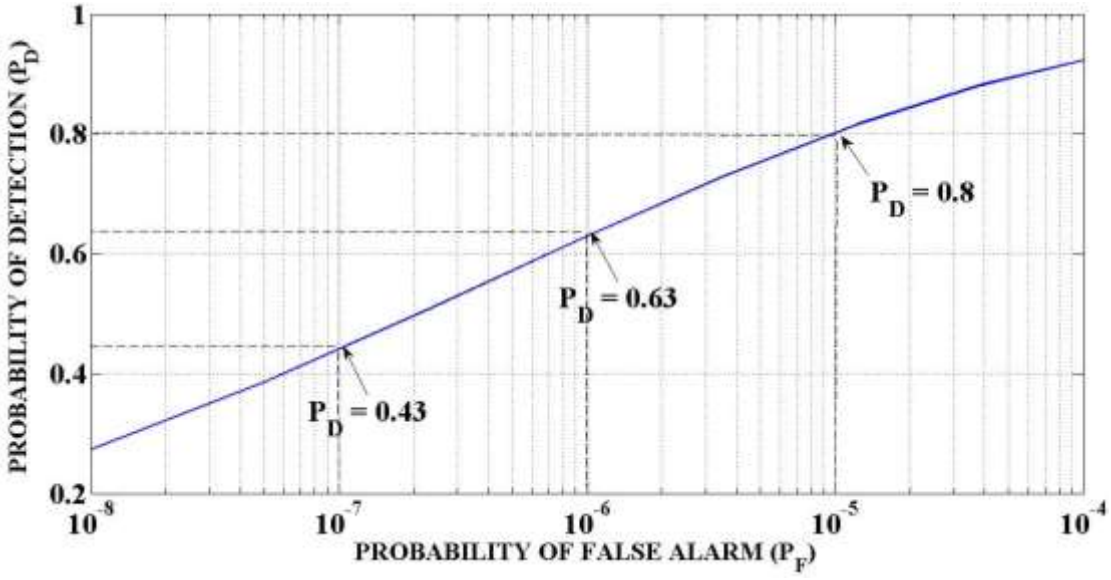

Figure 5.5: ROC for GPTT: E_b/N_o =2dB for (128, 64, 22) Extended BCH code

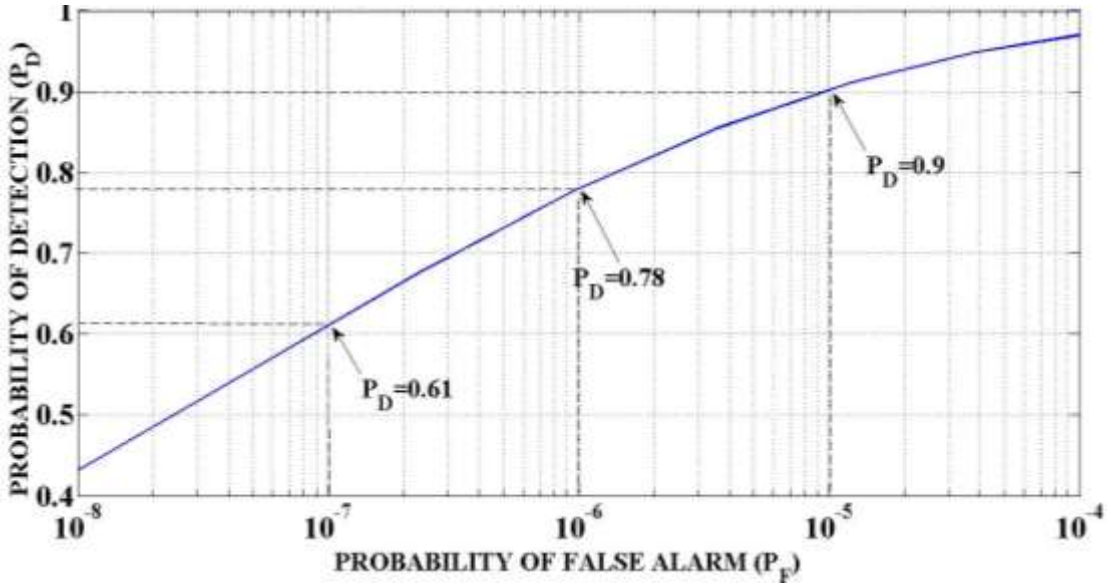

Figure 5.6: ROC for GPTT: E_b/N_o =2.5dB for (128, 64, 22) Extended BCH code

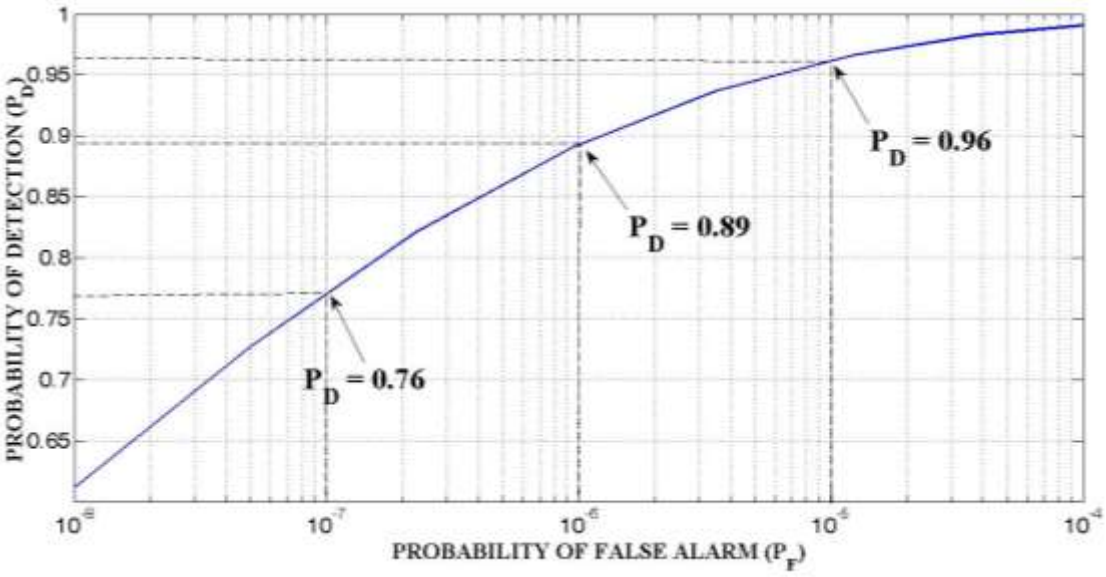

Figure 5.7: ROC for GPTT: E_b/N_o =3dB for (128, 64, 22) Extended BCH code

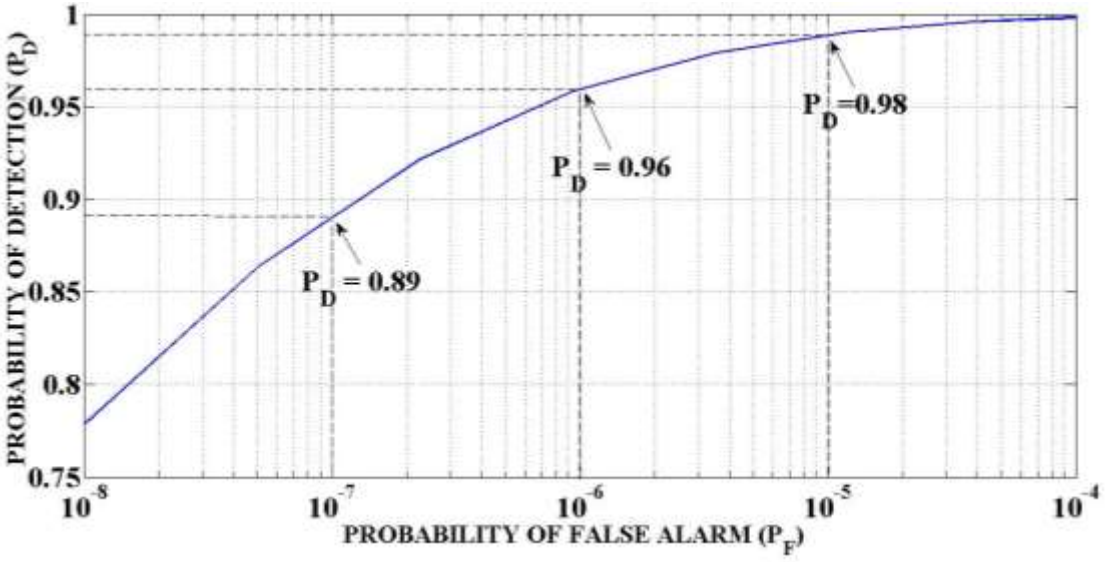

Figure 5.8: ROC for GPTT: E_b/N_o =3.5dB for (128, 64, 22) Extended BCH code

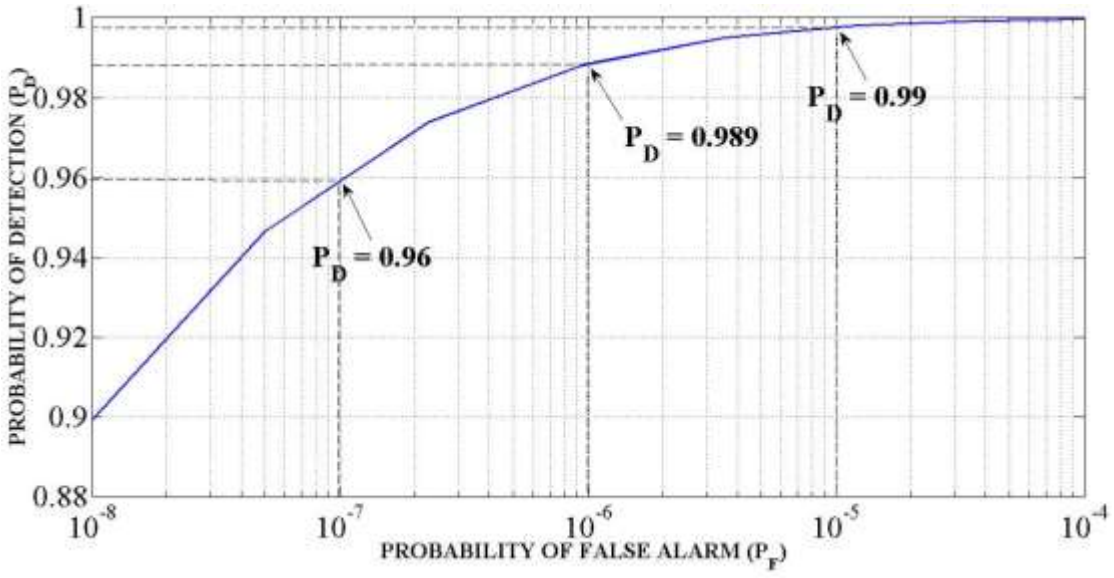

Figure 5.9: ROC for GPTT: E_b/N_o = 4dB for (128, 64, 22) Extended BCH code

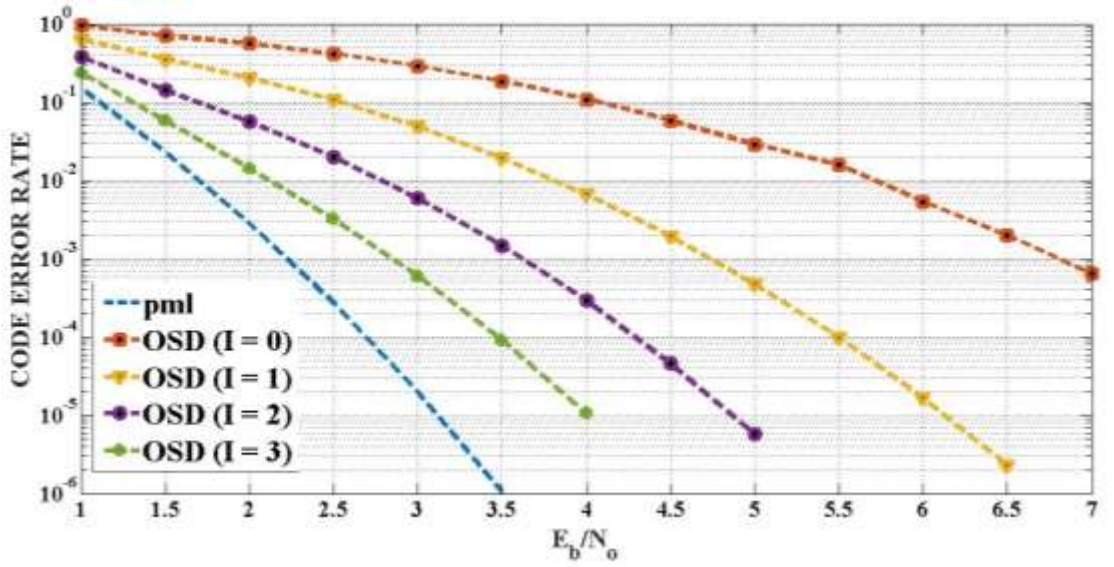

Figure 5.10: Code Error Rate of OSD at each reprocessing stage for (128, 64, 22) extended BCH code

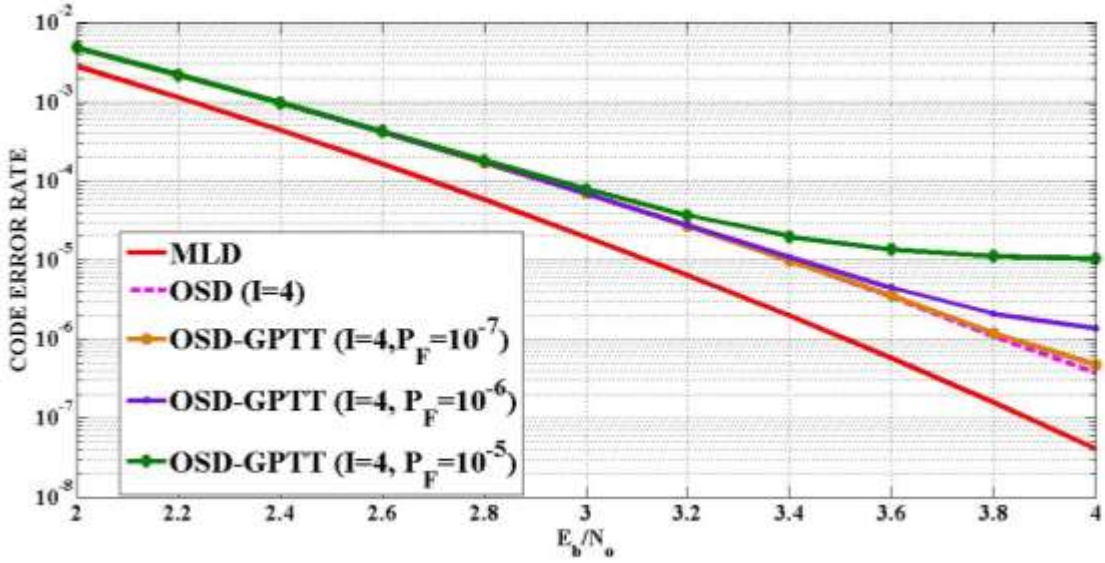

Figure 5.11: Code error rate of OSD-GPTT for (128, 64, 22) extended BCH code: Reprocessing order I=4

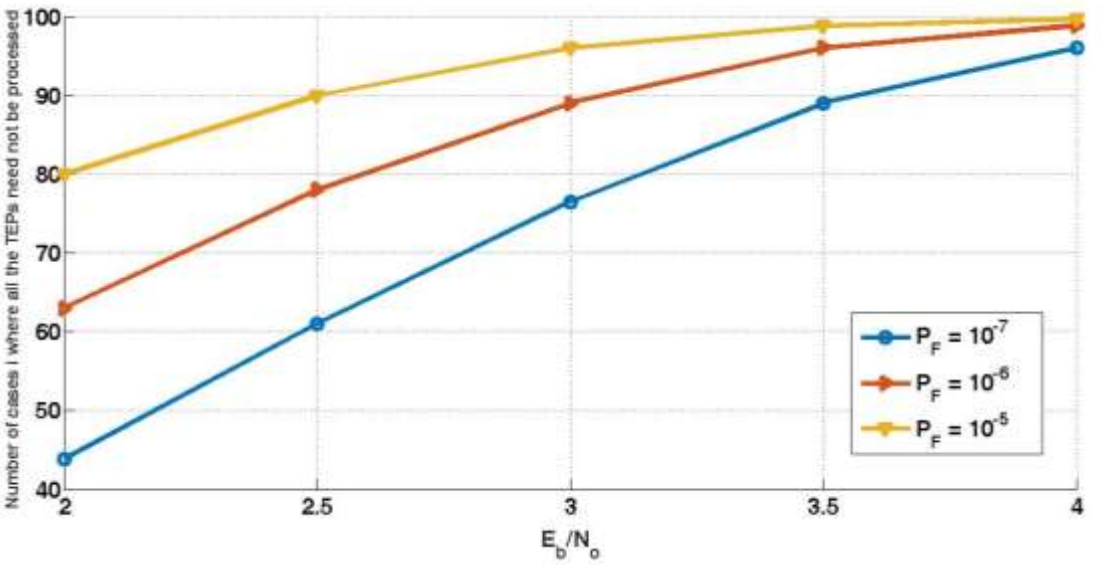

Figure 5.12: Reduction Capability in Percentage for (128, 64, 22) extended BCH code

5.2 Analysis

5.2.1 Generation of Codeword Analysis

Table 5.1 shows the generating matrix which is used to produce codeword by multiplying with transmitted bit and the codeword is modulated by BPSK signaling. Codeword can be obtained by multiplying the generator matrix with information bits. BPSK signaling produced 1 if the symbol in codeword is 1 and -1. If the symbol in codeword is 0. The first k bits of codeword represent the information digits and the remaining n-k digits represent the redundancy check bits. The first step is reorder the received symbols based on their reliability. In Table 5.1, k and n represents the length of information and codeword respectively. G denotes the systematic generator matrix. c_t represents the codeword and s_t represents the codeword after binary phase shift keying.

5.2.2 Ordered received Symbol Analysis

The output of the channel can be reordered according to their reliability. The absolute value of symbol which have greater probability can be placed as first position and the absolute value of received symbol which have least probability can be placed at last position. Table 5.2 shows the reordering of received symbols. After the reordering of received symbol the position of column of systematic generating matrix is changed. In table 5.2, ord represents the ordered received symbol according to their reliability measure.

5.2.3 Permuted Generator matrix analysis

The received vector can rearranged according to their reliability. When the received symbols are reordered, then the position of the symbol is changed. So, the column position of the generating matrix is changed which is not systematic. The matrix can be changed into systematic by elementary row operation and the systematic matrix after reordering the received symbol is called as permuted generating matix. mat2 represents permuted generating matrix.

5.2.4 Hard Decision Decoding

Table 5.3 shows systematic matrix which is obtained by using elementary row operation. Hard decoding of first four bit of ordered received symbols which is assumed as the required information bits.vv_2 represents the hard decoding of received symbol. In the hard decision decoding, value zero is taken as the threshold value and compare the given value with the threshold value. The value which is greater than the zero, it is taken as 1 and the value which is less than the zero taken as 0. vv_2 is multiplied with permuted generating matrix which produce first best codeword. cc_2 represents the first best codeword.

5.2.5 Mass Calculation Analysis

Reliability Based Deciding decodes the codeword which may or may not be required codeword. So, algorithm consider all the 2^k test error patterns and their corresponding codeword .It is considered all the test error patterns and find the Euclidean distance between received symbols and all possible codeword. The mass can be calculate by using equation 3.4 and obtained the required codeword which gives maximum value of mass. The codeword which gives the value of maximum mass can be declared as a best minimum Euclidean distance. Table 5.5 shows the all the values of mass of possible codewords and also shows the obtaining of transmitted bits. M and vt2 denotes the mass and the original transmitted bits respectively.

5.2.6 Analysis of E_2 and E_3

The empirical and analytical pdf of E_3 validating the expression 4.60 assuring a well-known binomial weight distribution of primitive binary BCH codes. Analytically, in binomial pdf of length n, it has no relation with E_b/N_0. It also proves from figure 5.1 that the distribution of E_3 is also independent of TEP order. Thus for BCH (128, 64, 22) code, the distribution of E_3 can be assumed to be constant. Figure 5.3 shows the probability density function of E_2 at 0dB and 3dB. Figure 5.3 shows that the pdf E_2 shift towards zero as E_b/N_0 increases.

Figure 5.4 shows the Pdf plot of E_2 at 0dB and 3dB and E_3 at 0dB. If the value of E_b/N_0 is increased that means the pdf of E_2 and E_3 can be well separated based on some predefined threshold.

The distribution of E_2 and E_3 depicted in figure 5.2, that they can be well separated based on some predefined threshold. The plots of empirical and the analytical pdf of E_2 and E_3 validating the expression (4.50) and (4.60) respectively.

The distribution of E_2 and E_3 depicted in figure 5.2, that they can be well separated based on some predefined threshold. Test statistic E_T is compared against a predefined threshold T in order to decide if $\bar{c}^*$a true permuted codeword is or not. Based on the test if $E_T > T$, decision in favor of H_0 is done, thus assumed a true permuted codeword is not yet found and the reprocessing algorithm goes on with the update TEP. Otherwise, if $E_T < T$, decision in favor of H_1 is done, thus assumed a true permuted codeword is found and the reprocessing algorithm stops. It is clear that predefined threshold measures the extent of TEP reduction.

Figure 5.3 shows the probability function of two random variable under different E_b/N_0. It is clear that the pdf of E_2 shift towards zero as E_b/N_0 increases, which means the number of error in the tail decreases with the increase in the E_b/N_0. But most peculiarly, the pdf of E_2 is almost constant and is unaffected by the change in the E_b/N_0. This noise independence characteristic of E_3 can be the starting point of the reduction in the number of TEP.

5.2.7 Performance and Complexity Trade-Off Based on GPTT Analysis

ROC stands for Receiver Operating Characteristics which plot the graph between probability of false alarm (P_F) and the probability of detection (P_D).The test performance can be evaluated by the well-known ROC parameters where probability of false alarm and the probability of detection can be evaluated. Figure 5.5-5.9 shows the probability of detection when probability of false alarm is $10^{-7}, 10^{-6}$ and 10^{-5} with different SNR.

5.2.8 Code error rate of OSD Analysis

Figure 5.10 depicts the error performance of the (128, 64, 22) extended BCH code. This plot includes the simulation results and the corresponding upper bounds computed from maximum likelihood performance. For constant E_b/N_o, if the order of OSD is increased then the code error rate should be decreased. Also, for constant order, when the value of E_b/N_o is increased then the code error rate is decreased. The maximum code error rate is given by MLD.

5.2.9 Computational Saving Analysis

Figure 5.12 shows the number of cases in percentage where all the TEPs need not be processed at different SNR for different probability of false alarm. When the probability of false alarm is 10^{-5} at 2dB SNR, number of cases in percentage where all the TEPs need not be processed is 20. That means GPTT algorithm saves 20 percent TEP generation which reduces the complexity of decoding. When the probability of false alarm is 10^{-5} at 2dB SNR, number of cases in percentage where all the TEPs need not be processed is 80. That means that GPTT algorithm saves 80 percent TEP need not to check.

5.3 Comparison

5.3.1 Code Error Rate Comparison between OSD and GPTT-OSD

The code error rate performance of an (128, 64,22) extended BCH is presented in Figure 5.11 while using proposed GPTT decoding procedure and is compared with the MLD performance and also with OSD performance. The code error rate is plotted for varying False Alarm Probability of GPTT thresholding. Since the False Alarm of the GPTT decoding results in the increase in code error rate, it can be seen that for small False Alarm rate the performance of OSD-GPTT almost overlaps with the OSD performance for all range of E_b/N_o (for example, when the false alarm is 10^{-7}, see OSD (pink line) overlaps with OSD-GPTT (orange line) for all range of E_b/N_o). However, for fairly large False Alarm rate, the OSD-GPTT curves elevates with higher code error rate compared to the OSD performance. Thus, we can see that lowering the false

alarm will maintain the code error performance of the OSD, but we will show later that lowering the false alarm rate decreases the TEP reduction percentage.

5.3.2 TEPs Comparison between OSD and GPTT-OSD

Figure 5.12 shows the percentage of computational saving in TEPs using OSD-GPTT. Reliability Based Algorithm generates 2^k Test error patterns for k length of information bits. If the length of k is increased, the decoding complexity is also increased. RBA shows for $k/n \geq 0.5$,$d_{min}/4$ reprocessing order is sufficient for decoding the required codeword. GPTT-OSD is proposed to decrease the TEPs at each reprocessing order. Figure 5.11 depicts that for small false alarm rate the performance of OSD-GPTT almost overlaps with OSD performance for all range of E_b/N_o. And, figure 5.12 depicts that, for lower false alarm rate, the percentage of TEPs which need not to be checked is increased when increased in E_b/N_o.

CHAPTER SIX: CONCLUSION

6.1 Conclusion

Reliability Based Algorithm was basically implemented in two stage a) determining the Most Reliable Independent (MRI) bits from Most Reliable Basis (MRB) of the code and b) order-I reprocessing on MRI using most likely Test Error Patterns (TEPs). Order-I reprocessing is designed to improve the hard decision decoded codeword progressively until either practically optimum or a desired performance is achieved. The approach of ML resource test is based on the cost function calculated from the soft valued samples of permuted received sequence as a stopping criterion after each stage j, $0 \leq j \leq I$ of order-I reprocessing which indeed proved excellent in reducing the average number of computation. Despite its simplicity and efficient decoding capability for small and medium length codes ($n \leq 150$), the major weak point of the Reliability Based Algorithm are a) complex and loose theoretical error performance bound b) no stopping criterion can be achieved between that of order$-j$ and order$-(j-1)$ reprocessing stages.

A new statistics of the ordered vector component is presented. The probability Density Function (PDF) and the Cumulative Distribution Function (CDF) for each statistics is derived. The distribution of test statistics E_2 and E_3 can be well separated based on some predefined threshold which presents a stopping criterion after each TEP test thus largely cutting unnecessary TEP tests in higher level reprocessing.

The Reliability Based Algorithm is used 2^k Test Error Patterns for k information bit which increased the complexity for increasing the length of information bits. Generalized Probabilistic Threshold Test (GPTT) algorithm is presented which has the property of instant stopping criterion. GPTT algorithm reduces the 80% time that all the TEPs need not to be processed at each reprocessing order which reduces computational complexity at the expense of negligible performance degradation.

When the probability of false alarm is increased, the code error rate of OSD-GPTT dominates the performance of OSD. When the probability of false alarm increased, the number of cases in percentage where all the TEPs need not be processed is increased. So, the choice of probability of false alarm limit the performance of TEPs reduction.

In future enhancement, the application of new statistical approach which is described by E_1, E_2, E_3 can be applied to derive the further simplified error performance bound of other order statistics based algorithms for linear block codes like Generalized GMD and chase-type decoding.

REFERENCES

1. W. Cary Huffman and Vera Pless, *Fundamentals of Error Correcting Codes,* Cambridge University Press, New York, 2003.
2. Proakis, J .G, *Digital Communications,* McGraw-Hill, New York, 1995.
3. S. Lin and D. J. Costello, *Error Control Coding: Fundamentals and Applications*, 2nd edition. Pearson Education Inc., 2004.
4. J. K. Wolf, "Efficient maximum likelihood decoding of linear block code using a trellis, "*IEEE Trans. Inform. Theory,* vol. IT-24, pp. 76-80, Jan, 1978.
5. G. D. Forney, Jr., "Generalized minimum distance decoding," *IEEE Trans. Inform. Theory,* vol. 12, pp. 125–131, Apr. 1966.
6. D. Chase, "A class of algorithms for decoding block codes with channel measurement information," *IEEE Trans. Inform. Theory,* vol. 18, pp. 170–181, Jan. 1972.
7. M. Fossorier and S. Lin, "First order approximation of the ordered binary symmetric channel", *IEEE Trans. Inform. Theory*, vol. 42, pp. 1381-1387, Sept. 1996.
8. M. Fossorier and S. Lin, "Soft-decision decoding of linear block codes based on ordered statistics", *IEEE Trans. Inform. Theory*, vol. 41, pp. 1379-1396, Sept. 1995.
9. M. Fossorier and S. Lin, "Error performance analysis for reliability based decoding algorithms", *IEEE Trans. Inform. Theory*, vol. 48, pp. 287-293, Jan. 2002.
10. D. Gazelle and J. Snyders, "Reliability-based code search algorithms for maximum likelihood decoding of block codes", *IEEE Trans. Inform. Theory*, vol. 43, pp. 239-249, Jan. 1997.
11. M. Fossorier and S. Lin, "Computationally efficient soft decision decoding of linear block codes based on ordered statistics", *IEEE Trans. Inform. Theory,* vol. 42, pp. 738-750, May 1996.
12. A. Valembois and M. Fossorier, "An improved method to compute lists of binary vectors that optimize a given weight function with application to soft decision decoding", *IEEE Comm. Letters,* vol. 5, no. 11, pp. 456-458, Nov. 2001.

13. M. Fossorier, S. Lin, and J. Snyders, “Reliability-based syndromedecoding of linear block codes”, *IEEE Trans. Inf. Theory*, vol. 44, no.1, pp. 388-398, Jan. 1998.
14. Saif E. A. Alnawayseh and P. Loskot, “Complexity Reduction of Ordered Statistic Decoding Using Side Information”, *IEEE Comm. Letters*, vol. 16, no. 2, pp. 249-251, Nov. 2012.
15. M. Fossorier, “Complementary reliability-based decodings of binary linear block codes”, *IEEE Trans. Inf. Theory*, vol. 43, no. 5, pp. 667-672, Sep. 1997.
16. Y. Wu and C. Hadjicostis, “Soft-decision decoding of linear block codes using efficient iterative G-space encodings”, in *Proc. Globecom*, pp.921- 925, Nov. 2001.
17. M. Fossorier, “Reliability-based soft-decision decoding with iterative information set reduction”, *IEEE Trans. Inf. Theory*, vol. 48, no. 12, pp. 3101-3106, Dec. 2002.
18. A. Valembois and M. Fossorier, “Box and match techniques applied to soft decision decoding”, IEEE Trans. Inf. Theory, vol. 50, no. 5, pp. 796-810, May 2004.
19. W. Jin and M. Fossorier, “Enhanced box and match algorithm for reliability based soft decision decoding of linear block codes”, in *Proc. Globecom*, pp.1-6, Nov. 2006.
20. Y. Wu and C. N. Hadjicostis, “Soft-Decision Decoding Using Ordered Recodings on the Most Reliable Basis”, *IEEE Trans. Inf. Theory*, vol.53, no.2, pp.829-836, Feb. 2007
21. W. Jin and M. Fossorier, “Reliability-based soft decision decoding with multiple biases”, *IEEE Trans. Inf. Theory*, vol. 53, no. 1, pp. 105-120, Jan. 2007.
22. A. Kabat, F. Guilloud and R. Pyndiah, “New approach to order statistics decoding of long linear block codes”, in *Proc. Globecom*, pp.1467-1471, Nov. 2007.
23. X. Ma, J. Liu and B. Bai, “New techniques for upper-bounding the ML decoding performance of binary linear codes”, *IEEE Trans. Commun.*, vol. 61, no. 3, pp.842 -851, 2013.
24. A. R. Williamson, M. J, Marshall and R.D. Wesel, “Reliability Output Decoding of Tail-Biting Convolution Codes”, *IEEE Trans. Commun*, vol. 62, no. 6, pp.1768-1778, April 2014

APPENDICES

A. THESIS CODE

```
clc
clear all
close all
k = 4;
n = 7;
G = [1 0 0 0 1 1 0; ...
    0 1 0 0 1 0 1; ...
    0 0 1 0 0 1 1; ...
    0 0 0 1 1 1 1];
 pat = [0 0 0 0; 1 0 0 0; 0 1 0 0; 0 0 1 0; 0 0 0 1; 1 1 0 0; 1 0 1 0; 1 0 0 1; 0 1 1
0; 0 1 0 1; 0 0 1 1; 1 1 1 0; 1 1 0 1; 1 0 1 1; 0 1 1 1];

mass_count = [];
dec_num = 10;
iter = 1;
zer = eye(k);
ber_count1 = 0;
err = 1;
for iii = 1:iter
    dec_num = mod((dec_num + 1),16);
    Vt = dec2bin(dec_num, 4);
    tran_dat = mod(Vt*zer, 2);
    Ct = mod(Vt*G, 2);
    St = 2*Ct - 1;
    noise = sqrt(7/8)*randn(1,7);
    Sr = St + noise;
%    Sr = [2.0295   -0.4445   1.0804   -0.3953   -1.6944   0.0070   1.1987];
   rum = abs(Sr);
   ord = sort(rum, 'descend');
   perm = [];
   for i = 1:n
```

```
    perm = [perm find(rum == ord(i))];
    iperm(find(rum == ord(i))) = i;
end
perm;
iperm;
mat2 = [];
for i = 1:n
    mat2 = [mat2 G(:,perm(i))];
end

test = k + 1;
for i = 1:k

    while(mat2(i, i) == 0)
        flag = 1;
        count = i;
        while(flag)
            count = count + 1;
            if((count<(k+1)) && (mat2(count,i) == 1))
                mat2(i,:) = xor(mat2(i,:), mat2(count,:));
                flag = 0;
            else
                if((count > k) && (flag == 1))
                    swap = mat2(:,i);
                    mat2(:,i) = mat2(:, test);
                    mat2(:, test) = swap;
                    prm = perm(test);
                    perm(test) = perm(i);
                    perm(i) = prm;

                    test = test+1;
                    flag = 0;
                end
            end
```

```
        end
    end
    for j = 1:k
        if((i~=j) && (mat2(j,i) == 1))
            mat2(j,:) = xor(mat2(i,:), mat2(j,:));
        end
    end
end
mat2;
for i = 1:n
    for j = 1:n
        if (perm(j)==i)
            iperm(i)=j;
        end
    end
 end
S2_r = Sr(perm);
S2_r;
v_2 = (S2_r>0);
vv_2 =v_2(1:k);
vv_2;
cc_2 = mod(vv_2*mat2, 2);
cc_2    ;
mass = sum((2*cc_2 - 1).* S2_r);
mass;
massimo = 0;
numero_TEP = 15;
M = [];
for zz=1:numero_TEP

    for w=1:k
        pattern(w) = pat(zz,w);
    end
```

```
        for w =1:n
           h(w) = 0;
        end

        for j = 1:k
           if (pattern(j) == 1)
              for w =1:n
                 h(w) = xor(h(w),mat2(j,w));
              end
           end
        end

        mas = mass;

        for w = 1:n
           if h(w) == 1
              if cc_2(w) == 1
                 mas = mas - 2*S2_r(w);
              else
                 mas = mas + 2*S2_r(w);
              end
           end
        end
        M = [M;mas];
%
        if (mas > massimo)
           massimo = mas;
           for w = 1:n
              hh(w) = h(w);
           end

           for w =1:n
              gg_2(w) = xor(cc_2(w), hh(w));
           end
```

```
        end

    end
    vr = gg_2(iperm);
    vr2 = vr(1:k)    ;
    if(vr2 == tran_dat)
        ber_count1 = ber_count1 +1;
        err = 0;
    end

end
M
```