

Nested Encoding for Simple HDD and Study of Turbo Equalization

A Project Report

submitted by

SAIDAM DEEPAK

*in partial fulfilment of the requirements
for the award of the degree of*

MASTER OF TECHNOLOGY



**DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY MADRAS.**

JUNE 2016

THESIS CERTIFICATE

This is to certify that the thesis titled **Nested Encoding for simple HDD and study of Turbo Equalization**, submitted by **SAIDAM DEEPAK**, to the Indian Institute of Technology, Madras, for the award of the degree of **Master of Technology**, is a bonafide record of the research work done by him under my supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Dr. K.Giridhar
Project Guide
Professor
Dept. of Electrical Engineering
IIT-Madras, 600 036

Place: Chennai
Date: 04th May 2016

ACKNOWLEDGEMENTS

I would like to express my greatest gratitude for the people who have helped and supported me throughout my project. I am grateful to my guide, Dr. K Giridhar, for his continuous support. This project would not have been a success without his valuable suggestions and insights.

I would like to thank all the faculties in the Department of Electrical Engineering for teaching me with such patience which has helped me during my project work and in my studies.

I would like to thank my family, I am indebted to them forever for their never ending support and for giving me the freedom to make the decisions in my life.

Finally I would like to thank all my friends without whom my life would not be memorable.

ABSTRACT

For battery operated IoT devices, hardware intensive soft decision turbo decoding may not be a suitable choice. With a simple scheme like Matrix Parity Check Code (MPCC), a 2D parity matrix check code where data is supported by Cyclic Redundancy Check (CRC) and Parity and corrections are done using them, would help the IoT devices as this scheme is power efficient and low complex. This is joined with a hard decision decoder as the decoder also considers CRC and parity. The work on MPCC gives an interesting result where a simple and flexible FEC scheme like MPCC which beats out many other sophisticated channel encoding schemes for small block lengths.

In a low Pilot density OFDM system, Virtual Pilots can be formed using Turbo Decoder. The Turbo Decoder based decision feedback equalizer helps in improving the channel estimation when pilot density is very less. With proper feedback of the decisions, the throughput of the system increases as the number of pilots will be low. Also in the interference limited cases such a scheme would help in better channel estimate and thereby improving the BER performance of the desired user.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	i
ABSTRACT	ii
LIST OF TABLES	iv
LIST OF FIGURES	vi
ABBREVIATIONS	vi
1 INTRODUCTION	1
2 Matrix Parity Check Code	3
2.1 Introduction	3
2.2 Different Possible Error Patterns for 2 and 3 bit errors	6
2.2.1 Different possible 2 bit error patterns	6
2.2.2 Different possible 3 bit error patterns	7
2.3 Error Detection and Correction Capability of MPCC	9
2.3.1 Selection of CRC polynomial	9
2.4 Transmitter and Receiver	10
2.4.1 Encoder	10
2.4.2 Receiver	11
2.5 Complexity and Coding Gain of MPCC	12
2.5.1 Complexity Analysis of MPCC	12
2.5.2 Code Rate	13
3 Decision Feedback Equalizer for Mossaic	14
3.1 OFDM Symbol Structure	15
3.2 Transmitter and Receiver Block Diagram	17
3.2.1 Transmitter:	17
3.2.2 Receiver:	17

3.3	Channel Estimation	18
3.4	Equalization	21
3.5	LLR Computation	21
3.6	LLR Feedback	24
3.6.1	Mean Thresholding	25
3.6.2	Median Thresholding	26
3.6.3	Using All LLRs in decision feedback	26
4	SIMULATION RESULTS	27
4.1	Simulation parameters for MPCC	27
4.2	Ber vs SNR for AWGN and Rayleigh channels	27
4.2.1	MPCC concatenated with LDPC	29
4.3	Simulation parameters for DFE Using Turbo Decoder	30
5	Conclusions and future work	35
5.1	Conclusions	35
5.1.1	MPCC	35
5.1.2	DFE using Turbo	35
5.2	Future Work	36
5.2.1	MPCC	36
5.2.2	DFE using Turbo	36

LIST OF FIGURES

2.1	Turbo encoded code block with MPCC as Systematic bits	4
2.2	Matrix Parity Check Code Structure	5
2.3	Single bit Error Detection	6
2.4	Double bit Error Detection	7
2.5	All possible 2 bit error patterns	8
2.6	3 bit error patterns	8
2.7	3 bit error pattern mistaken as single bit error.	10
2.8	Transmitter Block Diagram.	11
2.9	Receiver Block Diagram.	11
3.1	Two Transmitter and Receiver Architecture of Mossaic	15
3.2	OFDM Symbol Structure for Turbo Encoded Bits	16
3.3	OFDM Transmitter Chain	17
3.4	OFDM Receiver Chain	18
3.5	Gray Coded 16 Qam	24
4.1	Ber of MPCC over AWGN	28
4.2	Ber of MPCC over Rayleigh	28
4.3	Probability of Undetected Block Error Rate	29
4.4	BER of MPCC-LDPC	30
4.5	BER of MPCC-LDPC	31
4.6	MSE for Pilot spacing of 10 with Mean Thresholding and all LLR feed-back	32
4.7	MSE for Pilot spacing of 7 with Mean Thresholding and all LLR feed-back for PetB	32
4.8	MSE for Pilot spacing of 7 with Mean and Median Thresholding for 802.11g	33
4.9	MSE for Pilot spacing of 16 with Mean Thresholding and all LLR feed-back for 802.11g	33
4.10	MSE for Pilot spacing of 16 with Mean Thresholding and all LLR feed-back for PetB	34

ABBREVIATIONS

OFDM	Orthogonal Frequency Division Multiplexing
CRC	Cyclic Redundancy Check
AWGN	Additive White Gaussian Noise
MPCC	Matrix Parity Check Code
LLR	Log Likelihood Ratio
2D	Two dimensional
mLS	Modified Least Squares
FFT	Fast Fourier Transform
IFFT	Inverse Fast Fourier Transform

CHAPTER 1

INTRODUCTION

The project thesis is of two parts: MPCC and other is DFE using Turbo decoder. MPCC is FEC scheme suitable for IoT devices which require very low complexity and high battery life. Conventional FEC for down-link PDSCH in LTE is turbo code, which needs a complex parallel viterbi decoder at the UE end. This may not be best suited for devices like IoT with low computational ability. By introduction of $Data + CRC + parity$ the performance of the code improves. This is the scheme that is analysed. Though any mathematical proofs or results couldn't be derived, the simulation results were observed and reported.

Turbo Decoder based DFE is a DFE system which uses Turbo Decoder for feeding back decisions based on LLR. In a low pilot density OFDM system this kind of an Equalizer can be used to improve the channel estimate and thereby improving the performance of the system. With a strict criterion on the LLRs that is feedback we can always improve the performance of the system. The throughput of the system improves as we're allocating less pilots and more data. In an interference limited case the performance of the desired user can be improved in terms of better channel estimate and hence the BER performance. LLRs are feedback based on Mean, Median and also all the LLRs without any criterion were sent and these three systems were compared.

The outline of this project thesis is as follows. Chapter 2 describes the structure of MPCC and its decoding algorithm. Chapter 3 describes

the decision directed feedback using Turbo decoder, with different kinds of thresholding. Chapter 4 Simulation results are presented. In chapter 5 final conclusions are drawn.

CHAPTER 2

Matrix Parity Check Code

2.1 Introduction

Minimizing the operational power requirement is of at most importance while integrating Internet of Things(IoT) in 5G networks. The major step towards low power operations of IoT devices are connected mode DRX and extended DRX. Due to short/long sleep cycles, while the devices is being connected to the network, will improve the power budget of the IoT devices

Considering a specific class of battery operated, IoT applications which uses small block sizes for data transmission, a new FEC scheme is proposed for the downlink, which reduces the power requirement of FEC decoder at the device. As conventional FEC for downlink in LTE is turbo code which depends on CRC to control the number of turbo iterations in every code block. These early stopping methods using CRC improves the decoder power requirement. Now the proposal is to introduce the MPCC in between CRC and turbo parity bits, where MPCC(MPCC with CRC in it) fills the systematic portion. The proposed decoder can perform:

- Error detection and evaluation of error correction possibility.
- Error Correction

The structure of the turbo coded output is shown in Fig. 2.1 which consists of MPCC as systematic bits followed by the turbo parity bits.

The flexibility of the scheme is that we can use either MPCC which has very good error correcting and detecting capability or in case of very bad channel conditions we can switch to turbo decoder. Going to a turbo decoder is appreciated only when MPCC fails in correcting the error i.e in very bad channel conditions.

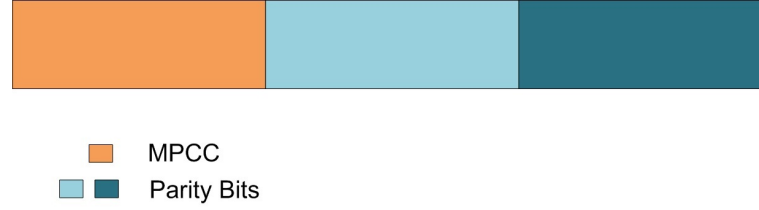


Figure 2.1: Turbo encoded code block with MPCC as Systematic bits

The structure of the Matrix Parity Check Code (MPCC) is shown in Fig. 2.2. MPCC is formed with 3 different kind of bits 1. Data bits, 2.CRC bits 3. Row and Column parity bits. Data bits are protected by both CRC and parity bits. CRC bits are calculated for the Data bits using traditional polynomial division. Parity bits are calculated using odd or even parity accordingly. MPCC designed here was 9×9 where 7×8 is data, appended by CRC row wise and then parity bits are calculated on CRC appended data matrix of size 8×8 .

It is not a strict requirement on the data matrix size of MPCC. We can still go to large/small block size but the problem here is that code rate of the structure changes and also complexity of the decoder changes. With small block size code rate decreases and with increased block size complexity of the decoder increases. Also, min distance (which is crucial for error detection and correction) of CRC polynomial which changes with changing block length. So, we need to trade off between code rate, coding gain, complexity and CRC for varying block length.

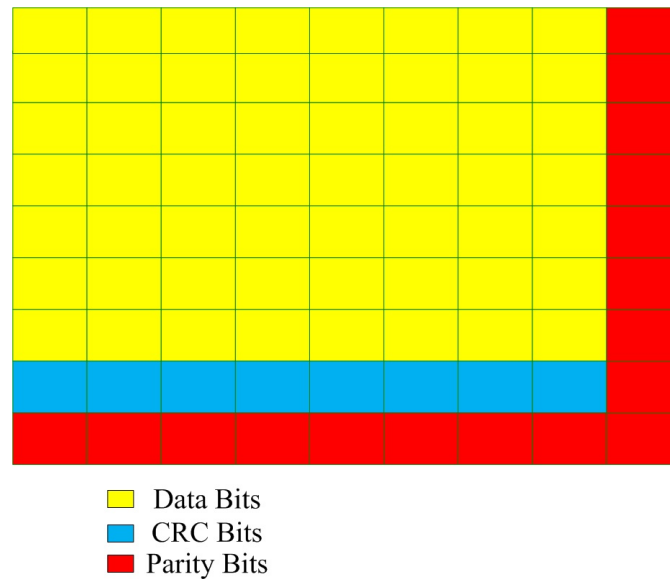


Figure 2.2: Matrix Parity Check Code Structure

The structure of the matrix itself helps in detecting the errors with the help of parity bits. Suppose that if there is an error at one location in the matrix code. This is shown up by the row and column parity bits. This is illustrated in Fig. 2.3, which shows the way in which parity bits can be helpful for detecting the errors in the MPCC. Though parity bits help in detection the error, its the CRC check that finally takes hold of the correction part. This is because of the ambiguity that Parity bits produce if there is a multiple bit error pattern(other than single bit error). This is shown in Fig. 2.4. Also, it can be directly observed that all one bit error patterns can be uniquely identified and can be corrected.

Now, in the case of Double or multiple bit errors the Parity bits can't uniquely identify the error location as in the case of single bit errors. The case shown in Fig. 2.4 is one of the possible cases of the occurrence of two bit errors. Here, though the parity bits detect two errors, they are not unique. With two Row and two Column Parity we can form two different sets of error locations(red and blue in Fig. 2.4) corresponding

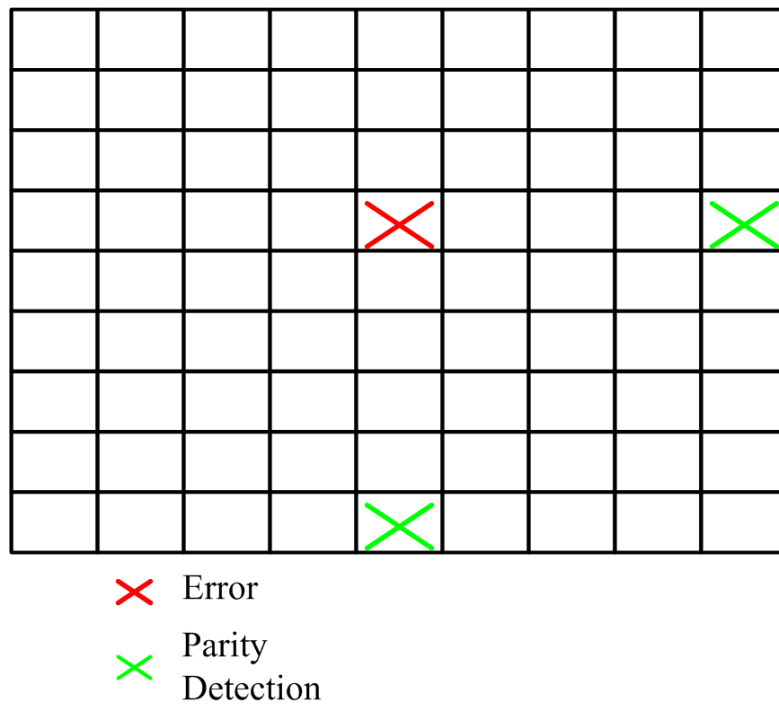


Figure 2.3: Single bit Error Detection

to different Row and Column Parity pair.

2.2 Different Possible Error Patterns for 2 and 3 bit errors

2.2.1 Different possible 2 bit error patterns

Different possibilities of multiple bit errors will give rise to different branches in error correcting algorithm, as each possibility of error pattern has a different way of correction, which will be explained later. All possible two bit error patterns are shown in Fig. 2.5. In the cases where two errors occur in row or column we've to search for the entire column or row respectively till we get a CRC Match. For the remaining case we've to search in 2 possible diagonal locations.

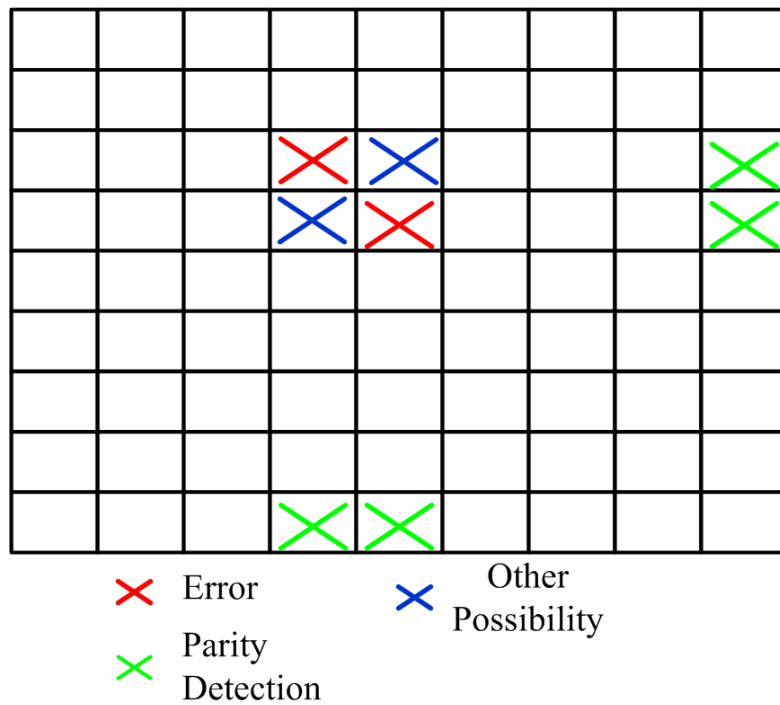


Figure 2.4: Double bit Error Detection

2.2.2 Different possible 3 bit error patterns

Many patterns of 3 bit errors are possible, but out of this only a few can be corrected reliably, but all 3 bit error patterns can be detected as CRC polynomial chosen in such a way that it is a multiple of $x+1$. The possible error patterns are shown in Fig. 2.6, here some combinations of two and single bit errors can be seen as 3 bit errors. Such combinations were not shown. Out of this only top left and bottom right sub figures are corrected as they are directly obvious. Ofcourse top right is taken as single bit error, but proper selection of CRC should solve this problem.

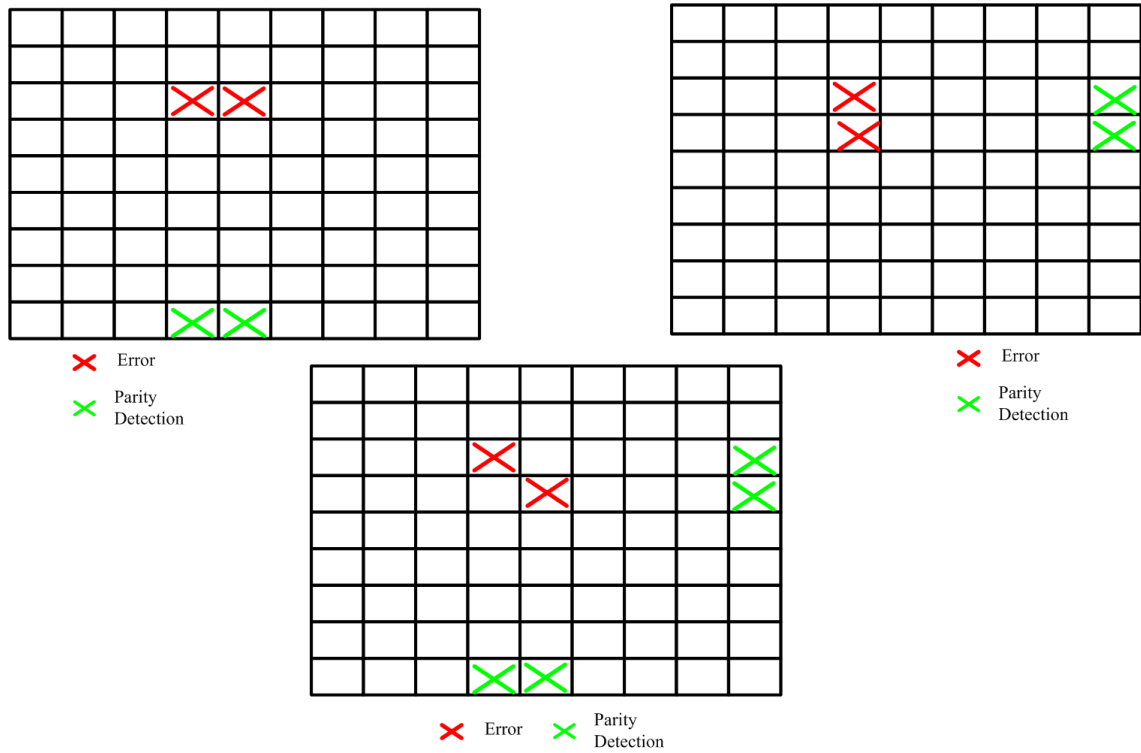


Figure 2.5: All possible 2 bit error patterns

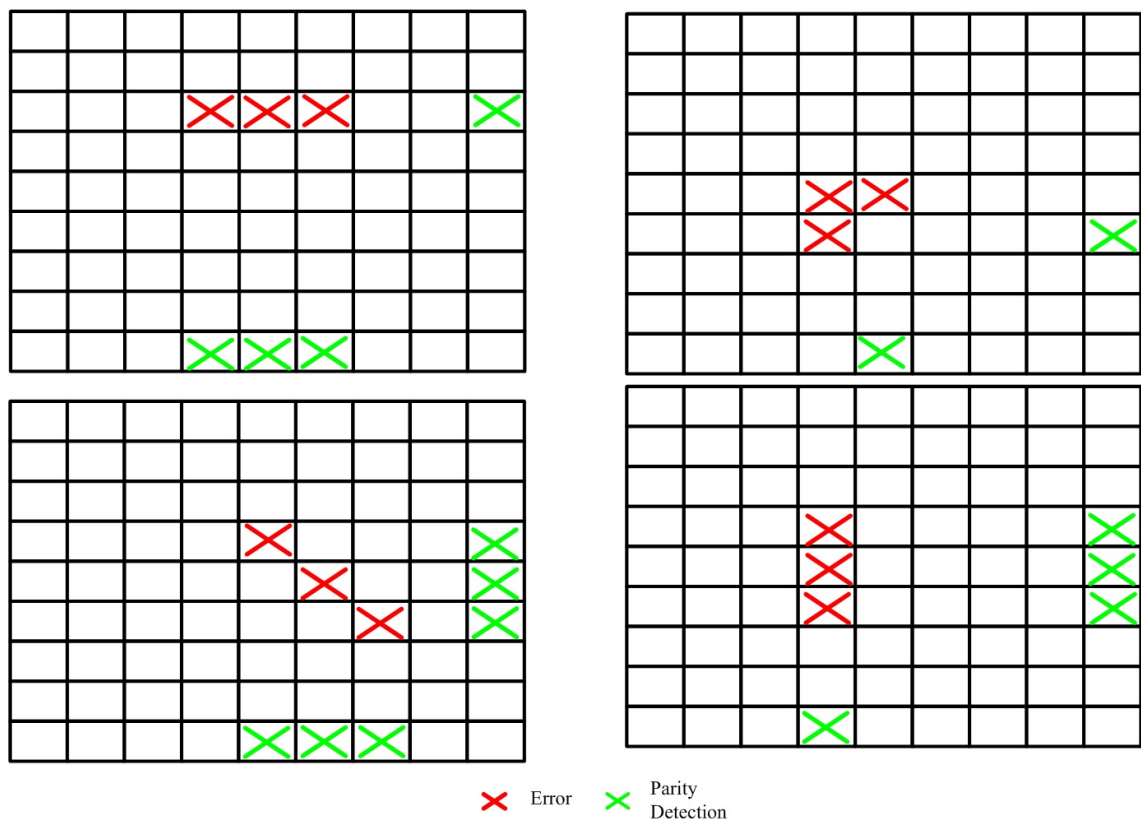


Figure 2.6: 3 bit error patterns

2.3 Error Detection and Correction Capability of MPCC

MPCC can detect errors using Row and Column parity bits, but in some cases as told earlier for 2 bit diagonal error pattern there are two different two bit error locations is possible. So here the CRC comes into play in deciding the which of two possible error locations are valid. So the selection of CRC generator polynomial accordingly is a vital factor for the desirable performance of MPCC.

2.3.1 Selection of CRC polynomial

As we can see some of the multiple bit error patterns can be taken as some other bit error pattern see Fig. 2.7. To avoid such cases, CRC polynomial should be selected such a way that Xor of all the corresponding bit's CRC shouldn't add to zero which then will be considered as no error which is false. CRC polynomial is chosen such that all the odd bit error patterns are detectable. For bit patterns like this case making sure that CRC condition in the equation is met is important and it avoids considering cases for higher bit error patterns. So, a careful selection of CRC polynomial is essential for our required matrix size. Inorder to avoid wrong error correction of Fig. 2.7

$$\begin{aligned} \text{CRC}(A,B,C,D) &= \text{CRC}(A) + \text{CRC}(B) + \text{CRC}(C) + \text{CRC}(D) \\ &\neq 0 \end{aligned} \tag{2.1}$$

With proper selection of CRC polynomial and parity bits the error detection and correction capability of the structure will increase. Also, as CRC is involved in error correction algorithm, it loses the inherent ca-

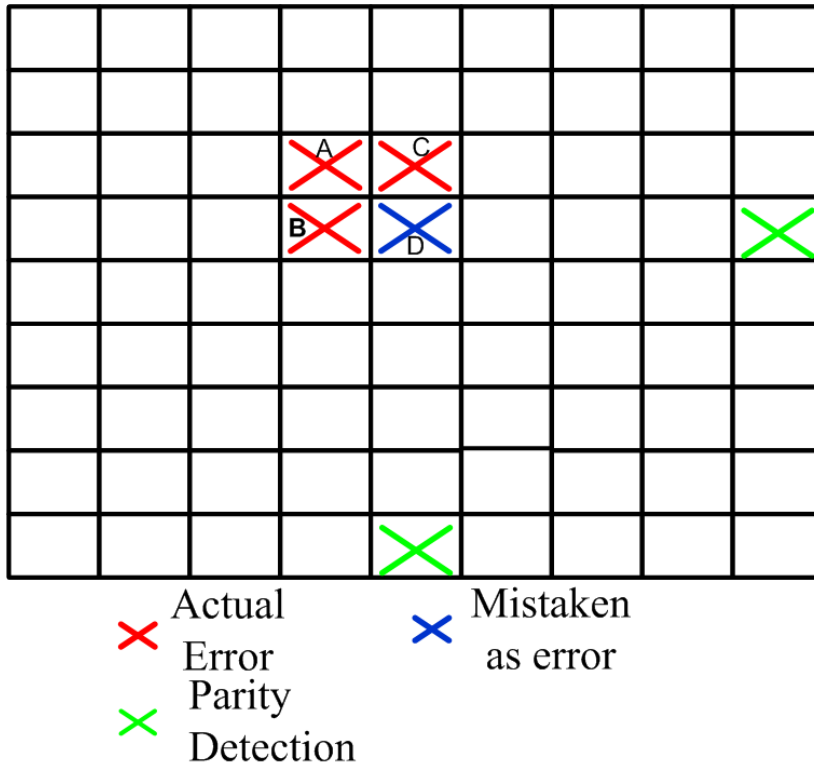


Figure 2.7: 3 bit error pattern mistaken as single bit error.

pability to detect errors in the code block. So, Undetected Block error rate of the structure is simulated which will be shown in the results.

2.4 Transmitter and Receiver

2.4.1 Encoder

Encoder for CRC+2D-parity check concatenated code is a CRC encoder followed by 2D-parity encoder. Hardware complexity increases with input block size. For a block size of 56 bits and CRC8(0x97) polynomial, encoder can be implemented using 53 XOR gates and a FIFO. The transmitter block diagram has been shown in Fig. 2.8

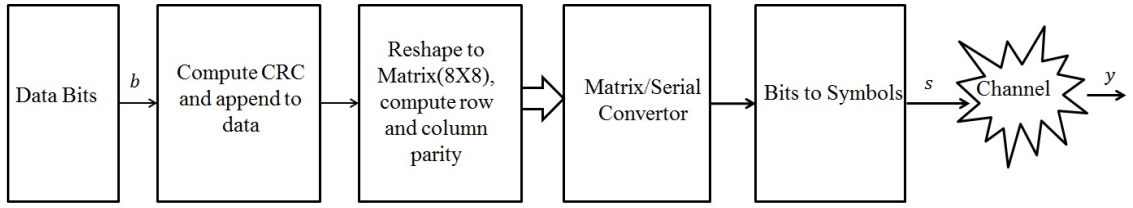


Figure 2.8: Transmitter Block Diagram.

2.4.2 Receiver

Receiver of MPCC has a hard decision decoder which makes decision directly on the received symbols. The block diagram of receiver is shown in Fig. 2.9

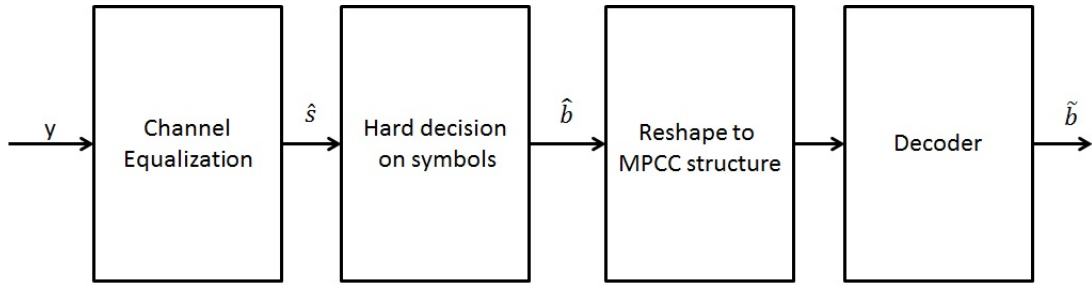


Figure 2.9: Receiver Block Diagram.

Decoding Algorithm

In order to reduce the computational complexity of the decoding process hard decision decoding has been implemented and also all the syndromes for Data bits are precomputed and use them as a look up table, which puts a requirement on the memory of the device. For feasible implementation of hard decision decoder, upto three bit error correction is done(few combinations of three bit error shown in Fig. 2.6)

Decoding Algorithm:

- Evaluate CRC syndrome for received Data stream, excluding parity and CRC bits.
- Evaluate Row and Column parity bits for received data, including CRC bits.
- Single Bit Correction: Perform correction only if row and column parity check indicates single bit error and the syndrome calculated in step 1 matches with the reference syndrome value from lookup table. This reference syndrome is read from the location pointed by parity check results.
- Two Bit Correction: Seven different type of two bit errors can occur in received data. Each type is uniquely specified by the row and column parity check results. For each type of two bit error, parity check result gives a small set of possible error locations. CRC syndrome matching as described in step 3 will provide the exact location of error bits.
- Three bits and beyond: Guaranteed error correction for three bits and beyond calls for larger CRC polynomials, smaller matrix sizes and hardware intensive decoding. But few typical three bit error patterns can be easily decoded using hard decision decoder, as mentioned in Two Bit Correction.

2.5 Complexity and Coding Gain of MPCC

2.5.1 Complexity Analysis of MPCC

The huge complexity of MPCC is due to CRC calculation that is to be done at every step to check whether right bits are corrected. CRC calculation has been made easy by storing all CRC of all single bit errors in Look up table. Whenever CRC is needed to be calculated for a certain pattern, it is required to just Xor all the respective position in the look up table. This way of calculating CRC reduces the complexity for single and multiple bit error patterns, but we can see the advantage of doing this

more in multiple bit error patterns.

The complexity in terms of number of additions and Memory(because of the look up table). All the calculations were done seperately upto 1bit, 2bit and partial 3 bit error patterns. These results were tabulated in the paper published.

2.5.2 Code Rate

For an $M \times N$ data matrix size the code rate of the MPCC structure is given by

$$CodeRate, R = \frac{MN}{MN + 2N + M + 2} \quad (2.2)$$

For a data matrix size of 7×8 , the code rate is 0.691

CHAPTER 3

Decision Feedback Equalizer for Mossaic

Mossiac: Multi Operator Simultaneously Shared Synchronised Air Interface for Communication, is a system where all the operators share same frequency and time. The scenario for two Operator with two mobile stations is shown in Fig. 3.1. The effect of interference is cancelled by using ML receiver, where modulation of all users and code rate of the desired UE is known. The main motivation of doing DFE is that, in Mossaic the pilspacing was 12 there by leading to very bad channel estimate on the desired user. At the UE we've symbols of all the interferers and also the desired user. If we do DFE and there by improve the channel estimation of the desired User, leading to better cancelling of the effect of desired User on recieved signal and there by improving the Symbol level LLR for the other interferers using ML. These symbol level LLR can be used in level-1 cooperation(D2D cooperation), improving the diversity of the other interfering channels by sharing. The modulation order is known by Multi-Utility Pilots.

$$y_{k,n} = h_{1,k,n}x_{1,k,n} + \sum_{m=1}^M h_{m,k,n}x_{m,k,n} \quad (3.1)$$

where:

m is the index of the interferer.

M is the number of interferers,

$x_{k,n}$ are the real-valued data symbols in subchannel k , transmitted at a time n .

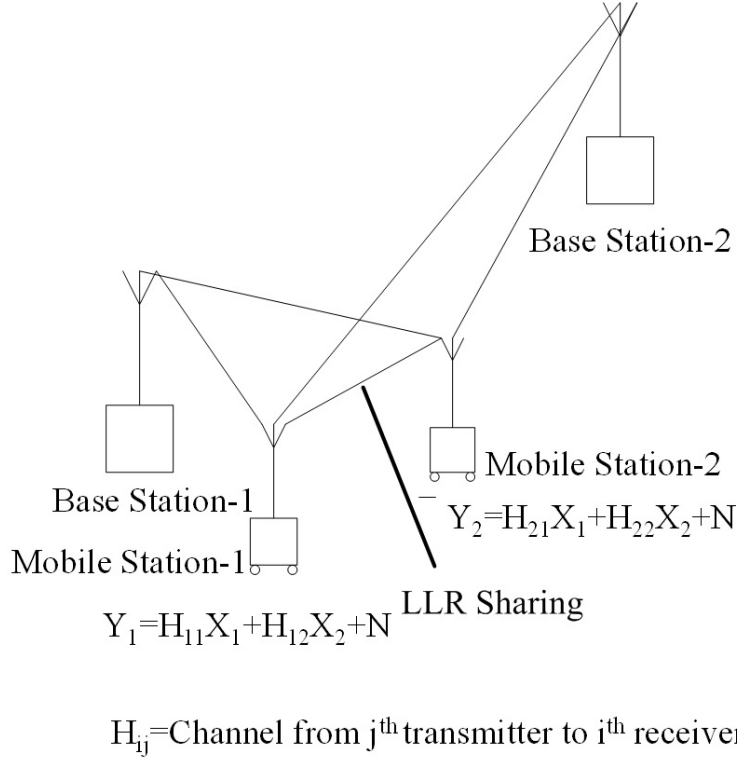


Figure 3.1: Two Transmitter and Receiver Architecture of Mossaic

$h_{k,n}$ are the rayleigh faded channel response in subchannel k , at a time n . For a system like MOSSAIC, where pilot spacing is low, Decision Feedback Equalization helps in improving the channel response of the system. Decisions used for improving the channel response are obtained from soft information of the turbo decoder. As we know the code rate and modulation of the desired user, turbo decoder soft information output can be used as the virtual pilots.

3.1 OFDM Symbol Structure

As turbo coding and decoding is used, information will be in systematic and parity bits. Let the symbols formed from systematic bits be called as Systematic Symbols and from the parity bits be called as Parity Symbols.

For a rate $1/r$ turbo code and for a block length of b bits we've rb bits. The Turbo decoder output will be soft bit information of systematic bits. Though there are ways to get the soft bit information for parity bits, it would increase the complexity of the turbo decoder. If all the Systematic Symbols and Parity Symbols are OFDM modulated as Systematic Symbols followed by Parity Symbols with pilots inserted, then we would only get a better estimate only on the portion of the OFDM Symbol containing only Systematic symbols. So, in order to get a better channel estimate the Systematic Symbols should be interleaved all over the OFDM Symbol so that virtual pilots can be all over the OFDM Symbol. The desired OFDM Symbol Structure is shown in Fig. 3.2

In Fig. 3.2 Systematic Symbols interleaved OFDM Symbol Structure is

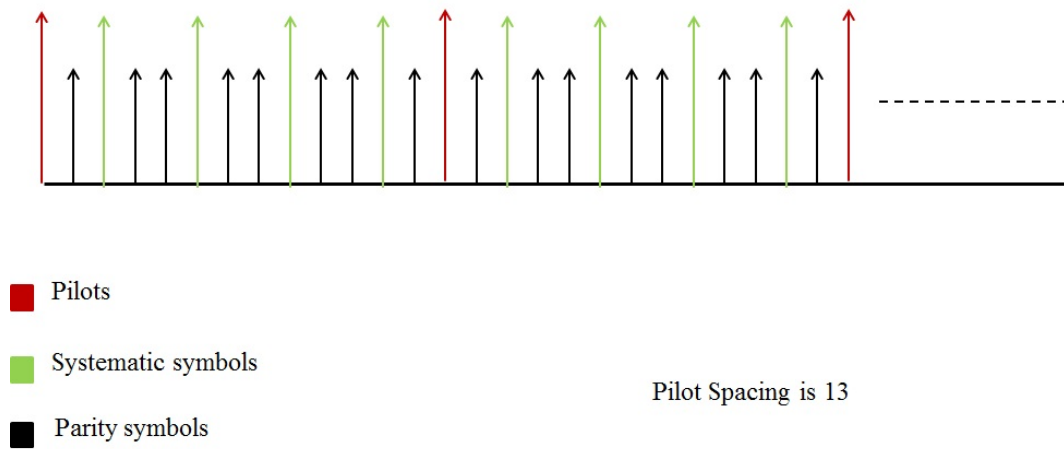


Figure 3.2: OFDM Symbol Structure for Turbo Encoded Bits

shown for a Pilot Spacing of 13. Though the subcarriers with Pilot and Systematic Symbols are shown with magnitude, that was just to discriminate the Parity symbols, Pilots and Virtual Pilots, but doesn't mean any power boost on Pilots and Symbols. The Structure can be varied accordingly depending upon the Pilot Spacing required.

3.2 Transmitter and Receiver Block Diagram

3.2.1 Transmitter:

The transmitter block diagram is shown in Fig. 3.3. This is similar to the general OFDM Transmitter Structure except for the Symbol Mapping block which inserts the Pilots at the required spacing and constructs the symbol according to the Symbol Structure discussed in the previous section. Symbol Mapper block maps the bits to a QAM constellation. For a pilot based channel estimation the number of pilots inserted be N_p and total data carriers be N_d . Therefore the total number of subcarriers in the system is $N_t = N_p + N_d$

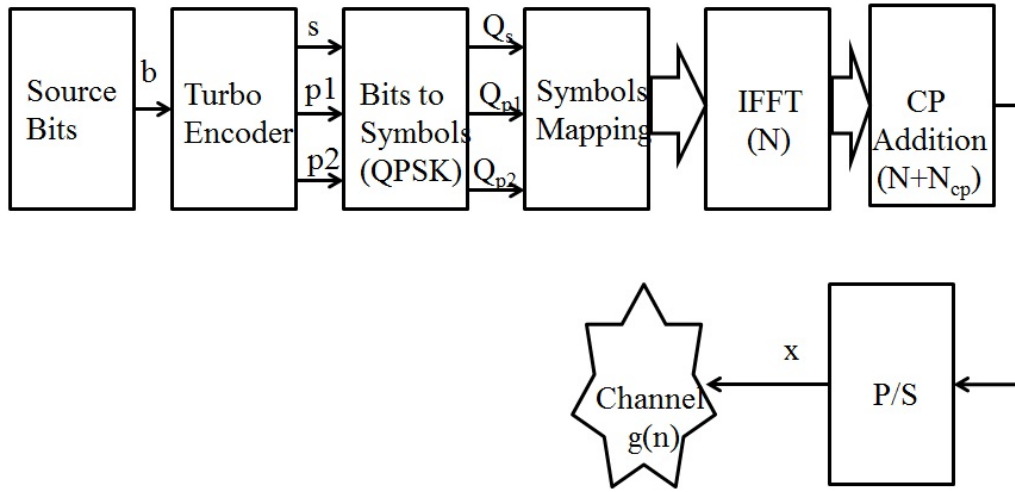


Figure 3.3: OFDM Transmitter Chain

3.2.2 Receiver:

The Receiver block diagram is shown in Fig. 3.4. The receiver is assumed to be perfectly synchronised to time and frequency. The Remapping Block in the receiver chain remaps the received OFDM Symbol

Structure such that all the systematic locations and parity locations are mapped back exactly as a group of systematic bit locations followed by parity bit locations. Other blocks like Channel Estimation and Bit level LLR will be explained in the next subsections.

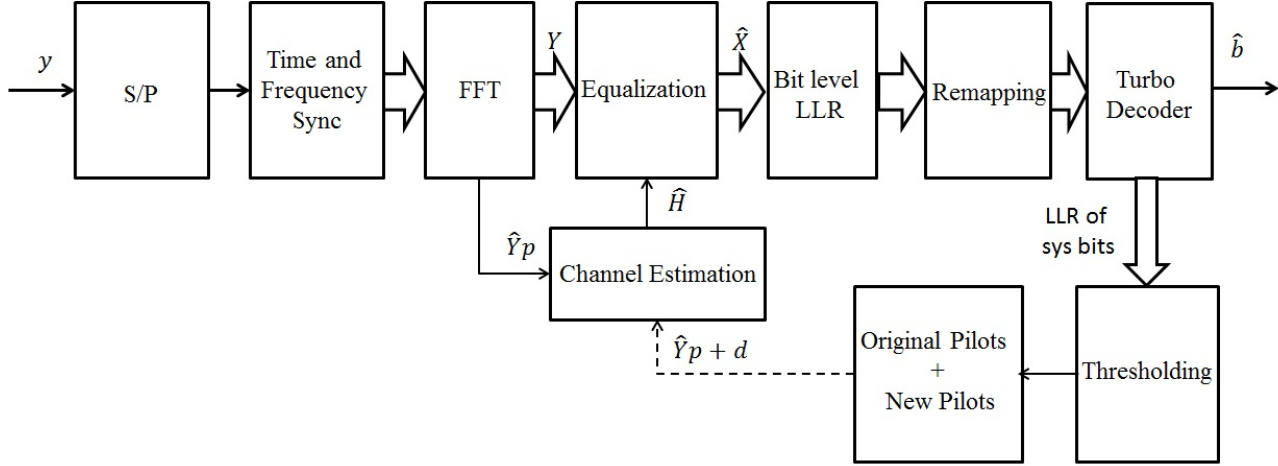


Figure 3.4: OFDM Receiver Chain

3.3 Channel Estimation

Pilot locations help in estimation of the Channel. Modified Least Squares(mLS) channel estimation is the technique that is used to estimate the Channel.

Let F be the FFT matrix of size $N \times N$ with entries as $\frac{1}{N}e^{-j\frac{2\pi kn}{N}}$ where $k = 0, 1, 2, 3, \dots, N-1$; $n = 0, 1, 2, 3, \dots, N-1$. The OFDM system is modelled using the N-point discrete-time Fourier transform (DFT_N) as

$$Y = F(F^H X \otimes g + n) \quad (3.2)$$

where X_k are the transmitted symbols, Y_k are the received symbols, n_t is the complex gaussian noise, g is the channel impulse response and \otimes denotes circular convolution. The system described above can be written as a set of N independent Gaussian channels

$$Y_k = H_k X_k + n_k \quad (3.3)$$

where h_k is the complex channel estimation given by $H = [H_0, H_1, H_2, \dots, H_{N-1}]^T = DFT_N(g) = Fg$ and $\mathbf{n} = [n_0, n_1, n_2, \dots, n_{N-1}]^T = DFT_N(n) = F\mathbf{n}$ is an i.i.d. complex zero-mean Gaussian noise vector. In matrix notation $Y = XFg + n$ where X is a matrix with the elements of X_k on its diagonal. The Least square (LS) estimator for the cyclic impulse response g minimizes $(Y - XFg)^H(Y - XFg)$ and generates

$$\hat{h}_{LS} = Q_{LS} F^H X^H Y \quad (3.4)$$

where $Q_{LS} = (F^H X^H X F)^{-1}$ so the LS estimate reduces to $\hat{h}_{LS} = X^{-1}Y$. For the case of mLS we only consider pilot locations. So Y in above equation turn to Y_p , X turns out to a diagonal matrix with symbols on pilot locations. $X_p = \text{diag}(\text{symbols on pilot locations})$, F matrix turns out to a modified dft matrix F_p where only those rows equal to pilot locations are kept and only columns upto the length of channel impulse response is kept. $F_p = e^{-j\frac{2\pi kn}{N}}$ where k is index on pilot locations and $n = 0, 1, 2, \dots, L - 1$. L is the impulse response length of the channel. So the modified Least Squares equation:

$$\hat{h}_{mLS} = Q_{mLS} F_p^H X_p^H Y_p \quad (3.5)$$

where $Q_{mLS} = (F_p^H X_p^H X_p F_p)^{-1}_{LXL}$, $F_p X_p$ is a $p \times p$ diagonal matrix. When the number of pilot locations are lower than the impulse response of the channel Q_{mLS} would turn out to a non invertible matrix. So regularisation of Q_{mLS} is required in cases when the pilots are lower than the impulse response length of channel. In order to regularise we add $\sigma^2 I$

where I is an identity matrix of size LXL

With regularisation the mLS estimate would be

$$\hat{h}_{mLS} = Q_{mLS} F_p^H X_p^H Y_p \quad (3.6)$$

where $Q_{mLS} = (F_p^H X_p^H X_p F_p + \sigma^2 I)_{LXL}^{-1}$, F_p is a $p \times L$ matrix, X_p is a $p \times p$ diagonal matrix. The mLS estimate is used in the initial estimation of the channel. When extra d virtual pilots from the turbo decoder are added the mLS estimate changes to

$$\hat{h}_{mLS} = Q_{mLS} F_{p+d}^H X_{p+d}^H Y_{p+d} \quad (3.7)$$

where $Q_{mLS} = (F_{p+d}^H X_{p+d}^H X_{p+d} F_{p+d} + \sigma^2 I)_{LXL}^{-1}$, F_{p+d} is a $(p+d) \times L$ matrix, X_{p+d} is a $(p+d) \times (p+d)$ matrix. The frequency domain response of the channel can be obtained by taking DFT of \hat{h}_{mLS} , $\hat{H}_{mLS} = F \hat{h}_{mLS}$.

The expression of \hat{h}_{mLS} is directly dependent on the number of pilot locations, implies that more are the number of equations (pilots) more better is the response and closer to the actual response. For a good estimate of the channel we at least need L pilot locations which is the impulse response of the channel. Here, feeding back of decisions based on LLR though an explicit way to do, making sure that wrong decisions were not feedback is most important, especially in cases of low SNR and also for higher modulation schemes where the performance of the turbo decoder is poor. This leads to error propagation especially in cases of higher modulation schemes and thereby leading to poor channel estimate.

3.4 Equalization

Equalization is an essential part of the processing at the receiver, necessary to take care of the multipath nature of the transmission channel. One of the big advantages of OFDM is that it allows simple per-subcarrier equalization.

Given that the estimated channel response is \hat{H} and Y is the received symbol after DFT. The equalisation of OFDM symbol is:

$$\begin{aligned}\hat{X}_k &= \frac{Y_k}{\hat{H}_k} \\ &= \frac{X_k H_k + N_k}{\hat{H}_k} \\ &= X_k \tilde{H}_k + \frac{N_k}{\hat{H}_k}\end{aligned}\tag{3.8}$$

Where k is the subcarrier index.

which is a zero forcing equalisation. When the extra pilots were added, still the equalisation procedure would remain the same, except for \hat{H} would be \hat{H}_{p+d}

3.5 LLR Computation

Log likelihood ratio is power technique in deciding wheter an estimate is good or bad given the channel conditions and SNR(signal to nois ratio). Always it is good to relay upon the magnitude of LLR. It is always good to keep the higher magnitude of LLR as they have low prob of bit being

in error. The definition of LLR:

$$\begin{aligned}
 LLR(b_i) &= \ln \frac{prob(b_i == 0)}{prob(b_i == 1)} \\
 &= \ln \frac{\sum_k e^{-\frac{\|y - hx_k\|^2}{2\sigma^2}}}{\sum_j e^{-\frac{\|y - hx_j\|^2}{2\sigma^2}}}
 \end{aligned} \tag{3.9}$$

where i indicates the bit index, x_k and x_j are the corresponding symbols for which bit b_i is 0 and 1 respectively.

LLR for BPSK:

For BPSK the LLR mapping is $\{-1, 1\}$ for $\{0, 1\}$. Substituting the values of the symbols back in the above expression and cancelling out some terms involving y^2 and x^2 the final expression for LLR is

$$LLR(b_i) = 4 * \frac{Real(y * conjugate(h))}{2\sigma^2} \tag{3.10}$$

LLR for grey coded QPSK:

For gray coded QPSK the LLR mapping is $\{-1-j, -1+j, 1-j, 1+j\}$ for $\{11, 10, 01, 00\}$. Substituting the values of symbols back in the above expression and cancelling out some terms involving y^2 and x^2 the final expression for LLR is

$$LLR(b_{1i}) = 4 * \frac{Imag(y * conjugate(h))}{2\sigma^2} \tag{3.11}$$

$$LLR(b_{2i}) = 4 * \frac{Real(y * conjugate(h))}{2\sigma^2} \tag{3.12}$$

where *real* and *imag* implies real and imaginary part of the complex number.

LLR for 16 grey coded Qam

For gray coded 16 Qam the constellation for which bit level llr are obtained is shown in Fig. 3.5. For a unit symbol energy on average, the constellation points should be divided by the average magnitude of the constellation energy, which is $\sqrt{10}$. The approximate way to generate LLR for rectangle constellation, M-Qam is to consider nearest symbol positions depending on the received y . By this kind of approximation the LLR equations are:

Soft bit for b_0 is:

$$\begin{aligned} LLR(b_0) &= 2(\text{real}(y) + 1) & \text{real}(y) < -2 \\ &= \text{real}(y) & -2 \leq \text{real}(y) < 2 \\ &= 2(\text{real}(y) - 1) & \text{real}(y) > 2 \end{aligned} \quad (3.13)$$

Soft bit for b_1 is:

$$\begin{aligned} LLR(b_1) &= \text{real}(y) - 2 & \text{real}(y) \leq 0 \\ &= -\text{real}(y) + 2 & \text{real}(y) > 0 \end{aligned} \quad (3.14)$$

Soft bit for b_2 is:

$$\begin{aligned} LLR(b_2) &= 2(\text{imag}(y) + 1) & \text{imag}(y) < -2 \\ &= \text{imag}(y) & -2 \leq \text{imag}(y) < 2 \\ &= 2(\text{imag}(y) - 1) & \text{imag}(y) > 2 \end{aligned} \quad (3.15)$$

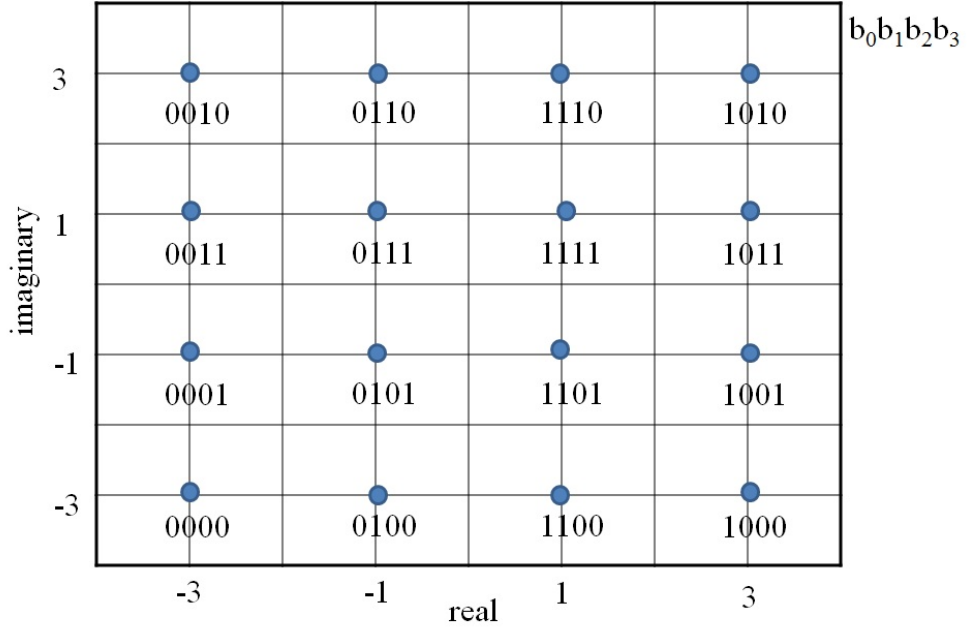


Figure 3.5: Gray Coded 16 Qam

Soft bit for b_3 is:

$$\begin{aligned}
 LLR(b_3) &= \text{imag}(y) - 2 & \text{imag}(y) \leq 0 \\
 &= -\text{imag}(y) + 2 & \text{imag}(y) > 0
 \end{aligned} \tag{3.16}$$

There is an extra scaling of noise term that is neglected. Also the model considered here calculation of LLR or soft bit is $y = x + n$ and for our model of channel where $y = hx + n$ proper scaling of terms is required. Also the decision regions change because of the energy scaling that is done in order to make the average symbol energy one. So all the decision regions and constants in the above equation will get a scaling of $1/\sqrt{10}$

3.6 LLR Feedback

Turbo Decoder gives bit level LLR for systematic bits. By using this LLR, symbols will be estimated so that they act as Virtual pilots or extra pilots. Symbols are estimated directly by hard decision decoding of LLR

and mapping them to $\{0,1\}$ accordingly, which are then given to modulator to get the symbol estimates for Channel Estimation with both pilots and Virtual Pilots(Systematic Symbols)

The performance of such Decision feedback system depends on the decisions that we feedback. This put a major constraint, which stops us from sending all the LLRs as some of them can be bad, depending on the SNR conditions. This effect of feeding back LLR with low quality is clearly seen in higher modulation schemes like 16 Qam or in the low SNR cases. The performance of dfe system over mean, median thresholding is Observed. Finally the results were compared against scheme where all the LLRs are used in the DFE.

3.6.1 Mean Thresholding

In mean thresholding, the LLRs were first seperated into positive and negative LLR. The mean for both the groups are calculated seperately, μ_1 and μ_2 be the postive and negative mean respectively. Then symbols are reconstructed in such a way that only if the consecutive LLRs are either greater μ_1 or less than μ_2 . With the consecutive locations of LLR symbols are estimated to be used in mLS algorithm. Consider $LLR_i, LLR_{i+1}, \dots, LLR_{i+\log_2 M}$ where M is the modulation order, only if

$$\mu_2 \leq LLR_i \leq \mu_1 \& \mu_2 \leq LLR_{i+1} \leq \mu_1 \& \dots \& \mu_2 \leq LLR_{i+\log_2 M} \leq \mu_1 \quad (3.17)$$

Where i goes from $i, i + M, i + 2M \dots$

3.6.2 Median Thresholding

Similar to mean, in this median thresholding is used by calculating median for both positive and negative LLR separately. Here too the decisions on the symbols is made for consecutive LLR. The equations for median thresholding would be similar to mean except for μ_1, μ_2 which are means would be changed to medians.

3.6.3 Using All LLRs in decision feedback

In this all the LLR that is outputted by the turbo decoder is used for the decision making and thereby all the Systematic symbols will be estimated. As the number pilot locations here when compared with the above cases are more, it is expected that the channel response would be better in high SNR. At low SNR as the turbo decoder performance is not that appreciable for higher order modulation schemes, when compared to lower order modulation schemes, this scheme of feeding all the LLRs for decision would lead to error propagation. This error propagation can be clearly seen in the 16 Qam for lower SNR which is shown in the simulation results chapter.

CHAPTER 4

SIMULATION RESULTS

4.1 Simulation parameters for MPCC

A Data Matrix size of 7×8 was chosen with a CRC polynomial of $CRC8(0X97)$. With QPSK Gray Coded modulation and even parity is chosen for transmission. The simulation was carried out for 20000 matrices for AWGN and Rayleigh channels.

4.2 Ber vs SNR for AWGN and Rayleigh channels

Fig. 4.1 shows the performance of MPCC over AWGN

Fig. 4.2 shows spectral performance of MPCC over Rayleigh Channel.

Fig. 4.3 shows the Probability of Undetected Block error rate for the chosen CRC polynomial which is $0x97$.

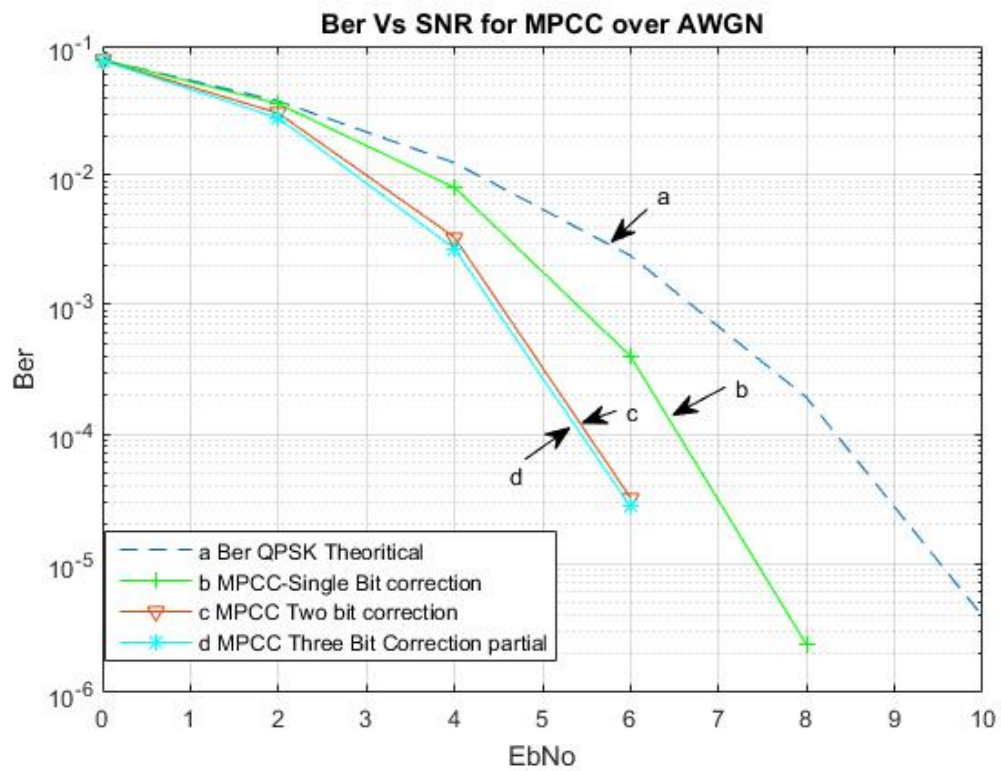


Figure 4.1: Ber of MPCC over AWGN

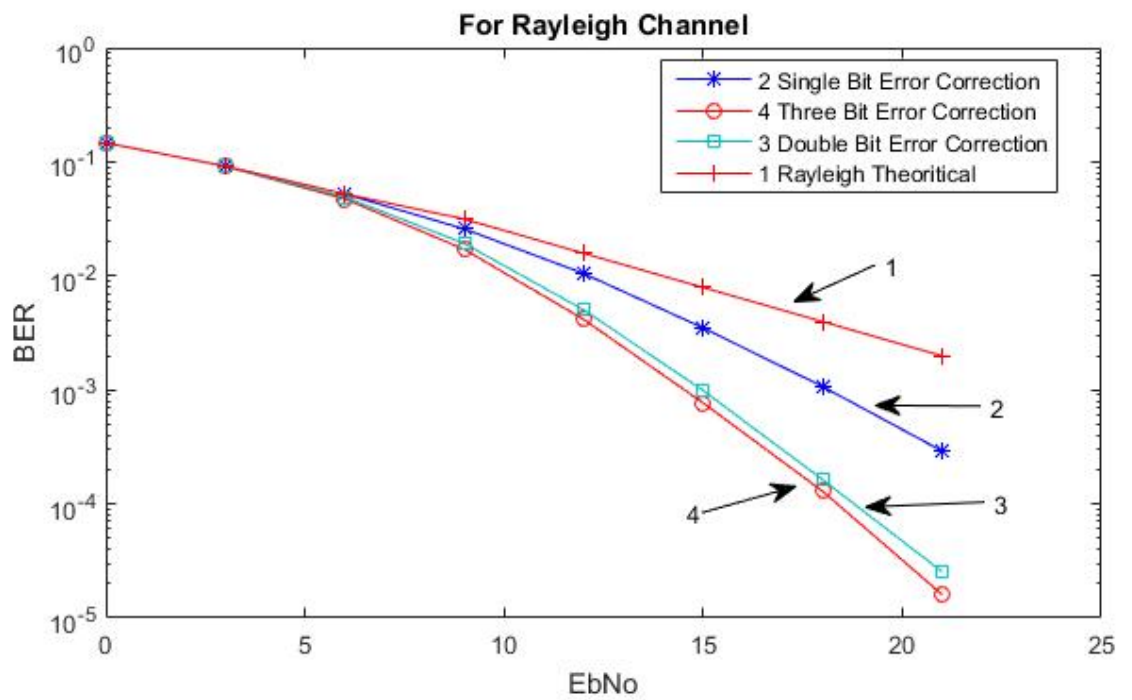


Figure 4.2: Ber of MPCC over Rayleigh

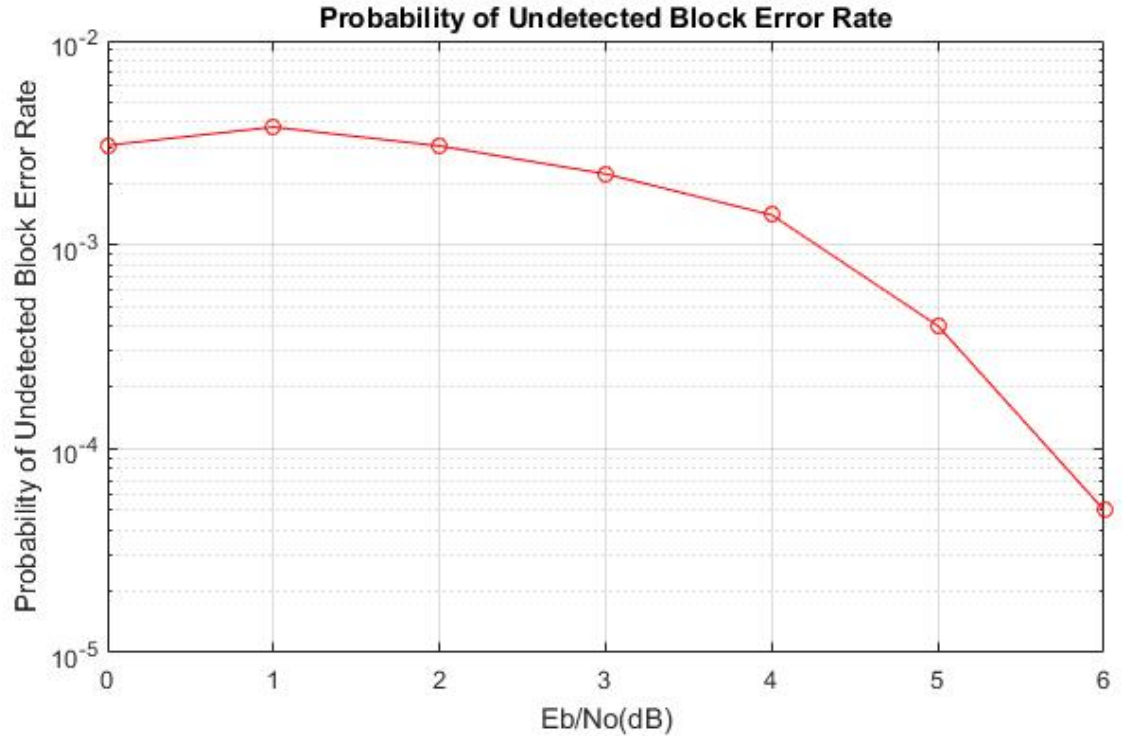


Figure 4.3: Probability of Undetected Block Error Rate

4.2.1 MPCC concatenated with LDPC

Iterative message passing codes are powerful at longer block lengths. But at a lower block length the gain due to iterative message passing is low. So, MPCC is made the inner code with LDPC code as the outer code, LDPC is run over 20 iterations and then MPCC decoding is applied.

Fig. 4.4 shows the Ber performance of MPCC-LDPC over AWGN.

Fig. 4.5 shows the Ber performance of MPCC-LDPC over rayleigh channel.

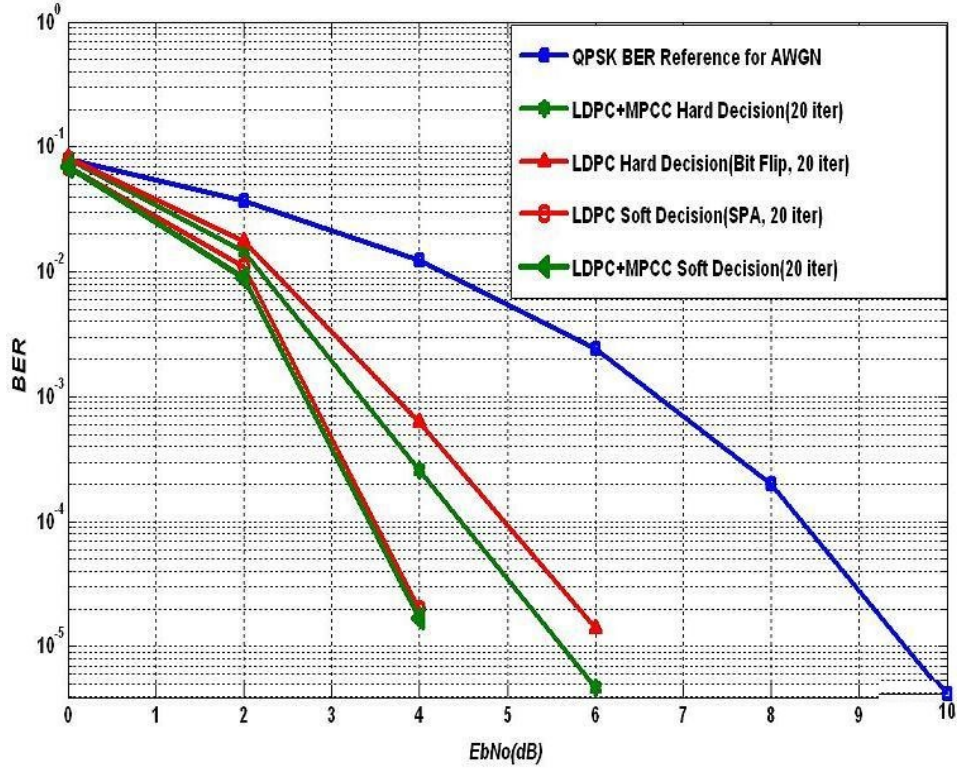


Figure 4.4: BER of MPCC-LDPC

4.3 Simulation parameters for DFE Using Turbo Decoder

The simulation is carried out in two parts 1. for 4Qam, 2.16Qam. For 4Qam the data size is 512 bits and for 16Qam it is 1024 bits. The Turbo Coder Constraint length is 5, with a convolutional code of $[13, 15], [13]$ as polynomials. Interleaver is 31, 64 for 512 and 1024 bits. Pilot spacing of 7 is shown for 16 Qam and pilot spacing of 16 is shown for 4Qam for both PedB and 802.11g channels with a bandwidth of $20MHz$ and Doppler frequency of 50Hz. Fig. 4.6 shows the MSE performance of 16 QAM with pilot spacing of 10 with Mean thresholding and all LLR's Feedback.

Fig. 4.7 shows the MSE performance of 16 QAM with pilot spacing of 7 over PetB with Mean thresholding and all LLR's Feedback.

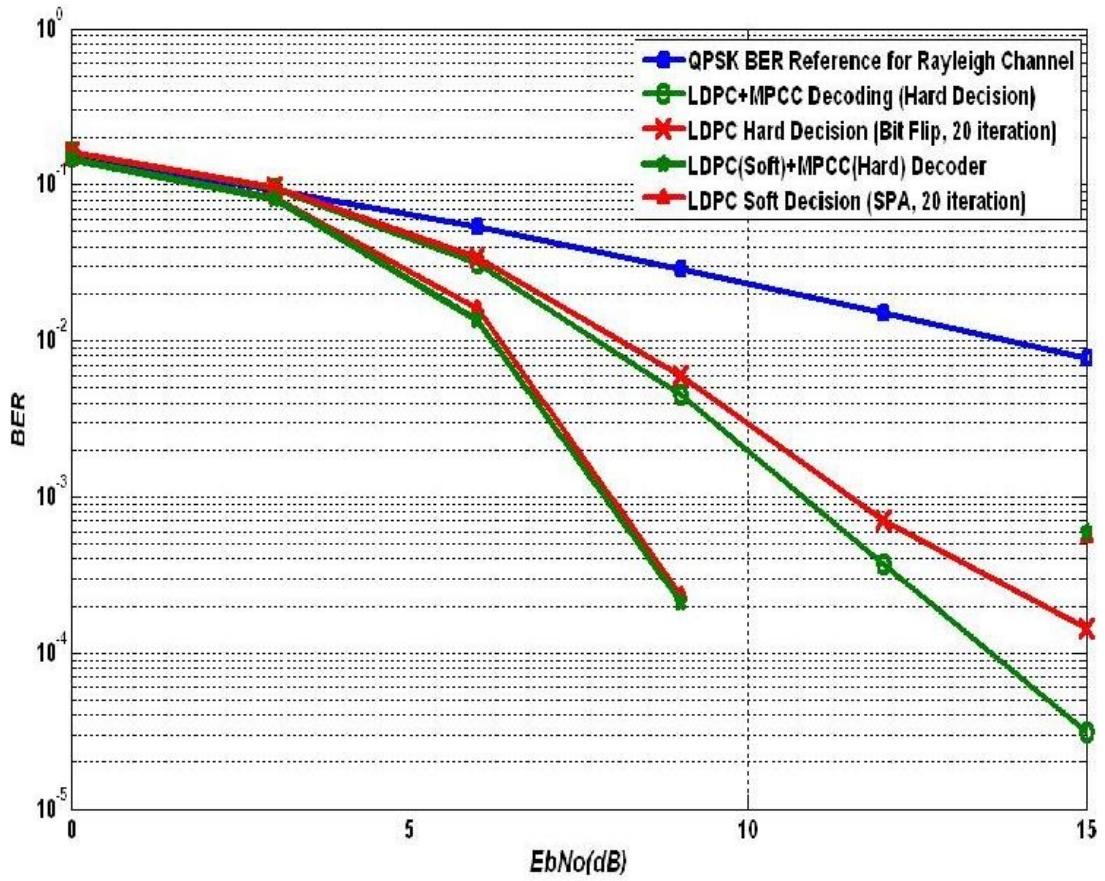


Figure 4.5: BER of MPCC-LDPC

Fig. 4.8 shows the MSE performance of 16 QAM with pilot spacing of 7 over PetB with Mean and Median thresholding.

Fig. 4.9 shows the MSE performance of 4 QAM with pilot spacing of 16 over 802.11g with all LLR's Feedback.

Fig. 4.10 shows the MSE performance of 4 QAM with pilot spacing of 16 over PetB with all LLR's Feedback.

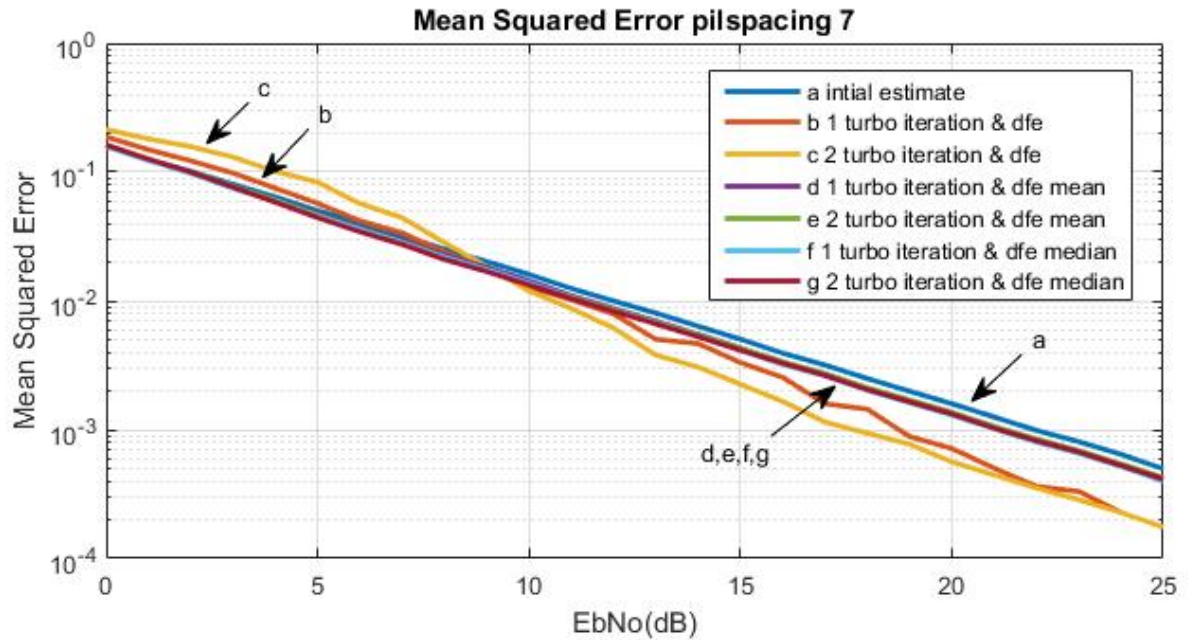


Figure 4.6: MSE for Pilot spacing of 10 with Mean Thresholding and all LLR feedback

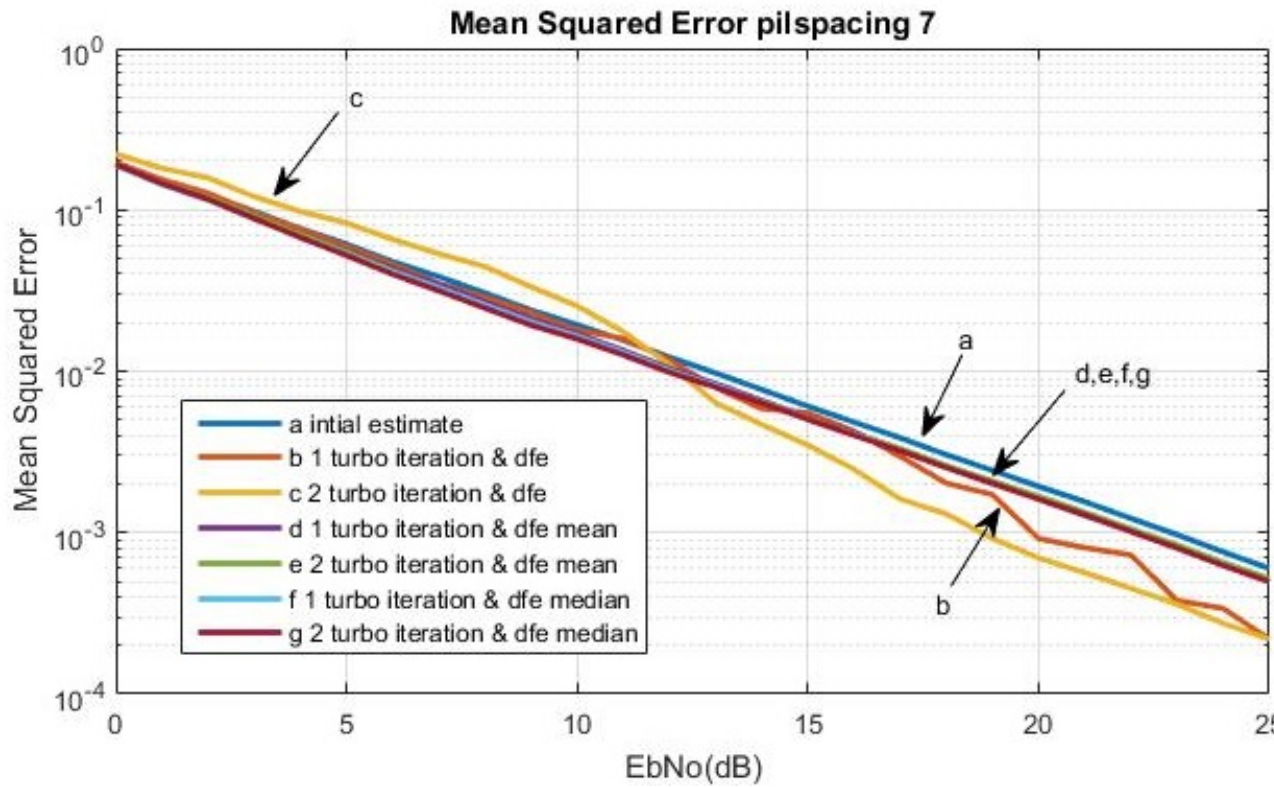


Figure 4.7: MSE for Pilot spacing of 7 with Mean Thresholding and all LLR feedback for PetB

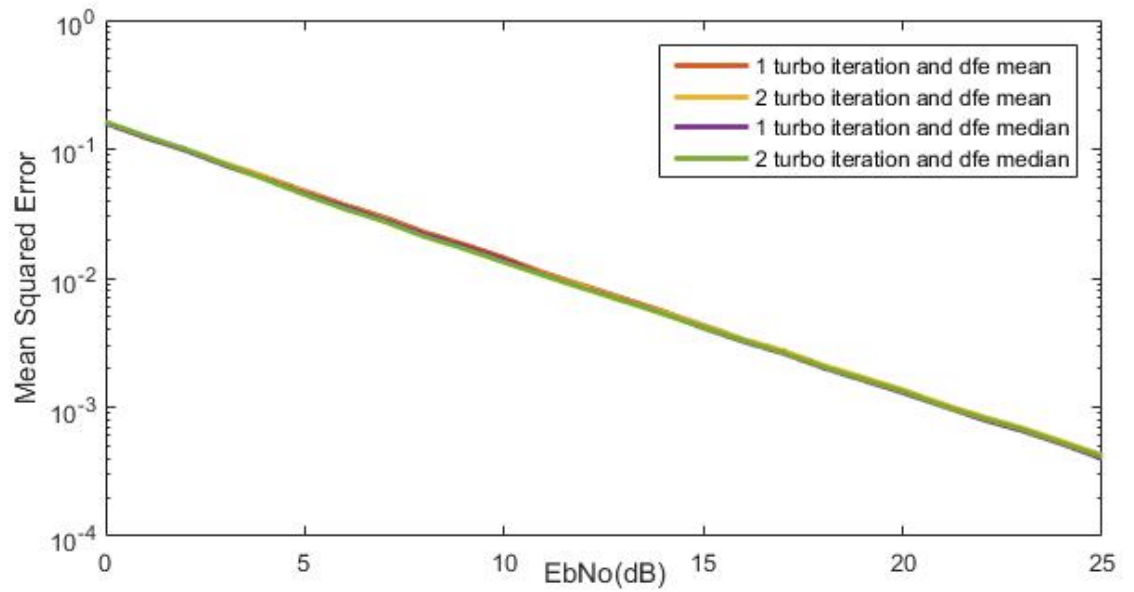


Figure 4.8: MSE for Pilot spacing of 7 with Mean and Median Thresholding for 802.11g

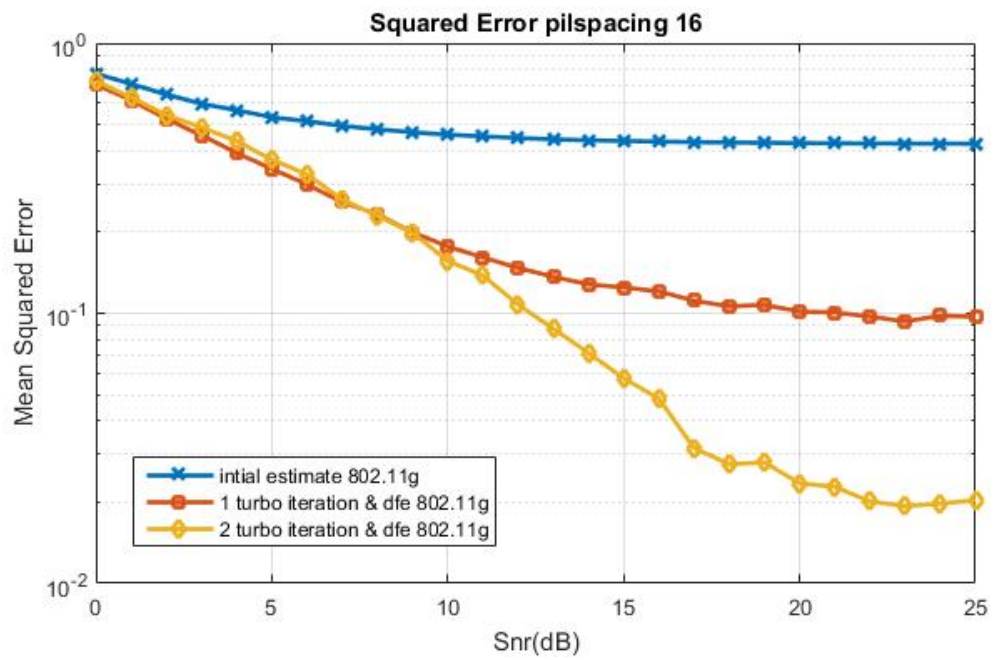


Figure 4.9: MSE for Pilot spacing of 16 with Mean Thresholding and all LLR feedback for 802.11g

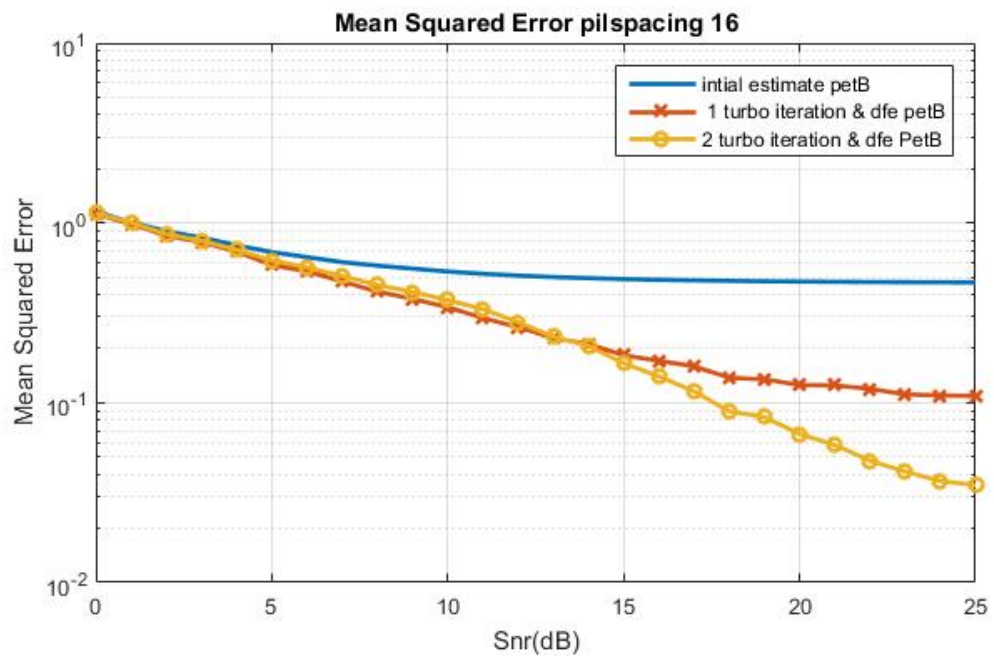


Figure 4.10: MSE for Pilot spacing of 16 with Mean Thresholding and all LLR feed-back for PetB

CHAPTER 5

Conclusions and future work

5.1 Conclusions

5.1.1 MPCC

Simple FEC like MPCC can beat out other sophisticated channel coding schemes for lower block length. The propose scheme is also very low complex and saves significant power. This will help the IoT devices gain in battery life.

5.1.2 DFE using Turbo

Turbo decoder performance depends on the initial channel estimate of the received signal. Also, the turbo decoder performance depends on the M-QAM that is transmitted. So, for a constellation like 16-QAM pilot spacing is crucial as the performance of decoder depends on both constellation and initial estimate of the channel. For the same pilot spacing the lower constellation gives a better performance gain when compared to the higher constellations. Also, error propagation is significant in higher constellations when compared to lower constellations. Applying schemes like threshold with mean and median have better performance when compared to all LLR feedback but the improvement is very small but there is

no error propagation in schemes like mean and median. Also, the performance of Mean and Median looks similar though Mean and Median may be equal because of the grouping of LLRs which form a symbol. This grouping may be the reason for which the performance of both Mean and Median looks similar. The improvement with all LLR feedback is more significant than mean and median threshold in high SNR cases. In case of 4-QAM the turbo decoder performance is better even in pilot spacing of 16. This improvement is because of the constellation is lower. So, for lower constellations it would be better to use all LLR feedback when compared to mean and median threshold. But for higher constellations it would be better to mean and median threshold at low SNRs and all LLR feedback for high SNR cases.

5.2 Future Work

5.2.1 MPCC

Soft decision Decoder for structure like MPCC. Mathematical framework to analyze the bounds on the BER performance.

5.2.2 DFE using Turbo

Schemes for better estimate in the low SNR cases so that there won't be error propagation. Implementation of schemes that would work in steps like that of Ransac. Finding the performance of these schemes in the presence of interferers.

References

1. Lei Zhou, Haibo Xu, Hui Tian, Youjun Gao, Lei Du and Lan Chen Performance Analysis of Power Saving Mechanism with Adjustable DRX Cycles in 3GPP LTE, Vehicular Technology Conference, 2008
2. Jung-Fu Cheng, Two-Level Early Stopping Algorithm for LTE Turbo Decoding, Vehicular Technology Conference, 2008
3. Hyeji Kim, Youngjoo Lee, Ji-Hoon Kim, Low-complexity CRCaided early stopping unit for parallel turbo decoder, Electronics Letters, Pages: 1660-1662, Volume: 51, Issue: 21, Year 2015
4. Philip Koopman, Tridib Chakravarty, Cyclic Redundancy Code (CRC) Polynomial Selection For Embedded Networks, The International Conference on Dependable Systems and Networks, DSN-2004
5. Yanbin Zhang, Qi Yuan, A Multiple Bits Error Correction Method Based on Cyclic Redundancy Check Codes, ICSP2008 Proceedings
6. CHEN Shi-yi, LI Yu-bai, Error Correcting Cyclic Redundancy Checks based on Confidence Declaration, 6th International Conference on ITS Telecommunications Proceedings -2006

LIST OF PAPERS BASED ON THESIS

1. Midhun Madhusoodanan, Saidam Deepak and K.Giridhar **Power Efficient Flexible FEC Scheme for 5G IoT Applications** WWRF36 conference 2016.