Modem design for DVB-S2

Theresh Babu Benguluri

A Thesis Submitted to
Indian Institute of Technology Hyderabad
In Partial Fulfillment of the Requirements for
The Degree of Master of Technology

Department of Electrical Engineering

Jun 2019

Declaration

I declare that this written submission represents my ideas in my own words, and where ideas or words of others have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the Institute and can also evoke penal action from the sources that have thus not been properly cited, or from whom proper permission has not been taken when needed.

B. Theresh Babu

(Theresh Babu Benguluri)

EE17MTECH11014

(Roll No.)

Approval Sheet

This Thesis entitled **Modem design for DVB-S2** by Theresh Babu Benguluri is approved for the degree of Master of Technology from IIT Hyderabad

R. Wandhare 24/06/19

Dr. Rupesh Wandhare
External Examiner
Dept. of Elec Eng
IITH

Dr. Abhinav Kumar
Internal Examiner
Dept. of Elec Eng
IITH

G.V.V. Sarma

Dr. G V V Sharma
Adviser
Dept. of Elec Eng
IITH

Dr. Abhinav Kumar
Chairman
Dept. of Elec Eng
IITH

Acknowledgments

There are many people that have supported, helped and encouraged me along the way and without them this thesis would not be possible. First of all, I would like to express my gratitude and thanks to my advisor **Prof G V V Sharma** for giving me full core master's thesis topic and for his excellent guidance and mentorship. All the various discussions including conceptual, theoritical and philosophical with Prof Sharma are very much valuable for my personal and professional development. I really enjoyed the days with Prof Sharma.

I am very much thankful to **Dr Lakshmi Prasad Natarajan**,for his valuable and fruitful discussions on LDPC Codes. Courses I've done under Dr Natarajan helped during my thesis.

I want to thank my Communication and Signal Processing faculty, **Dr Aditya Siripuram, Dr Abhinav Kumar, and Prof Sumohana Channappayya** for their excellency.

I want to thank my *family* and my best friend Raveendra for their belief and care.

I would like express my thanks and best wishes to Sandeep Kumar Khyalia,Siddharth Mourya,Sai Manasa, Raktim Goswami and Abhishek Bairagi for their contributions and help.I am very thankful to my lab mates Swaroopa Reddy, Hemanth Kumar,Tanmay Agarwal, Prasanna Kumar and Prakruti Bilagi and others for creating stress free, joyful and working environent.

I'm feeling very much grateful to my friends Swaroopa Reddy,Mubeen Abdul, Charan Tej, Subbareddy, Satish Reddy and Giridhar for care and fun. I really enjoyed my iith days.I want to thank all the IITH-Administration and Cleaning Department for their valuable services.

Finally I want to thank all the open source contributors for their developements of github,gitlab,python and Latex

Dedication

To My Lovely Family

For giving me so much care, motivation and love

Abstract

The Need of providing high quality video transmission services like Direct-To-Home services was becoming a hot topic in the area of wireless communication. This imposes a huge demand in terms of high data rate and high quality of service in satellite video transmission inflicting strict constraints on the power and bandwidth efficiencies of the Physical Layer (PHY-layer) wireless communication systems. Much more extensive research was going on to fulfil these requirements. DVB-S2 standard was one among them.

DVB-S2 standard is a 2003 version under the DVB project. In this, channel capacity has been achieved by employing LDPC codes as inner encoder and BCH codes as outer encoder. Which will meet the demands of channel capacity to near shannon limit. The total system of Channel encoder with BCH+LDPC coding, efficent mapping schemes of PSK and APSK, efficient transmission design techniques like Interleaving,Physica Layer Framing and Pulse shaping ,and a robust receiver synchronization block which will gives to hightly reliable communication over a noisy channel.

In this thesis, main focus is on the implementation of modulator, demodulator,Channel Encoder and Decoder techniques as independent modules in particular for each transmitter block suitable demodulator block which is identified in the DVB-S2 standard will be designed and verified using the numerous simulations.Throughout the whole thesis wherever possible, all the subblock equations have been optimized and the key equations taken from various standard papers for implementation have been given in more understandable format.

By using Berlekamp-Massey algorithm for BCH Decoding and standard BCH Encoding process, the perfomance of BCH codes was analyzed using SNR vs BER graph. For the case of LDPC codes message passing algorithm was used for decoding and standard encoding used , the performance of same is analyzed with SNR vs BER graph. The performance of PSK, APSK modulation schemes are analyzed using SNR vs BER graph simulations with an optimized way of demapping. The Efficient

Transmitter Design Techniques listed in the standard i.e Interleaver, Physical Layer Framing and Pulse shaping are analyzed with SNR vs BER graph by taking QPSK mapping scheme. In Receiver Synchronization, each synchronization block have been implemented independently with BPSK/QPSK mapping schemes and analyzed with SNR vs BER performance. Results of all subblocks have been verified using the standard document.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Thesis Organization

A quick review on DVB-S2 [1] and a glance on every block and its subblocks of the standard are presented in Chapter 1. Channel Coding Techniques which are employed in the standard and the simulation results are presented in Chapter 2. In Chapter 3,theoretical and simulation study on higher order mapping schemes i.e QPSK,8-PSK,16-APSK and 32-APSK are presented. Efficient Transmitter Design techniques which are specified in standard and its results are placed in Chapter 4. In Chapter 5, the robust receiver synchronization techniques and its modules results are presented. The future works possible and the conclusions upon this thesis are presented in Chapter 6.

1.2 A review on DVB-S2

In this in order to meets these demands powerful channel coding techniques must be used in conjunction with efficient modulation schemes like QPSK, 8-PSK, 16-APSK and 32-APSK. A powerful FEC system based on LDPC codes cascaded with BCH codes simultaneously meets these conflicting demands and can perform close to the theoretical limits of a communication system [2–5] Encoding and decoding modules for a family of LDPC codes, covering a range of code lengths and rates, will ensure reliable bandwidth-efficient communication, and will also facilitates adaptive modulation and coding.
Features of DVB-S2 standard,

1. Modulation: QPSK, 8PSK, 16APSK, 32APSK

2. Code rate: $\frac{1}{4}, \frac{1}{3}, \frac{2}{5}, \frac{1}{2}, \frac{3}{5}, \frac{2}{3}, \frac{3}{4}, \frac{3}{4}, \frac{4}{5}, \frac{5}{6}, \frac{8}{9}$ for short frames and the additional $\frac{9}{10}$ rate for normal frames and (as in DVB-S2). Rate can change on the fly.

3. Overall code length: Shall be 64800 for normal frames and 16200 for short frames. Code length can change on the fly. Programmable on block by block basis. Programmable code matrix.

4. Iterations: Automatic or user defined.

1.3 Transmitter

1.3.1 Block diagram

Our focussed block for transmitter is as follows,

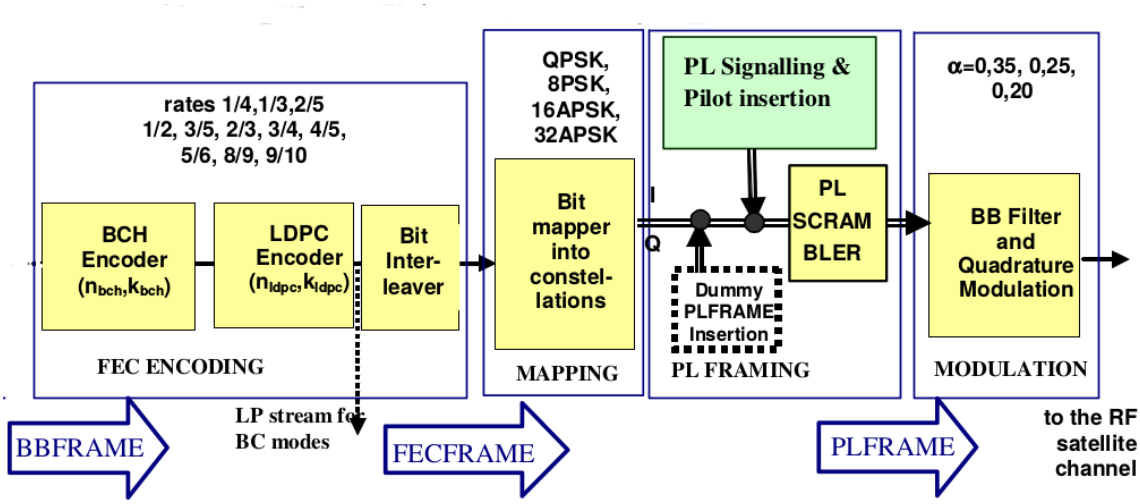


Figure 1.1: Transmitter Block Diagram of DVB-S2.

Input data stream will be went through mode adoptation and stream adoptation as specified in [1], the output of stream adoptation called Base-Band Frame (**BBFRAME**) will be given to FEC encoding block,the output of FEC encoding block is **FECFRAME** and given to MAPPING block which gives **XFECFRAME** . This **XFECFRAME** will given to Physical Layer(PL) FRAMING and the output frame is **PLXFECFRAME**. Through out this document we will follow this abbrevations.

1.3.2 FEC Encoding

Fig. 1.2, Fig.1.3 shows the Coding parameters of both LDPC, BCH for normal, short frame lengths.

Coding parameters (for normal FECFRAME n_{ldpc} = 64 800)

LDPC code	BCH Uncoded Block K_{bch}	BCH coded block N_{bch} LDPC Uncoded Block k_{ldpc}	BCH t-error correction	LDPC Coded Block n_{ldpc}
1/4	16 008	16 200	12	64 800
1/3	21 408	21 600	12	64 800
2/5	25 728	25 920	12	64 800
1/2	32 208	32 400	12	64 800
3/5	38 688	38 880	12	64 800
2/3	43 040	43 200	10	64 800
3/4	48 408	48 600	12	64 800
4/5	51 648	51 840	12	64 800
5/6	53 840	54 000	10	64 800
8/9	57 472	57 600	8	64 800
9/10	58 192	58 320	8	64 800

Figure 1.2: Coding parameters for Normal Frame

Coding parameters (for short FECFRAME n_{ldpc} = 16 200)

LDPC Code identifier	BCH Uncoded Block K_{bch}	BCH coded block N_{bch} LDPC Uncoded Block k_{ldpc}	BCH t-error correction	Effective LDPC Rate k_{ldpc}/16 200	LDPC Coded Block n_{ldpc}
1/4	3 072	3 240	12	1/5	16 200
1/3	5 232	5 400	12	1/3	16 200
2/5	6 312	6 480	12	2/5	16 200
1/2	7 032	7 200	12	4/9	16 200
3/5	9 552	9 720	12	3/5	16 200
2/3	10 632	10 800	12	2/3	16 200
3/4	11 712	11 880	12	11/15	16 200
4/5	12 432	12 600	12	7/9	16 200
5/6	13 152	13 320	12	37/45	16 200
8/9	14 232	14 400	12	8/9	16 200
9/10	NA	NA	NA	NA	NA

Figure 1.3: Coding parameters for Short Frame

- The Encoding process is done using cascaded BCH codes,LDPC codes as outer and inner layers of Forward Error Correction Encoding. FEC Encoder block (FECFRAME) has two block lengths, $N = 64\ 800$ bits and 16200 bits.
- Each BBFRAME (k_{bch} bits) given to FEC Encoder block, Which generates FECFRAME. BCHFEC ,is the parity bits block after sending to BCH Encoder will be concatenated after BBFRAME. LDPCFEC,is the parity bits block after

sending to LDPC encoder will be concatenated after BCHFEC field. [1]

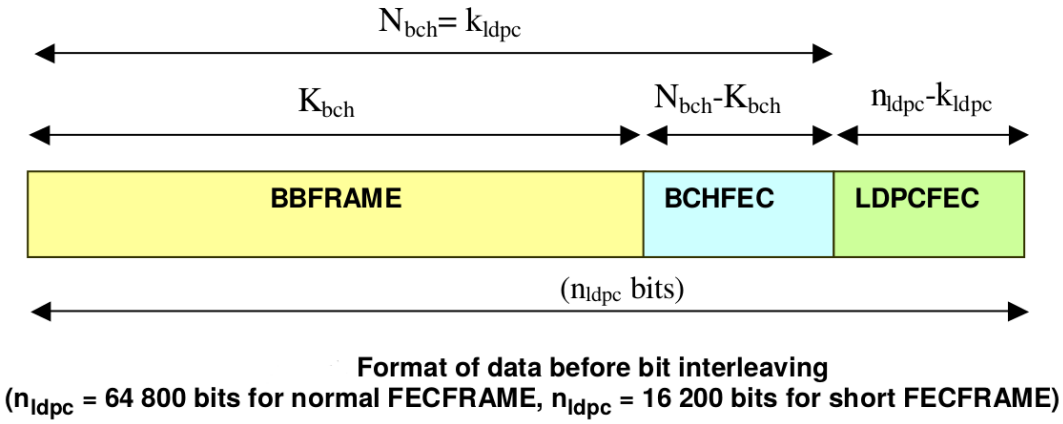


Figure 1.4: Format of data before bit interleaving

Outer Layer Encoding Process Using BCH codes

- After the BBFRAME generation by the source encoding process, which consists of Information bits is going to proceed towards BCH Encoder block.

- Suppose information bits from BBFRAME are, $m = (m_{k-1}, \ldots, m_1, m_0)$ and the Codeword after the BCH Encoder will have information bits concatenated with parity bits $c = [m|d]$ i.e $(m_{k-1}, \ldots, m_1, m_0, d_{n-k-1}, \ldots, d_1, d_0)$. Where n, k are the block length and information length of BCH Encoder. d denotes the parity bits vector.

- information bits polynomial and coderword polynomial will be decided by the coderate and frame length.

- A suitable generator polynomial will be decided by type of the frame.

Fig . 1.5 and Fig. 1.6 describes the Generator polynomial details of BCH Codes for Normal and Short Frames.

$g_1(x)$	$1+x^2+x^3+x^5+x^{16}$
$g_2(x)$	$1+x+x^4+x^5+x^6+x^8+x^{16}$
$g_3(x)$	$1+x^2+x^3+x^4+x^5+x^7+x^8+x^9+x^{10}+x^{11}+x^{16}$
$g_4(x)$	$1+x^2+x^4+x^6+x^9+x^{11}+x^{12}+x^{14}+x^{16}$
$g_5(x)$	$1+x+x^2+x^3+x^5+x^8+x^9+x^{10}+x^{11}+x^{12}+x^{16}$
$g_6(x)$	$1+x^2+x^4+x^5+x^7+x^8+x^9+x^{10}+x^{12}+x^{13}+x^{14}+x^{15}+x^{16}$
$g_7(x)$	$1+x^2+x^5+x^6+x^8+x^9+x^{10}+x^{11}+x^{13}+x^{15}+x^{16}$
$g_8(x)$	$1+x+x^2+x^5+x^6+x^8+x^9+x^{12}+x^{13}+x^{14}+x^{16}$
$g_9(x)$	$1+x^5+x^7+x^9+x^{10}+x^{11}+x^{16}$
$g_{10}(x)$	$1+x+x^2+x^5+x^7+x^8+x^{10}+x^{12}+x^{13}+x^{14}+x^{16}$
$g_{11}(x)$	$1+x^2+x^3+x^5+x^9+x^{11}+x^{12}+x^{13}+x^{16}$
$g_{12}(x)$	$1+x+x^5+x^6+x^7+x^9+x^{11}+x^{12}+x^{16}$

Figure 1.5: Generator polynomials of the BCH codes for normal frame

LDPC Encoder(INNER Encoder)

- LDPC encoder systematically encodes an information block of size K,$m = (m_0, m_1, \ldots, m_{K-1})$ onto a codeword of size N $c = (m_0, m_1, \ldots, m_{K-1}, p_0, p_1, \ldots, p_{N-K-1})$
- Fig . 1.7 and Fig. 1.8 describes the coding parameters of the LDPC codes for Normal and Short Frames.
- Fig 1.9 and Fig . 1.10 shows the q values as specified in [1],

1.3.3 Bit Interleaver

Fig. 4.2 shows the interleaving sturcture for 8-PSK,16-APSK,32-APSK scheme.

1.3.4 Mapping

The input sequece FECFRAME will be went through PSK modem (QPSK,8-PSK)and APSK modem(16-APSK,32-APSK) will be extensively disussed in chapter 3.

$g_1(x)$	$1+x+x^3+x^5+x^{14}$
$g_2(x)$	$1+x^6+x^8+x^{11}+x^{14}$
$g_3(x)$	$1+x+x^2+x^6+x^9+x^{10}+x^{14}$
$g_4(x)$	$1+x^4+x^7+x^8+x^{10}+x^{12}+x^{14}$
$g_5(x)$	$1+x^2+x^4+x^6+x^8+x^9+x^{11}+x^{13}+x^{14}$
$g_6(x)$	$1+x^3+x^7+x^8+x^9+x^{13}+x^{14}$
$g_7(x)$	$1+x^2+x^5+x^6+x^7+x^{10}+x^{11}+x^{13}+x^{14}$
$g_8(x)$	$1+x^5+x^8+x^9+x^{10}+x^{11}+x^{14}$
$g_9(x)$	$1+x+x^2+x^3+x^9+x^{10}+x^{14}$
$g_{10}(x)$	$1+x^3+x^6+x^9+x^{11}+x^{12}+x^{14}$
$g_{11}(x)$	$1+x^4+x^{11}+x^{12}+x^{14}$
$g_{12}(x)$	$1+x+x^2+x^3+x^5+x^6+x^7+x^8+x^{10}+x^{13}+x^{14}$

Figure 1.6: Generator polynomials of the BCH codes for short frame

LDPC code	BCH Uncoded Block K_{bch}	BCH coded block N_{bch} LDPC Uncoded Block k_{ldpc}	BCH t-error correction	LDPC Coded Block n_{ldpc}
1/4	16 008	16 200	12	64 800
1/3	21 408	21 600	12	64 800
2/5	25 728	25 920	12	64 800
1/2	32 208	32 400	12	64 800
3/5	38 688	38 880	12	64 800
2/3	43 040	43 200	10	64 800
3/4	48 408	48 600	12	64 800
4/5	51 648	51 840	12	64 800
5/6	53 840	54 000	10	64 800
8/9	57 472	57 600	8	64 800
9/10	58 192	58 320	8	64 800

Figure 1.7: LDPC coding paramters for normal frame

1.3.5 Physical Layer Framing(PLFRAMING)

PLFRAMING will gives us the support to synchronize the receiver. One Header block namely PLHEADER will be generated. And will be discussed in chapter 4. Fig . 1.12 shows the Physical Layer Header Sturcture

LDPC Code identifier	BCH Uncoded Block K_{bch}	BCH coded block N_{bch} LDPC Uncoded Block k_{ldpc}	BCH t-error correction	Effective LDPC Rate k_{ldpc}/16 200	LDPC Coded Block n_{ldpc}
1/4	3 072	3 240	12	1/5	16 200
1/3	5 232	5 400	12	1/3	16 200
2/5	6 312	6 480	12	2/5	16 200
1/2	7 032	7 200	12	4/9	16 200
3/5	9 552	9 720	12	3/5	16 200
2/3	10 632	10 800	12	2/3	16 200
3/4	11 712	11 880	12	11/15	16 200
4/5	12 432	12 600	12	7/9	16 200
5/6	13 152	13 320	12	37/45	16 200
8/9	14 232	14 400	12	8/9	16 200
9/10	NA	NA	NA	NA	NA

Figure 1.8: LDPC coding paramters for short frame

q values for normal frames

Code Rate	q
1/4	135
1/3	120
2/5	108
1/2	90
3/5	72
2/3	60
3/4	45
4/5	36
5/6	30
8/9	20
9/10	18

Figure 1.9: q value table for normal frame

q values for short frames

Code Rate	q
1/4	36
1/3	30
2/5	27
1/2	25
3/5	18
2/3	15
3/4	12
4/5	10
5/6	8
8/9	5

Figure 1.10: q value table for shortframe

Modulation	Rows (for n_{ldpc} = 64 800)	Rows (for n_{ldpc} = 16 200)	Columns
8PSK	21 600	5 400	3
16APSK	16 200	4 050	4
32APSK	12 960	3 240	5

Figure 1.11: Bit interleaver structure

1.3.6 Baseband shaping

- To mitigate the tradeoff between high data rate and bandwidth, A pulse shaper will be performed after the mapping schemes and it can be varied according to the excess bandwidth called roll-off factor.

- extensively discussed in chapter 4.

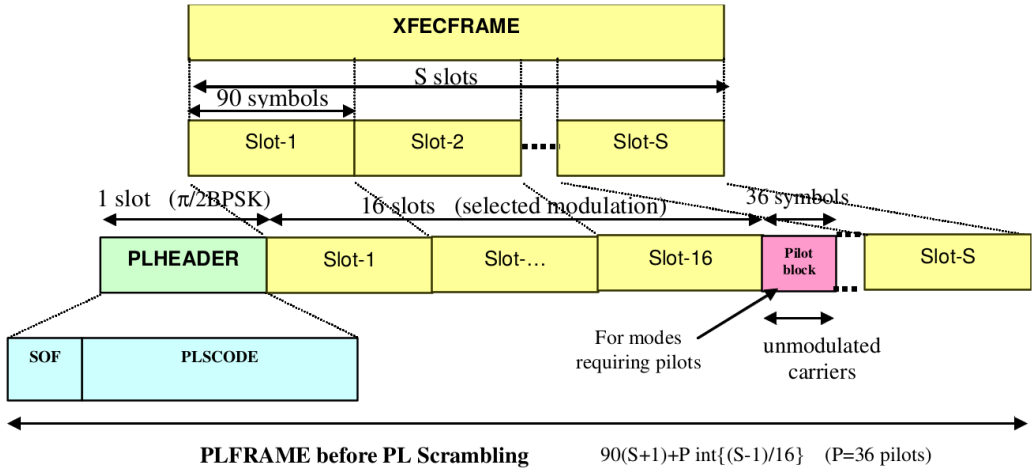


Figure 1.12: Physical Layer Header PLHEADER structure

1.4 Receiver

1.4.1 Block Diagram

Fig .1.13 shows the receiver structure.

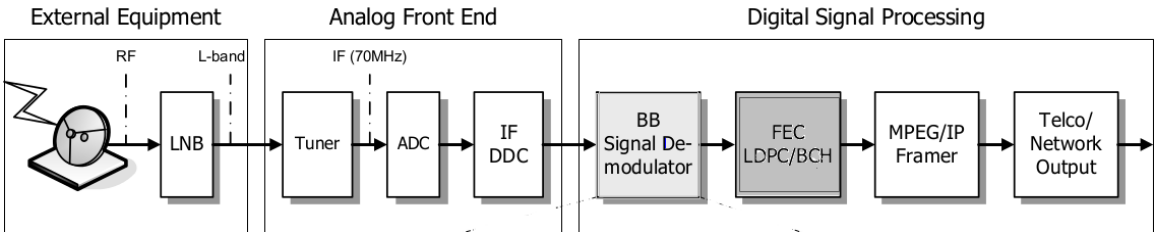


Figure 1.13: Block diagram of the receiver

1.4.2 Baseband Demodulator block

As specified in [6], digital synchronization techniques for reliable communication such as Symbol Timing Offset Recovery, Frame Offset Recovery, Carrier Frequency offset Recovery, Phase offset recovery and A digital Automatic gain controller subblocks are listed below. A theoretical and implementation results are listed in chapter 4.

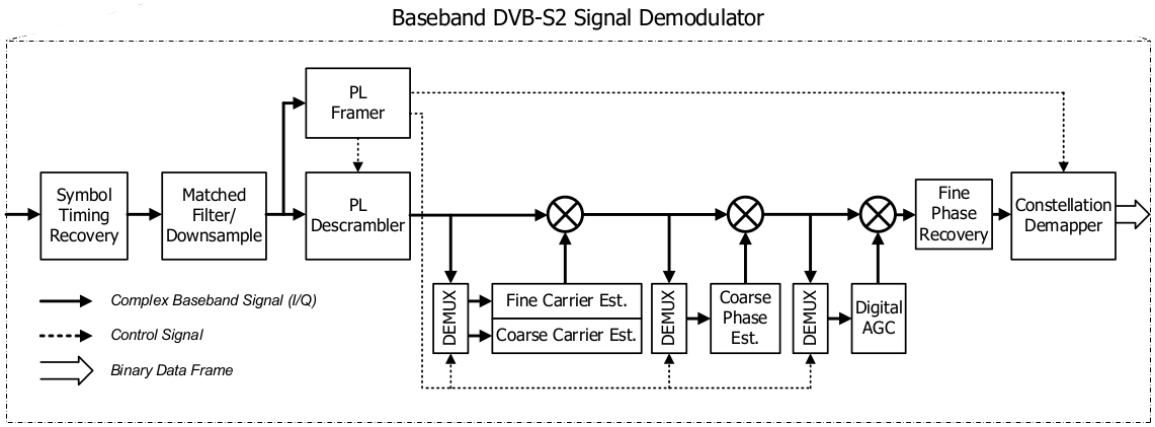


Figure 1.14: Synchronization Block

Symbol Timing Recovery

Fig 1.15 used to mitigate the effects of Symbol Timing recovery using Gardener Timing Error Detector Algorithm.

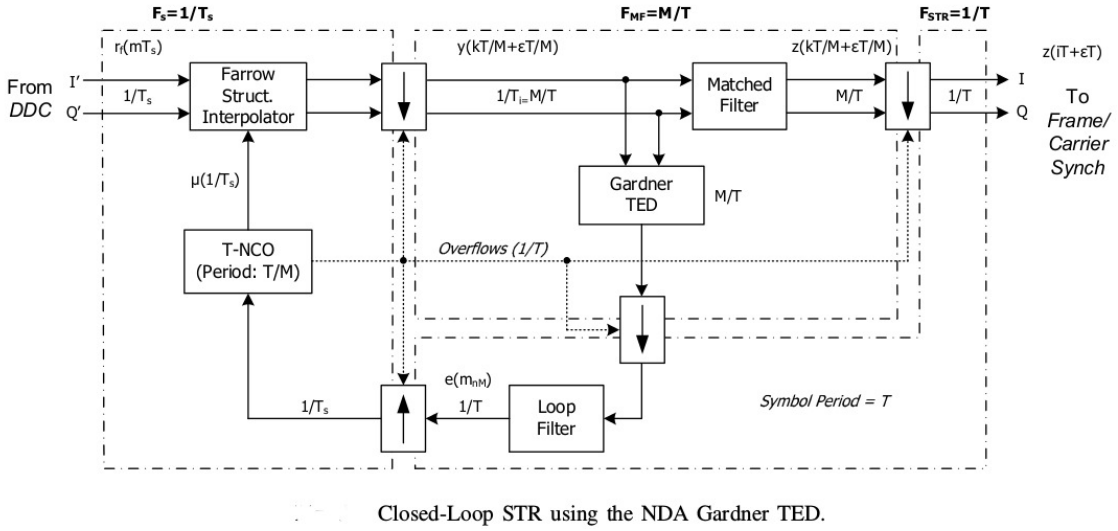
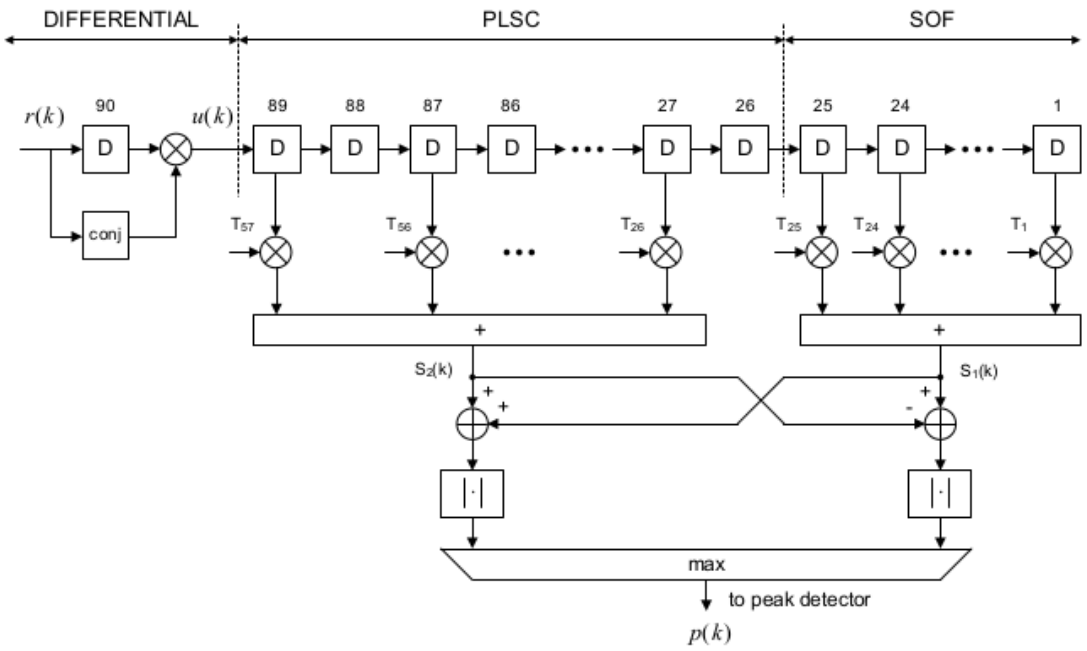


Figure 1.15: Timing synchronization block diagram

Frame Synchronization

Fig. 1.16 used to mitigate the effects of Frame offset recovery using Global Summation of SOF/PLSC Detectors.



Differential PL header detection.

Figure 1.16: Frame synchronization block diagram

Carrier Frequency Recovery

Fig. 1.17, Fig. 1.18 Shows the Frequency offset recovery using the LR Technique.

Phase Recovery

Fig. 1.19, Fig .1.20 shows the Coarse Phase offset recovery, Fine phase offset recovery based on the Feed-Forward Maximum likelihood Technique.

Digital Automatic Gain Control

Fig. 1.21 shows the DAGC block and based upon the Data-Aided Vector-Tracker (DA-VT).DA-VT uses the known pilots to estimate the amplitude offset factor.

1.4.3 FEC Decoding

FEC Decoding will be carried out by the LLR calculation, LDPC Decoding using Message Passing Algorithm and BCH Decoding using the BERLEKAMP-MASSEY Algorithm are given in the chapter 2.

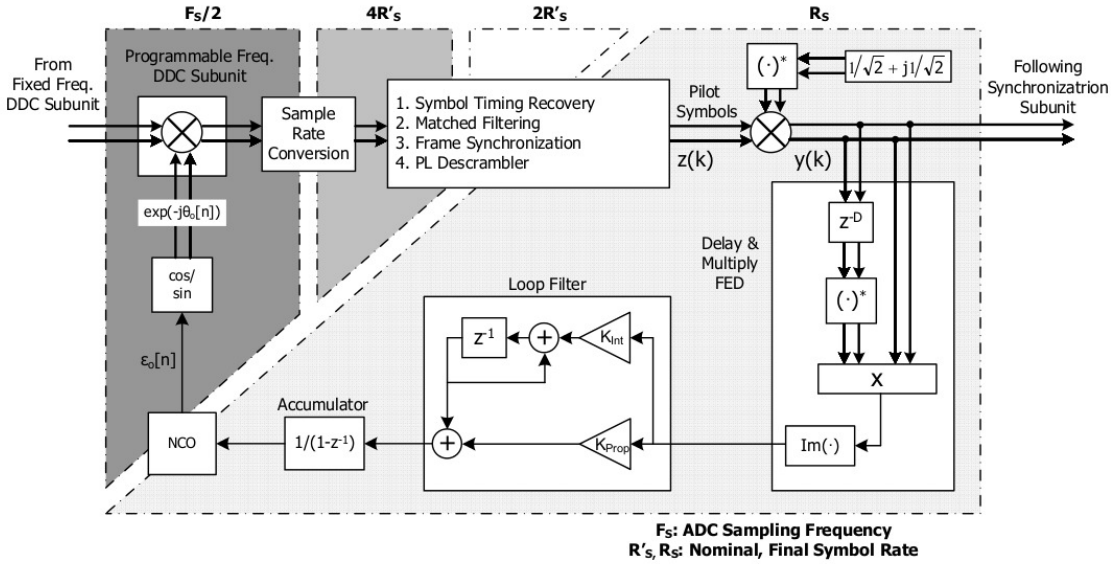
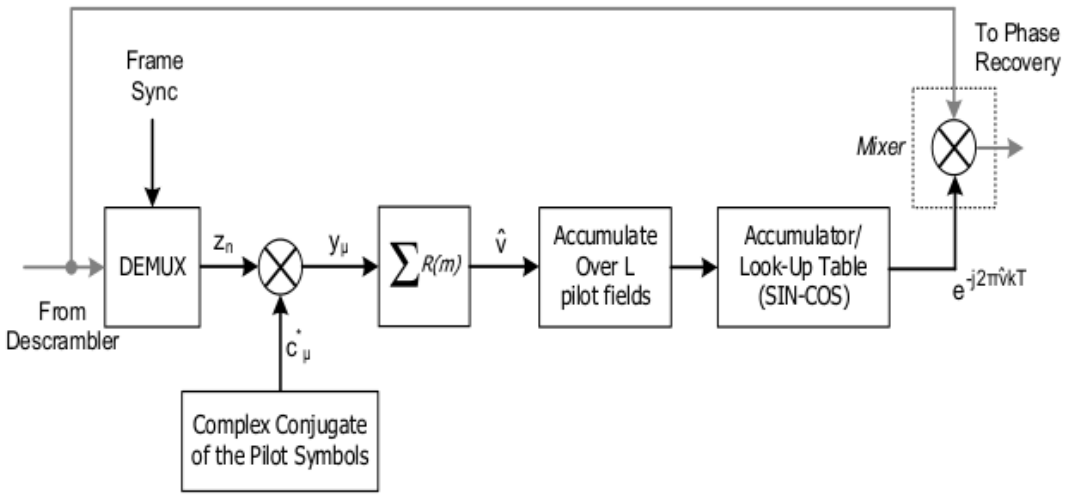


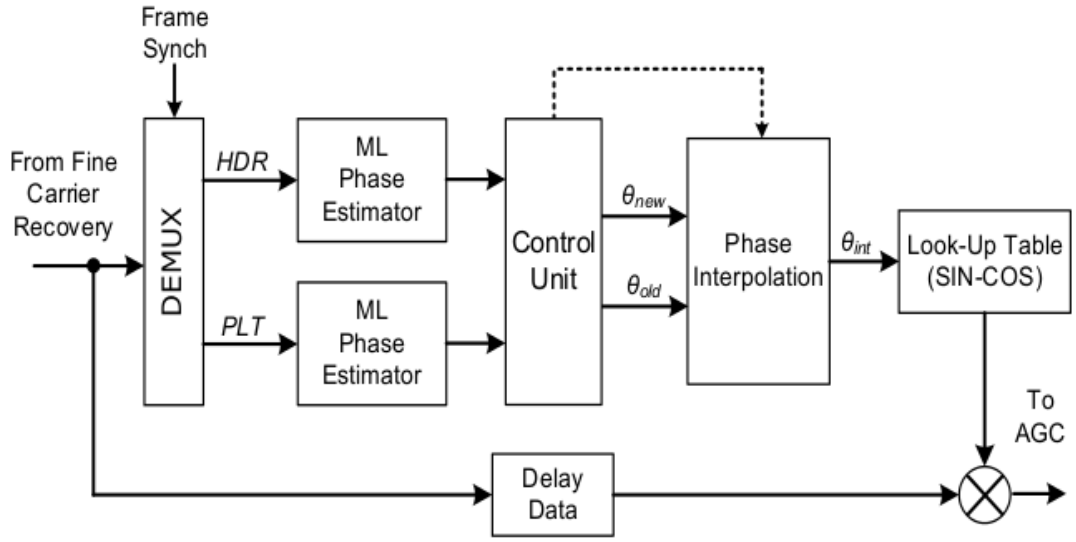
Fig. 4. Coarse carrier frequency recovery.

Figure 1.17: Coarse Carrier Frequencysynchronization block diagram



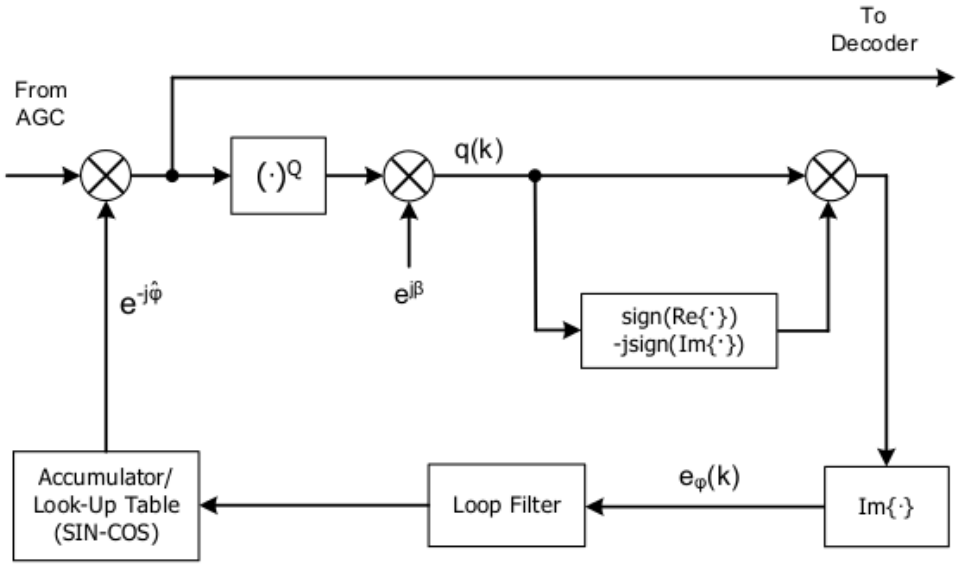
Fine carrier frequency recovery.

Figure 1.18: Fine Carrier Frequency synchronization block diagram



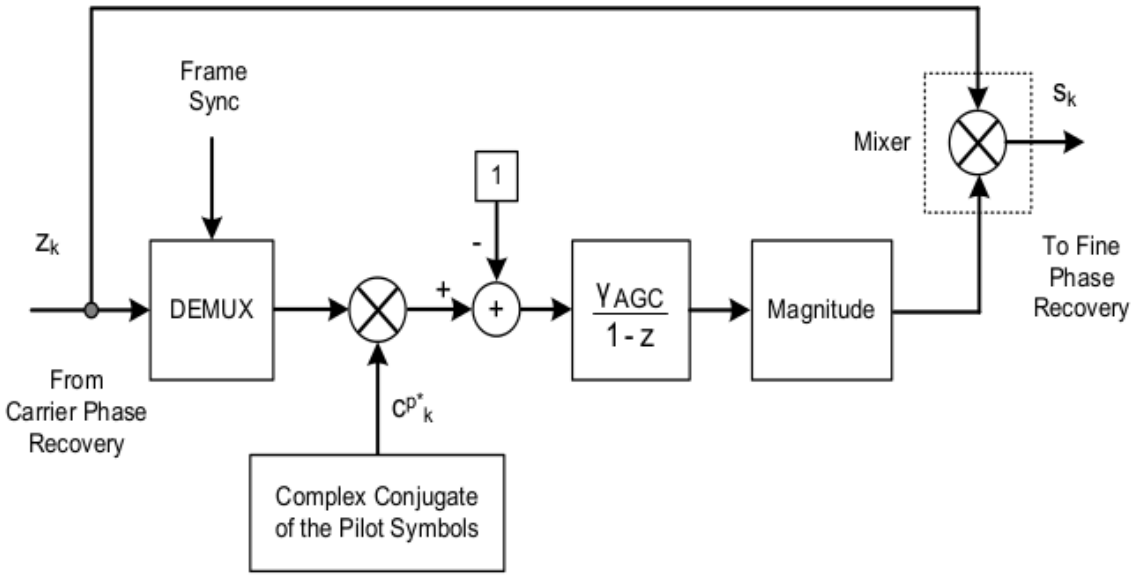
Coarse phase recovery.

Figure 1.19: Coarse phase synchronization block diagram



Fine phase recovery.

Figure 1.20: Fine phase synchronization block diagram



Digital automatic gain control (DAGC).

Figure 1.21: Automatic Grain controller block diagram

Chapter 2

Channel Coding techniques

2.1 Bose–Chaudhuri–Hocquenghem (BCH) codes

2.1.1 Encoding

The classic algorithm we are going to follow,

- The generator polynomial is obtained as

$$g(x) = \prod_{i=1}^{m} g_i(x) \tag{2.1}$$

where $m = 12$ and each $g_i(x)$ is based upon the Frame length. Let $\vec{m}$ be a $k \times 1$ message vector and

$$m(x) = m_{k-1}x^{k-1} + m_{k-2}x^{k-2} + \cdots + m_1 x + m_0 \tag{2.2}$$

be the corresponding Message polynomial.

- Let

$$m(x)x^{n-k} = q(x)g(x) + d(x) \tag{2.3}$$

and

$$c(x) = m(x)x^{n-k} + d(x) \tag{2.4}$$

Therefore, Codeword bits are written as,

$$c = [m_0, m_1, \ldots, m_{k-1}, d_0, d_1, \ldots, d_{n-k-1}] \tag{2.5}$$

2.1.2 Decoding

[7], [8] Let $r(x)$ is the received polynomial.

$$r(x) = r_0 + r_1 x + \cdots + r_{n-1} x^{n-1} \tag{2.6}$$

$$r(x) = c(x) + e(x) \tag{2.7}$$

$$\tag{2.8}$$

$$\vec{r} = \vec{c} + \vec{e} \tag{2.9}$$

The Standard Decoding process of BCH Codes are carried out by 3 steps.

1. Syndrome Calculation $S(x)$
2. Calculation of Key Equation Solver i.e Error Locator Polynomial $\Delta(x)$
3. Finding the Roots of the Error Locator Polynomial using Chien search Algorithm.

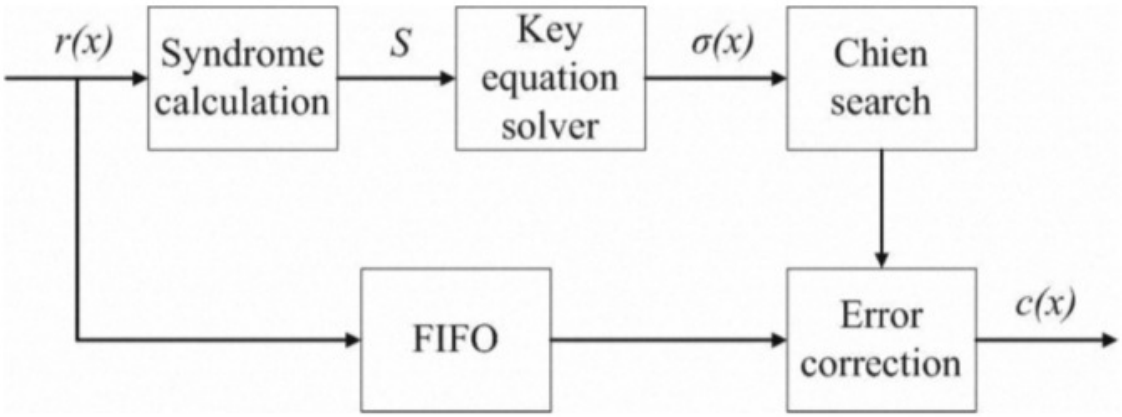


Figure 2.1: BCH Decoding Block Diagram

Syndrome Calculation

$$\vec{S} = [S_1, S_2, \ldots, S_{2t}] \tag{2.10}$$

Definie, Syndrome

$$S_i = r(\alpha^i) = c(\alpha^i) + e(\alpha^i) = e(\alpha^i) \tag{2.11}$$

Where α^i is a root of the codeword.

Calculation of Error Locator Polynomial $\Delta(x)$: Berlekamp's Decoding Algorithm

1. Intialization : $k = 0, \Lambda^{(0)}(x) = 1, T^{(0)} = 1$

2. Let $\Delta^{(2k)}$ be the coefficient of x^{2k+1} in $\Lambda^{(2k)}[1 + S(x)]$

3. Compute
$$\Delta^{(2k+2)}(x) = \Lambda^{(2k)}(x) + \Delta^{(2k)}[x.T^{(2k)}(x)] \tag{2.12}$$

4. Compute
$$T^{(2k+2)}(x) = \begin{cases} x^2 T^{(2k)}(x) & \text{if } \Delta^{(2k)} = 0 \text{ or } \deg\left[\Lambda^{(2k)}(x)\right] > k \\ \frac{x\Lambda^{(2k)}(x)}{\Delta^{(2k)}} & \text{if } \Delta^{(2k)} \neq 0 \text{ or } \deg\left[\Lambda^{(2k)}(x)\right] \leq k \end{cases} \tag{2.13}$$

5. Set $k = k + 1$.If $k < t$ then go to step3.

6. Return the Error Locator polynomial $\Lambda(x) = \Lambda^{(2k)}(x)$

Roots of Error Locator polynomial :The Chien's Search Algorithm

1. Take α^j as test root . $0 \leq j \leq n - 1$.

2. if Λ_i test every root and if its equals to zero. Then that is root.

Decision

After finding the roots, flip the bit values at the root positions which gives you the estimated codeword. i.e $\hat{c}$.

2.1.3 Simulation Results

Fig. 2.2 shows the SNR vs BER graph for BCH Code $(3072, 3240, 12)$ i.e information length $k = 3072$ bits, code word length $n = 3240$ bits and error correcting capability $t = 12$.

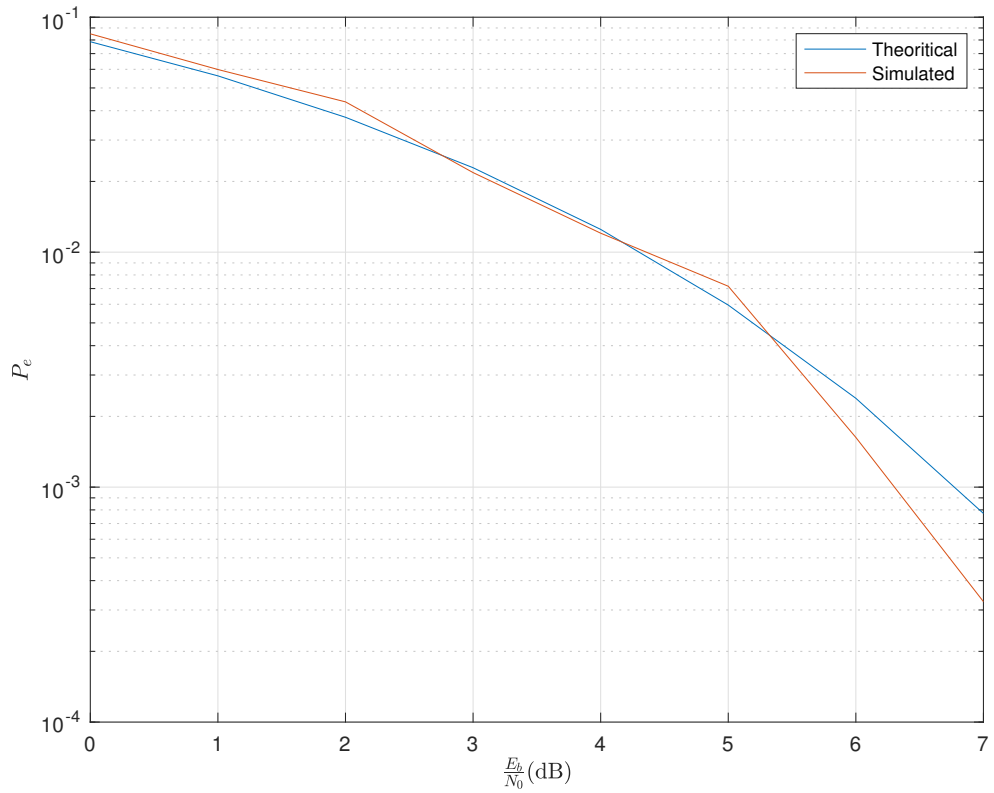


Figure 2.2: SNR vs BER performance of BCH Codes

2.2 Low-Density Parity Check codes

LDPC codes are popular linear block codes with closest shannon limt channel capacity [2]. As an example Lets take (7,4) Hamming parity check matrix.

$$H = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \tag{2.14}$$

Above H matrix having the parameters, information bits $k = 4$ i.e $m = m_0, \ldots, m_3$, parity bits are $m = n - k = 3$ i.e $p = [p_0, p_1, p_2]$ and code word length $n = 7$ i.e $c = [m\ p]$.

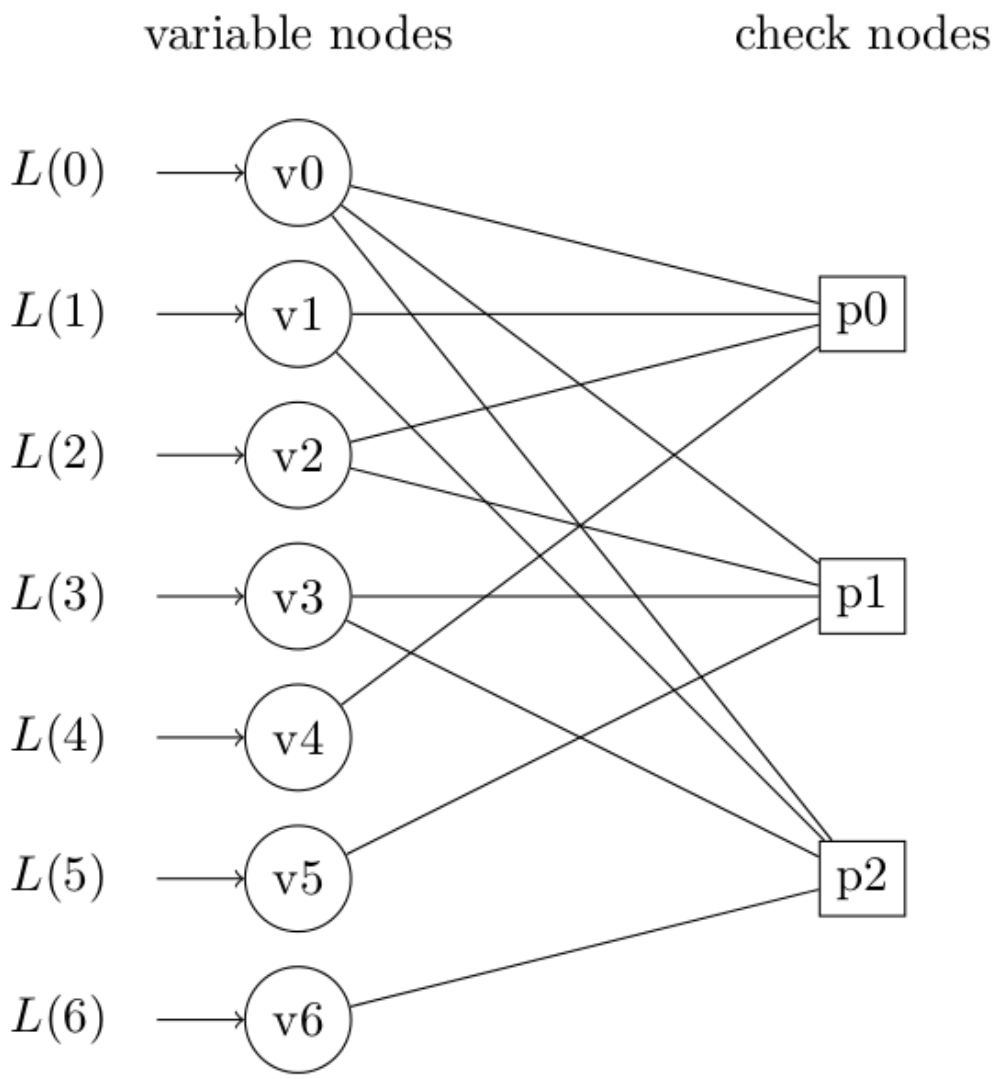


Figure 2.3: Tanner Graph Representation for (7,4) Hamming parity check matrix

2.2.1 Encoding

Encoding can be carried out by using

$$H \times c^T = 0 \tag{2.15}$$

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} m_0 \\ m_1 \\ m_2 \\ m_3 \\ p_0 \\ p_1 \\ p_2 \end{bmatrix} = 0 \tag{2.16}$$

solving we get

$$p_0 = m_0 \oplus m_1 \oplus m_2 \tag{2.17}$$

$$p_1 = m_0 \oplus m_2 \oplus m_3 \tag{2.18}$$

$$p_2 = m_0 \oplus m_1 \oplus m_3 \tag{2.19}$$

This is called Systematic Encoding.i.e Encoder will ensures information bits followed by parity bits.

2.2.2 Decoding

Useful Calculations for proceeding LDPC Decoding

1. Calculation of Input Channel Log Likelihood Ratio LLR

$$L(x_j) = \log\left(\frac{Pr(x_j = 1|y)}{Pr(x_j = -1|y)}\right) \quad X = 1 - 2c \tag{2.20}$$

$$= \log\left(\frac{f(y|x_j = 1)Pr(x_j = 1)}{f(y|x_j = -1)Pr(x_j = -1)}\right) \tag{2.21}$$

$$= \log\left(\frac{\frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(y_j-1)^2}{2\sigma^2}}}{\frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(y_j+1)^2}{2\sigma^2}}}\right) \tag{2.22}$$

$$= \log\left(e^{\frac{2y_j}{\sigma^2}}\right) \tag{2.23}$$

$$L(x_j) = \frac{2y_j}{\sigma^2} \tag{2.24}$$

2. Check Node Operation :
   Let's assume that we have initialized all LLR values to variable nodes and we sent to check nodes. V_j represents all the variable nodes which are connected

to j^{th} check node. Using the min-sum approximation [3], the message from j^{th} check node to i^{th} variable node given by, since parity node equation for the first

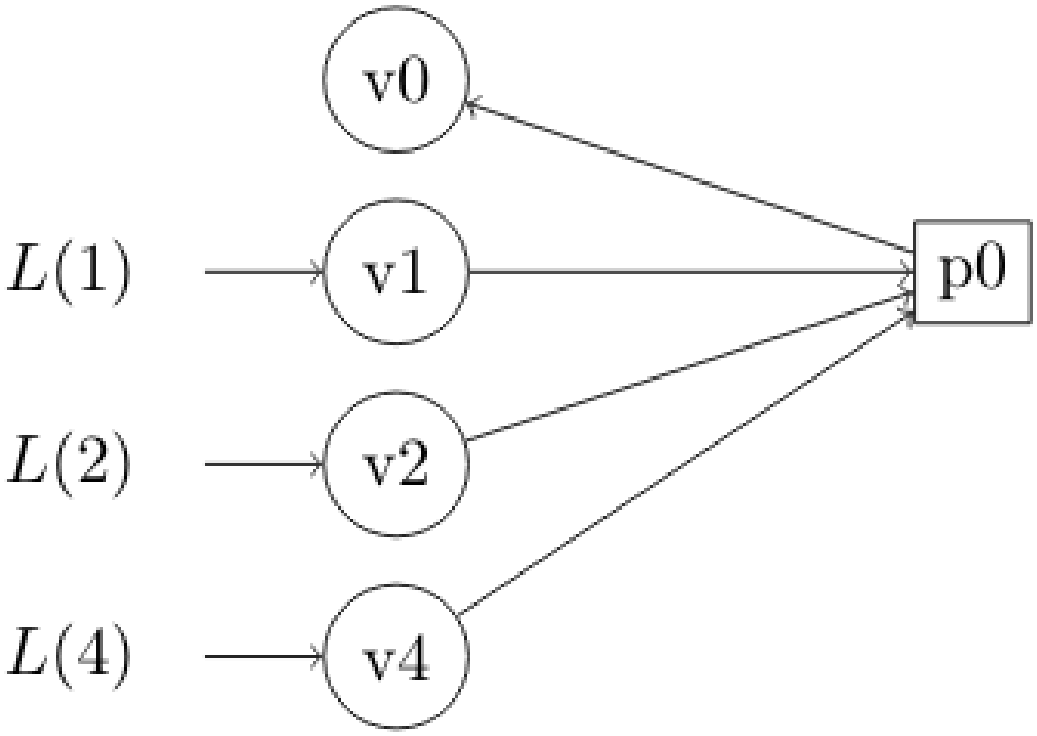


Figure 2.4: Check node operation

check node is $p_0 = m_0 + m_1 + m_2 + m_4$. we need to calculate

$$L_{ext0,0} = \log\left(\frac{Pr(x_0 = 0|y_1, y_2, y_4)}{Pr(x_0 = 1|y_1, y_2, y_4)}\right) \tag{2.25}$$

Defining,

$$L_1 = \log\left(\frac{Pr(x_1 = 0|y_1)}{Pr(x_1 = 1|y_1)}\right) = \log\left(\frac{p_1}{1 - p_1}\right) \tag{2.26}$$

$$L_2 = \log\left(\frac{Pr(x_2 = 0|y_2)}{Pr(x_2 = 1|y_2)}\right) = \log\left(\frac{p_2}{1 - p_2}\right) \tag{2.27}$$

$$L_4 = \log\left(\frac{Pr(x_4 = 0|y_4)}{Pr(x_4 = 1|y_4)}\right) = \log\left(\frac{p_4}{1 - p_4}\right) \tag{2.28}$$

$$\tag{2.29}$$

Using Table. 4.1 we can find the

c0	c1	c2	c4
0	0	0	0
1	0	0	1
1	0	1	0
0	0	1	1
1	1	0	0
0	1	0	1
0	1	1	0
1	1	1	1

Table 2.1: Probability of a varibale node from other check nodes

$$p_0 = Pr(c_0 = 0|c_1, c_2, c_4) \tag{2.30}$$

$$\begin{aligned} p_0 &= p_1p_2p_4 + p_1(1-p_2)(1-p_4) \\ &\quad + (1-p_1)p_2(1-p_4) + (1-p_1)(1-p_2)p_4 \\ 1-p_0 &= p_1p_2(1-p_4) + p_1(1-p_2)p_4 \\ &\quad + (1-p_1)p_2p_4 + (1-p_1)(1-p_2)(1-p_4) \end{aligned}$$

by rearranging above equations,

$$p_0 - (1-p_0) = p_1 - (1-p_1) + p_2 - (1-p_2) + p_4 - (1-p_4) \tag{2.31}$$

Where p_i is the probability. Getting message from check to variable node by taking all variable node informations.

$$\frac{1-\frac{1-p_0}{p_0}}{1+\frac{1-p_0}{p_0}} = \frac{1-\frac{1-p_1}{p_1}}{1+\frac{1-p_1}{p_1}} \times \frac{1-\frac{1-p_2}{p_2}}{1+\frac{1-p_2}{p_2}} \times \frac{1-\frac{1-p_4}{p_4}}{1+\frac{1-p_4}{p_4}} \tag{2.32}$$

$$\frac{1-e^{-L_{ext0,0}}}{1+e^{-L_{ext0,0}}} = \frac{1-e^{-L_1}}{1+e^{-L_1}} \times \frac{1-e^{-L_2}}{1+e^{-L_2}} \times \frac{1-e^{-L_4}}{1+e^{-L_4}} \tag{2.33}$$

$$-\tanh\left(\frac{L_{ext0,0}}{2}\right) = \left(-\tanh\left(\frac{L_1}{2}\right)\right)\left(-\tanh\left(\frac{L_2}{2}\right)\right)\left(-\tanh\left(\frac{L_4}{2}\right)\right)$$

$$\tanh\left(\frac{L_{ext0,0}}{2}\right) = \left[\prod_{k\in V_j\setminus i} \alpha_{k,0}\right]\left[\prod_{k\in V_j\setminus i} \tanh\left(\frac{\beta_{k,0}}{2}\right)\right] \tag{2.34}$$

$$L_{ext0,0} = \left(\prod_{k\in V_j\setminus i} \alpha_{k,0}\right) 2tanh^{-1}\left(\prod_{k\in V_j\setminus i} \tanh\left(\frac{\beta_{k,0}}{2}\right)\right) \tag{2.35}$$

$$= \left(\prod_{k \in V_j \backslash i} \alpha_{k,0} \right) 2tanh^{-1} \log^{-} \log \left(\prod_{k \in V_j \backslash i} \tanh \left(\frac{\beta_{k,0}}{2} \right) \right) \quad (2.36)$$

$$= \left(\prod_{k \in V_j \backslash i} \alpha_{k,0} \right) 2tanh^{-1} \log^{-1} \left(\sum_{k \in V_j \backslash i} \log \tanh \left(\frac{\beta_{k,0}}{2} \right) \right) \quad (2.37)$$

$$L_{ext0,0} = \left(\prod_{k \in V_j \backslash i} \alpha_{k,0} \right) f \left(\sum_{k \in V_j \backslash i} f \left(\beta_{k,0} \right) \right) \quad (2.38)$$

Where,

$$\alpha_{k,j} = sign(L_{k,j}) \quad (2.39)$$

$$\beta_{k,j} = |L_{k,j}| \quad (2.40)$$

$$f(x) = -\log \left(\tanh \frac{x}{2} \right) \quad (2.41)$$

Using the Fig 2.5 and using its 45° symmetricity, we can approximate the above

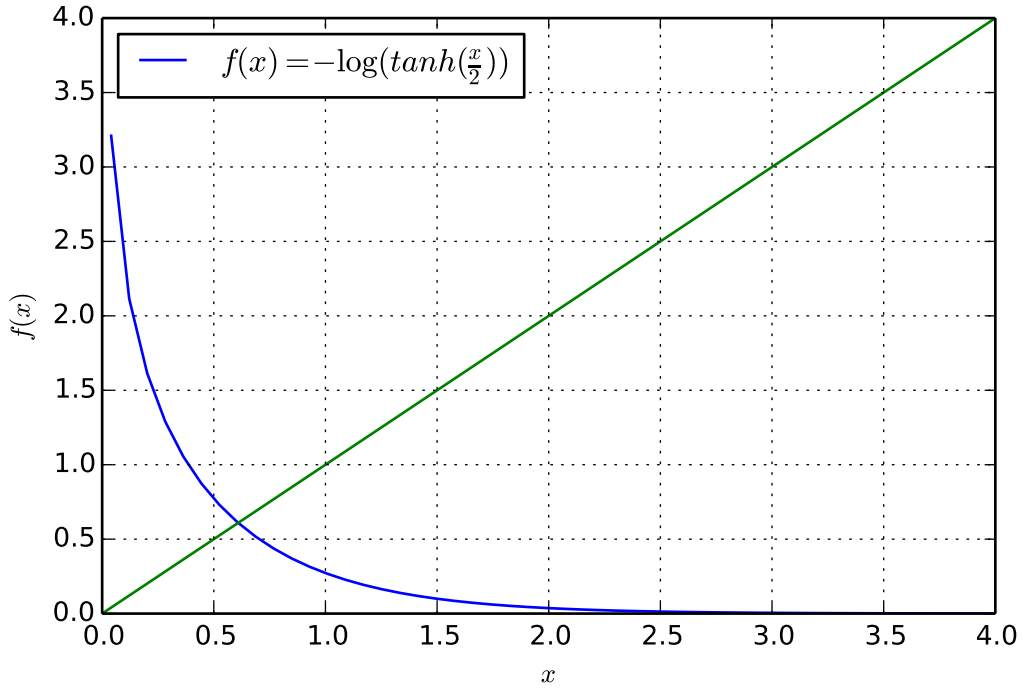


Figure 2.5: Plot of function $f(x)$

equation as, given by minimum sum approximation [3]

$$f\left(\sum_{k\in V_j\backslash i} f\left(\beta_{k,0}\right)\right) \approx f\left(f\left(\min_{k\in V_j\backslash i}\left(\beta_{k,0}\right)\right)\right) \tag{2.42}$$

$$= \min_{k\in V_j\backslash i}\left(\beta_{k,0}\right) \tag{2.43}$$

Combining (2.43) in (2.38),

$$L(r_{j=0,i=0}) = \left(\prod_{k\in V_j\backslash i} \alpha_{k,0}\right)\left(\min_{k\in V_j\backslash i}\left(\beta_{k,0}\right)\right) \tag{2.44}$$

3. Variable Node Operation :
   Let C_i denotes all the check nodes connected to i^{th} variable node. The message from i^{th} variable node to j^{th} check node given by,

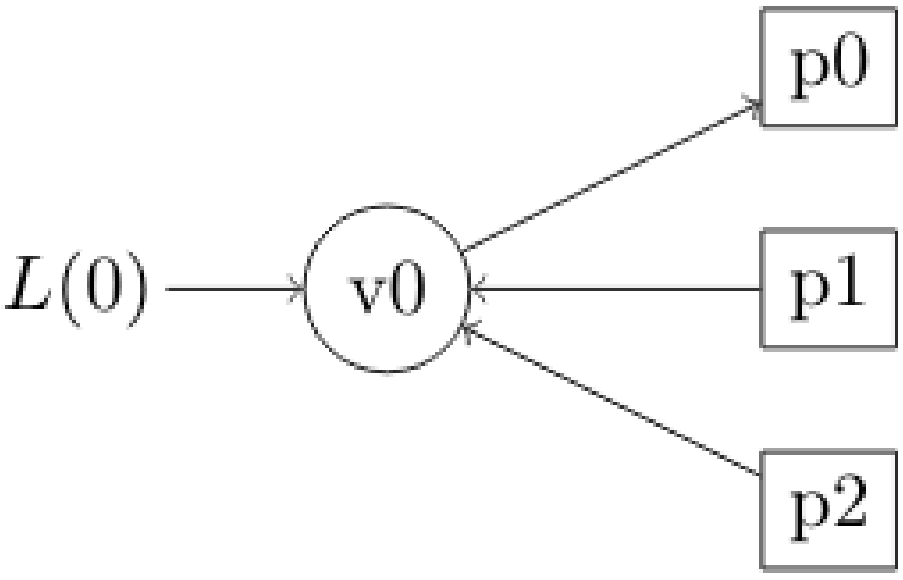


Figure 2.6: Variable node operation

$$L(q_{i=0,j=0}) = \log\left(\frac{Pr(x_j=1|y_0,y_1,y_2)}{Pr(x_j=-1|y_0,y_1,y_2)}\right) \quad X = 1-2c \tag{2.45}$$

$$= \log\left(\frac{f(y_0,y1,y_2|x_j=1)Pr(x_j=1)}{f(y_0,y_1,y_2|x_j=-1)Pr(x_j=-1)}\right) \tag{2.46}$$

$$= \log\left(\frac{\frac{1}{\sqrt{2\pi\sigma^2}}e^{\frac{-(y_0-1)^2}{2\sigma^2}}\frac{1}{\sqrt{2\pi\sigma^2}}e^{\frac{-(y_1-1)^2}{2\sigma^2}}\frac{1}{\sqrt{2\pi\sigma^2}}e^{\frac{-(y_2-1)^2}{2\sigma^2}}}{\frac{1}{\sqrt{2\pi\sigma^2}}e^{\frac{-(y_0+1)^2}{2\sigma^2}}\frac{1}{\sqrt{2\pi\sigma^2}}e^{\frac{-(y_1+1)^2}{2\sigma^2}}\frac{1}{\sqrt{2\pi\sigma^2}}e^{\frac{-(y_2+1)^2}{2\sigma^2}}}\right) \tag{2.47}$$

$$= \log\left(e^{\frac{2(y_0+y_1+y_2)}{\sigma^2}}\right) \tag{2.48}$$

$$L(q_{i=0,j=0}) = \frac{2(y_0+y_1+y_2)}{\sigma^2} = L(x_i) + \sum_{k\in C_i\backslash j} L(r_{ki}) \tag{2.49}$$

Message Passing Algorithm using min-sum Approximation

Transmitted frames = N, Total number of bits = N × 7 and Total number of information bits = N × 4. For Each Frame,

1. Initialize $L(q_{ij})$ using (2.24) for all i, j for which $h_{ij} = 1$ with channel LLR's.
2. Update $\{L(r_{ji})\}$ using 2.44.
3. Update $\{L(q_{ji})\}$ using 2.49.
4. Update $\{L(V_i)\}$ using,

$$L(V_i) = L(x_i) + \sum_{k\in C_i} L(r_{ki}) \quad i = 0,\ldots,6. \tag{2.50}$$

5. Proceed to step 2.
   After maximum specified iterations,

Decoding can be done using,

$$\hat{c_i} = \begin{cases} 1 & L(V_i) < 0 \\ 0 & else \end{cases} \tag{2.51}$$

Simulation of (7,4) Hamming parity check using LDPC coding

For frames N=10000. Fig 2.7 shows the Comparison of Probability error with channel coding and without channel coding. Since the parity check matrix taken was not much sparse, we are not getting near Shannon limit performance. For good sparse matrix i.e number of entries in H $<< m \times n$

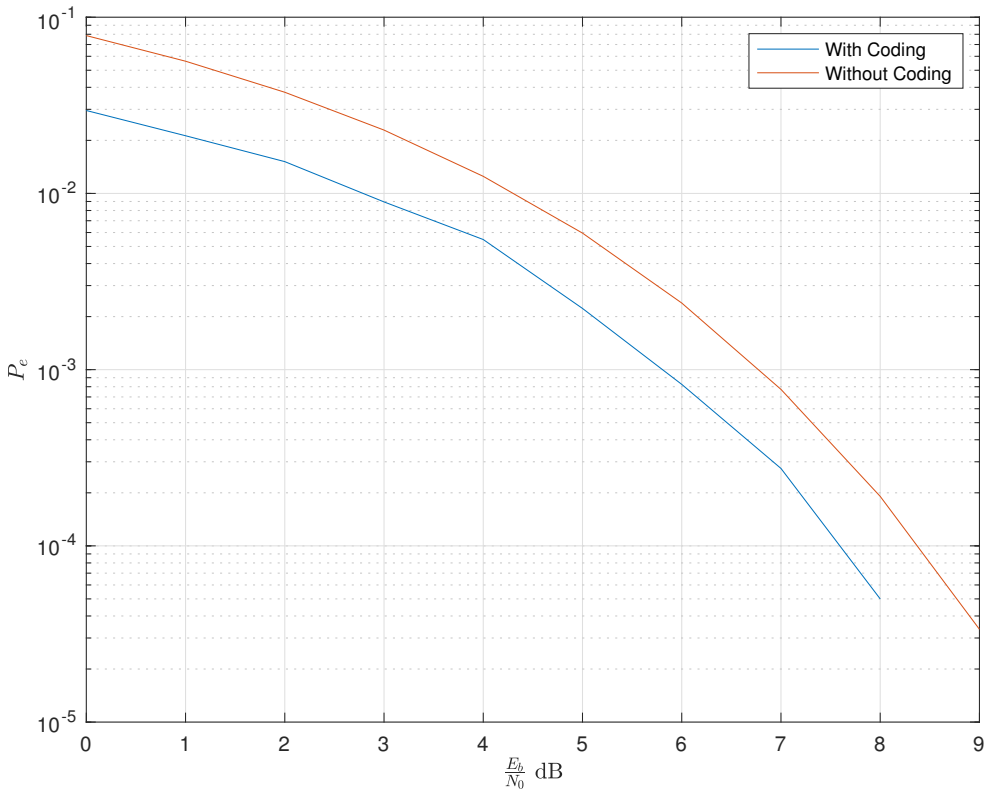


Figure 2.7: SNR vs BER curves using LDPC channel coding and no channel coding

2.2.3 Implementation of DVB-S2 standard

Encoding

The parity check matrix defined in standard, $H_{N-K\times N} = [A_{N-K\times K}|B_{N-K\times N-K}]$. Where N is the frame length and K is the information length of the LDPC Encoder.

$$H = \left[\begin{array}{c|cccccc} & 1 & 0 & 0 & \dots & 0 \\ & 1 & 1 & 0 & \dots & 0 \\ A & 0 & 1 & 1 & \dots & 0 \\ & \vdots & \vdots & \vdots & \vdots & 1 \\ & 0 & 0 & 0 & \dots & 1 \end{array}\right]$$

and Matrix A defined as

$$A_{N-K\times K} = \begin{bmatrix} a_{0,0} & a_{0,1} & \dots & \dots & a_{0,K-1} \\ a_{1,0} & a_{1,1} & \dots & \dots & a_{1,K-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{N-K-1,0} & a_{N-K-1,1} & \dots & \dots & a_{N-K-1,K-1} \end{bmatrix}$$

and the process of generating the matrix is given by the algorithm specified in the DVB-S2 [1].

Say our, Information bits are $m = [m_0, m_1, \dots, m_{K-1}]$ and the Codeword with the frame length is given by the $c = [m_0, m_1, \dots, m_{K-1}, p_0, p_1, \dots, p_{N-K-1}]$.

The systematic Encoding process is carried out by the,

$$H \times c^T = 0 \tag{2.52}$$

Therefore by arranging the equations for each parity bit p_i,

$$p_i = a_{i,0}m_0 \oplus a_{i,1}m_1 \oplus \cdots \oplus a_{i,K-1}m_{K-1} \oplus p_{i-1} \quad i = 0, 1, \dots, N-K-1 \tag{2.53}$$

Decoding

A generalized and approximated Log Likelihood Ratio given by [9]

1. Calculation of Log Likelihood Ratio's for Higher Order Mapping Schemes

$$LLR(b_j) = \log\left(\frac{Pr(b_j = 0|y)}{Pr(b_j = 1|y)}\right) \tag{2.54}$$

$$LLR(b_j) = \log\left(\frac{\sum_{i\in\left\{K_{b_j=0}\right\}} \frac{1}{2\sigma^2} e^{\frac{-|y-x_i|^2}{2\sigma^2}}}{\sum_{i\in\left\{K_{b_j=1}\right\}} \frac{1}{2\sigma^2} e^{\frac{-|y-x_i|^2}{2\sigma^2}}}\right) \tag{2.55}$$

Using the maximum logarithmic Approach,

$$\log\left(e^a + e^b\right) \approx \max\left(a, b\right) \tag{2.56}$$

The LLR expression can be written as,

$$LLR(b_j) = \max_{i\in\left\{K_{b_j=0}\right\}}\left(\frac{-\left|y-x_i\right|^2}{2\sigma^2}\right) - \max_{i\in\left\{K_{b_j=1}\right\}}\left(\frac{-\left|y-x_i\right|^2}{2\sigma^2}\right) \tag{2.57}$$

2. Approximated LLR's for QPSK Mapping Scheme
   For the QPSK mapping scheme , b_1, b_0 are the MSB and LSB of the mapped symbols.

$$LLR(b_1) \approx \max(L_0, L_1) - \max(L_3, L_2) \tag{2.58}$$
$$LLR(b_0) \approx \max(L_0, L_2) - \max(L_1, L_3) \tag{2.59}$$

   Where,

$$L_i = -\frac{|y - x_i|^2}{2\sigma^2} \quad i = 0, 1, 2, 3 \tag{2.60}$$

3. Approximated LLR's for 8-PSK Mapping Scheme
   For the 8-PSK mapping scheme , b_2, b_0 are the MSB and LSB of the mapped symbols.

$$LLR(b_2) \approx \max(L_0, L_1, L_3, L_2) - \max(L_6, L_7, L_5, L_4) \tag{2.61}$$
$$LLR(b_1) \approx \max(L_0, L_1, L_5, L_4) - \max(L_3, L_2, L_6, L_7) \tag{2.62}$$
$$LLR(b_0) \approx \max(L_0, L_2, L_6, L_4) - \max(L_1, L_3, L_7, L_5) \tag{2.63}$$

   Where,

$$L_i = e^{-\frac{|y-x_i|^2}{2\sigma^2}} \quad i = 0, 1, \ldots, 7 \tag{2.64}$$

2.2.4 Simulation Resutls

Fig. 2.8 Shows the SNR vs BER comparison of LDPC codes for Framelength $N = 64800$ bits, Code Rate $R = \frac{1}{2}$ and number of decoding iterations are 5 with BPSK mapping scheme. Fig. 2.9 Shows the SNR vs BER comparison of LDPC codes for Framelength $N = 64800$ bits, Code Rate $R = \frac{3}{5}$ and number of decoding iterations are 5 with QPSK mapping scheme.

Similarly, Fig. 2.10 Shows the SNR vs BER comparison of LDPC codes for Framelength $N = 64800$ bits, Code Rate $R = \frac{3}{5}$ and number of decoding iterations are 5 with 8-PSK mapping scheme.

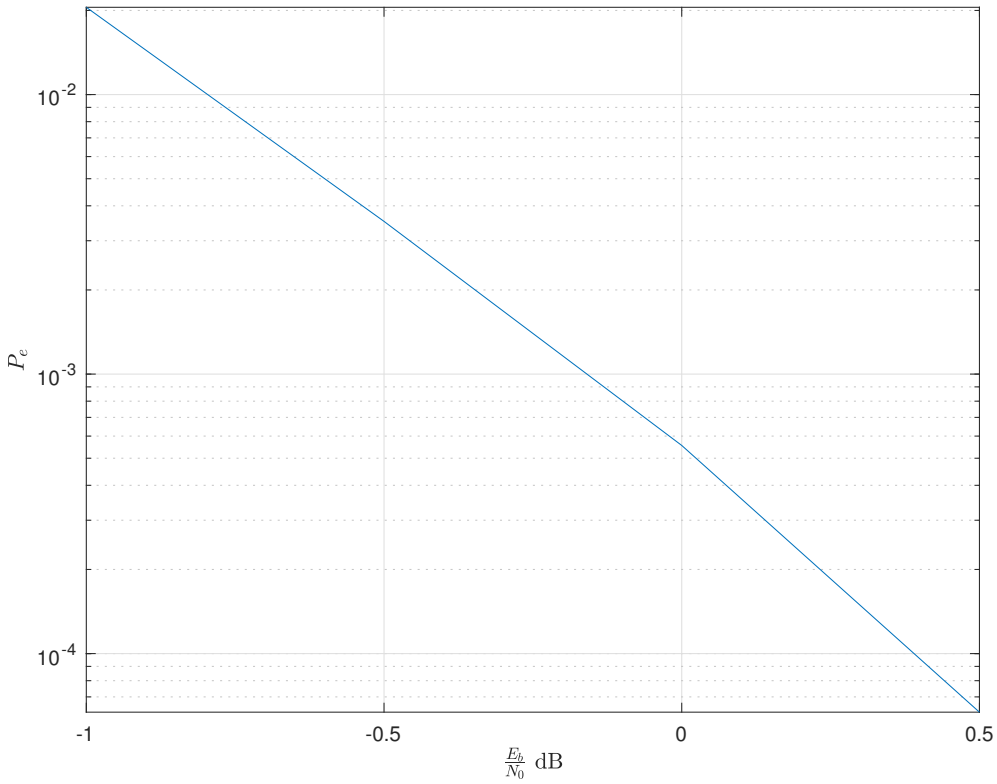


Figure 2.8: SNR vs BER performance of LDPC Codes code rate $R = \frac{1}{2}$ using BPSK mapping

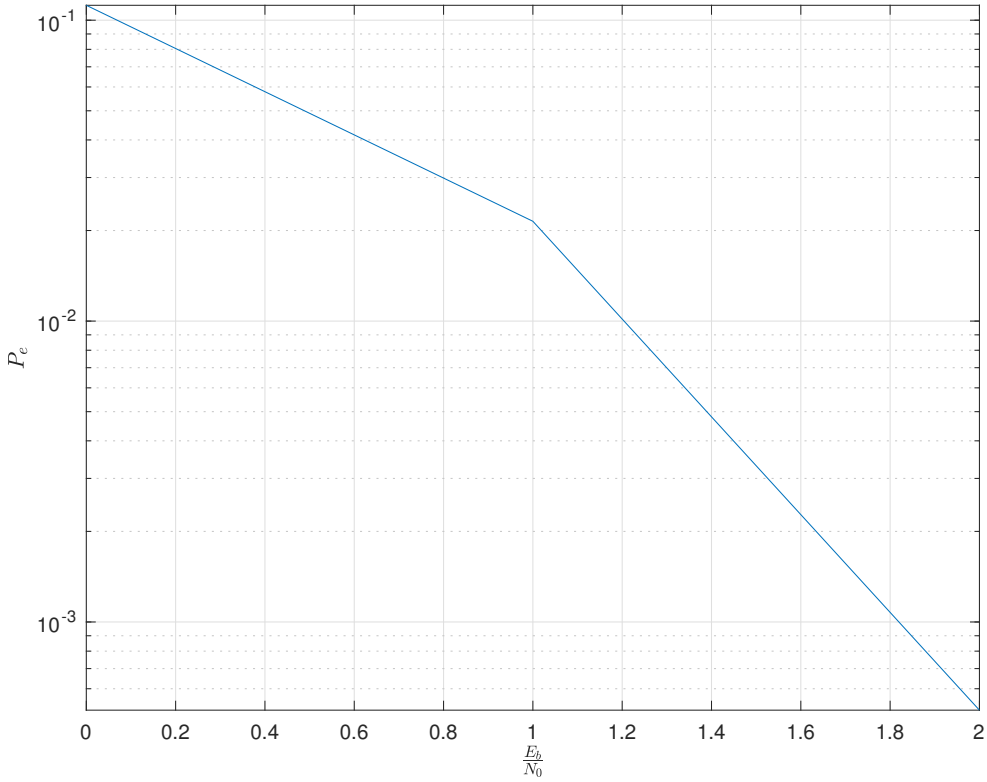


Figure 2.9: SNR vs BER performance of LDPC Codes with code rate $R = \frac{3}{5}$ using QPSK mapping

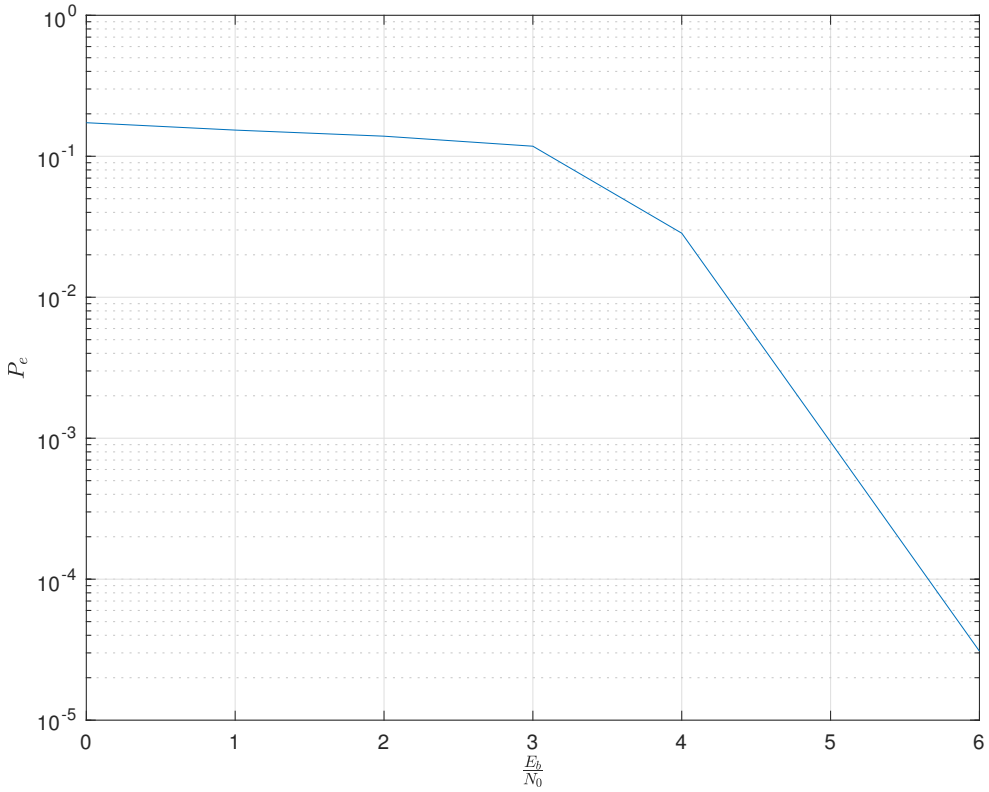


Figure 2.10: SNR vs BER performance of LDPC Codes code rate $R = \frac{3}{5}$ using 8-PSK mapping

Chapter 3

Mapping Schemes

3.1 Phase Shift Keying (PSK)

Let

$$Y_k = X_k + V_k, \quad k = 1, \ldots, N \tag{3.1}$$

where X_k is the transmitted symbol from the constellation symbols of specified mapping $\{X\}$ in the kth time slot and $V_k \sim \mathcal{N}(0, \sigma^2)$.

3.1.1 QPSK

Constellation Mapping symbol set $\{X\}$ is generated by

$$X_k \in \left\{ e^{j\frac{2\pi n}{4}} \right\} \quad n = 0, 1, 2, 3 \tag{3.2}$$

Demapping can be done by using,

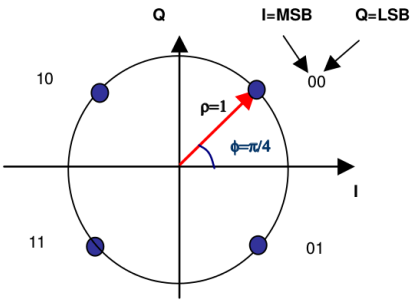


Figure 3.1: Constellation diagram of QPSK

$$\frac{2\pi}{4}i < \angle Y_k < \frac{2\pi}{4}(i+1) \implies \hat{X_k} = X_i \quad i = 0, \ldots, 3 \tag{3.3}$$

Fig. 3.1 Shows the Constellation mapping for QPSK scheme and similarly Fig. 3.2 Shows the Simulation diagram.

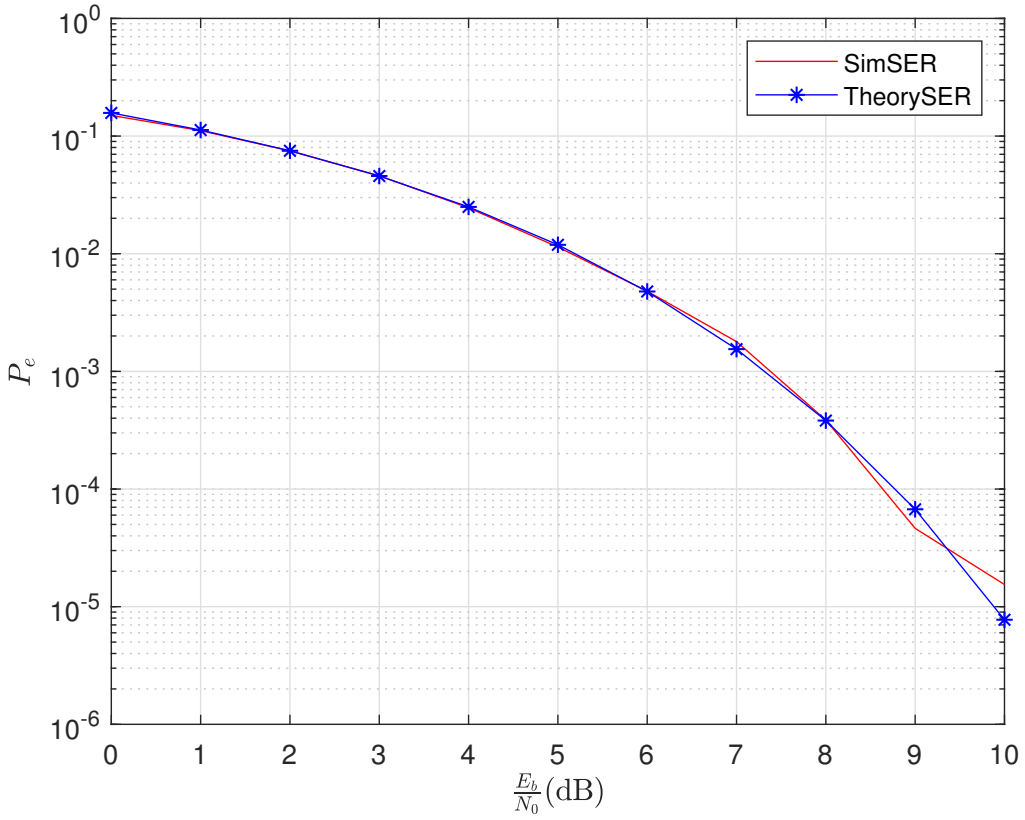


Figure 3.2: SNR vs BER for QPSK

3.1.2 8PSK

Constellation Mapping symbol set $\{X\}$ is generated by

$$X_k \in \left\{ e^{j\frac{2\pi n}{8}} \right\} \quad n = 0, 1, \ldots, 7 \tag{3.4}$$

Demapping can be done by using,

$$\frac{2\pi}{8}i < \angle Y_k < \frac{2\pi}{8}(i+1) \implies \hat{X_k} = X_i \quad i = 0, \ldots, 7 \tag{3.5}$$

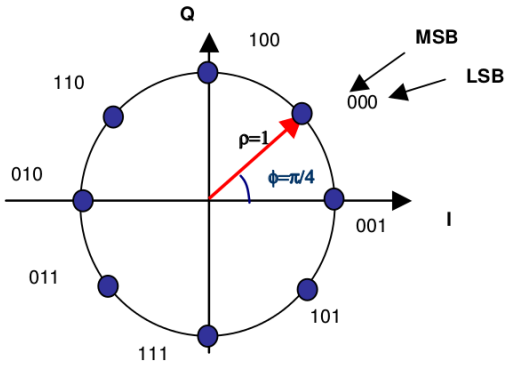


Figure 3.3: Constellation diagram of 8-PSK

Fig. 3.3 Shows the Constellation mapping for 8-PSK symbols and Fig. 3.4 Shows the Simulation diagram.

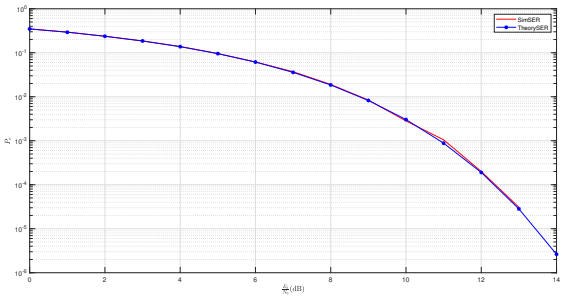


Figure 3.4: SNR vs BER for 8-PSK

3.2 Amplitude Phase Shift Keying (APSK)

3.2.1 16-APSK

Constellation Mapping symbol set $\{X\}$ is generated by

$$X_k \in \{X\} = \begin{cases} r_1 e^{j\left(\phi_1 + \frac{2\pi}{4}n\right)} & n = 0, \ldots, 3 \\ r_2 e^{j\left(\phi_2 + \frac{2\pi}{12}n\right)} & n = 0, 1, \ldots, 11 \end{cases} \tag{3.6}$$

Demapping can be done by using,

$$|Y_k| < \frac{r_1 + r_2}{2} \&\& \frac{2\pi}{4} i < \angle Y_k < \frac{2\pi}{4}(i+1) \tag{3.7}$$

$$\implies \hat{X}_k = X_i \quad i = 0, \ldots, 3 \tag{3.8}$$

$$|Y_k| > \frac{r_1 + r_2}{2} \&\& \frac{2\pi}{12} i < \angle Y_k < \frac{2\pi}{12}(i+1) \tag{3.9}$$

$$\implies \hat{X}_k = X_i \quad i = 4, \ldots, 15 \tag{3.10}$$

Where $\frac{r_2}{r_1} = 2.6, \phi_1 = 45, \phi_2 = 15$ Fig. 3.5 shows the Constellation mapping for 16-APSK symbols and Fig. 3.6 Shows the Simulation diagram.

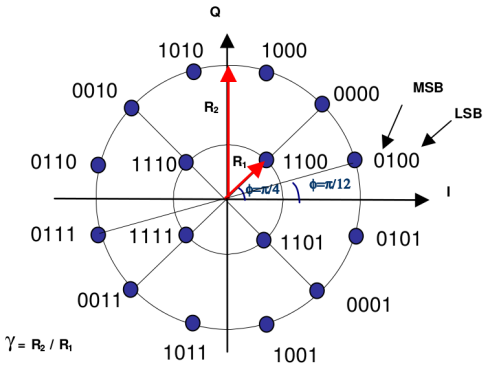


Figure 3.5: Constellation diagram of 16APSK

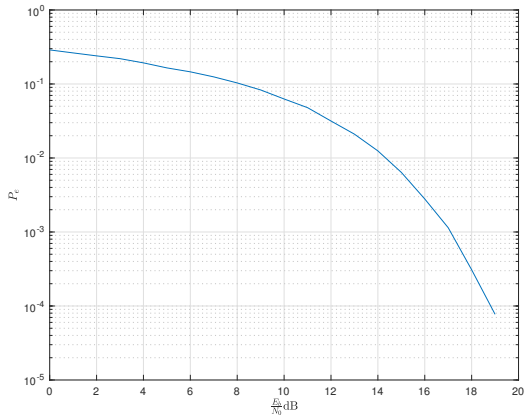


Figure 3.6: SNR vs BER for 16-APSK

3.2.2 32-APSK

Constellation Mapping symbol set $\{X\}$ is generated by

$$X_k \in \{X\} = \begin{cases} r_1 e^{j\left(\phi_1 + \frac{2\pi}{4}n\right)} & n = 0, \ldots, 3 \\ r_2 e^{j\left(\phi_2 + \frac{2\pi}{12}n\right)} & n = 0, 1, \ldots, 11 \\ r_3 e^{j\left(\phi_3 + \frac{2\pi}{16}n\right)} & n = 0, 1, \ldots, 16 \end{cases} \tag{3.11}$$

Where $\frac{r_2}{r_1} = 2.54$, $\frac{r_3}{r_2} = 4.33$, $\phi_1 = 45, \phi_2 = 15, \phi_3 = 0$. Fig. 3.7 shows the Constella-

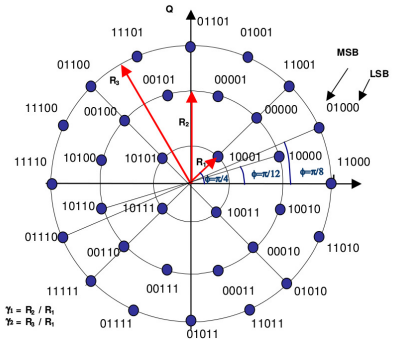


Figure 3.7: Constellation diagram of 32APSK

tion mapping for 32-APSK symbols. Demapping can be done by using,

$$|Y_k| < \frac{r_1 + r_2}{2} \&\& \frac{2\pi}{4} i < \angle Y_k < \frac{2\pi}{4}(i+1) \tag{3.12}$$

$$\implies \hat{X}_k = X_i \quad i = 0, \ldots, 3 \tag{3.13}$$

$$\frac{r_1 + r_2}{2} < |Y_k| < \frac{r_2 + r_3}{2} \&\& \frac{2\pi}{12} i < \angle Y_k < \frac{2\pi}{12}(i+1) \tag{3.14}$$

$$\implies \hat{X}_k = X_i \quad i = 4, \ldots, 15 \tag{3.15}$$

$$|Y_k| > \frac{r_2 + r_3}{2} \&\& \frac{2\pi}{12} i < \angle Y_k < \frac{2\pi}{12}(i+1) \tag{3.16}$$

$$\implies \hat{X}_k = X_i \quad i = 16, \ldots, 31 \tag{3.17}$$

Fig. 3.8 shows the Simulation diagram.

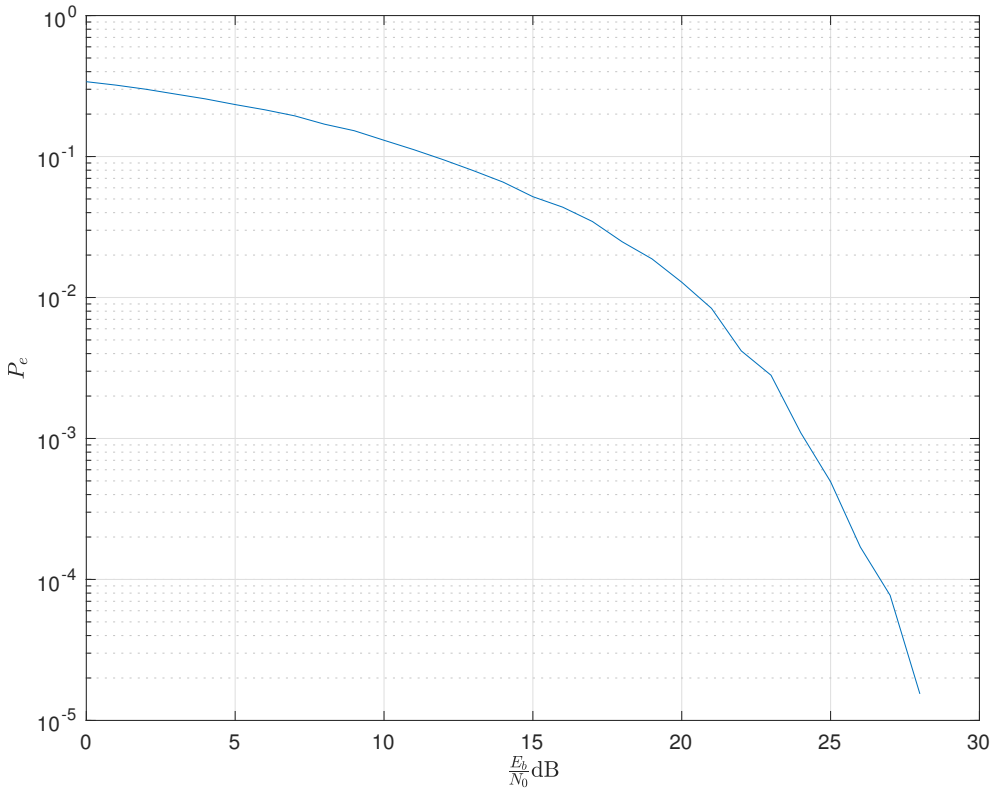


Figure 3.8: SNR vs BER for 32-APSK

Chapter 4

Efficient Transmitter Design Techniques

4.1 Interleaver/Deinterleaver

For 8PSK, 16APSK, and 32APSK mapping schemes, a block interleaver [1] is used to mitigate the effects of bursty channel. For Concatenated Channel coding schemes bit interleaving is necessary. The mapped data is serially put as column wise and searially read out row-wise. Fig. 4.1 shows bit interleaving scheme for 8PSK.

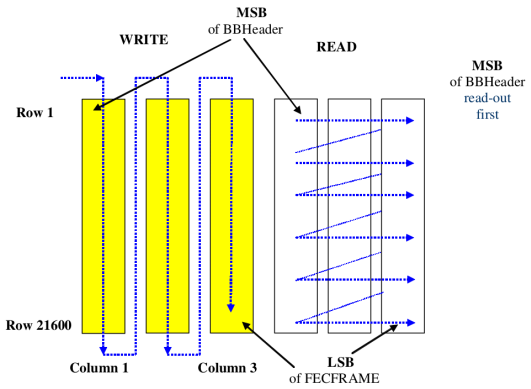


Figure 4.1: Bit Interleaver Structure for 8PSK mapping scheme

Fig. 4.2 shows the BER comparison of 8PSK mapping scheme with and without interleaver.

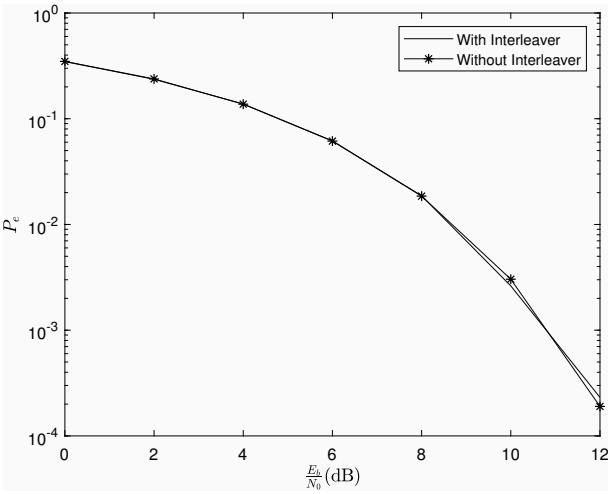


Figure 4.2: Bit interleaver for 8PSK

4.2 Physical Layer Framing(PLFRAMING)

PLFRAMING is useful for specifying the modulation scheme, code rate and frame characteristics, useful for Frame synchronization at the receiver. Fig. 4.3 shows the Typical Structure of PLFRAME according to [1]. *SOF*=Starting Of Frame and *PLSC*=Physical Layer Signalling Code.

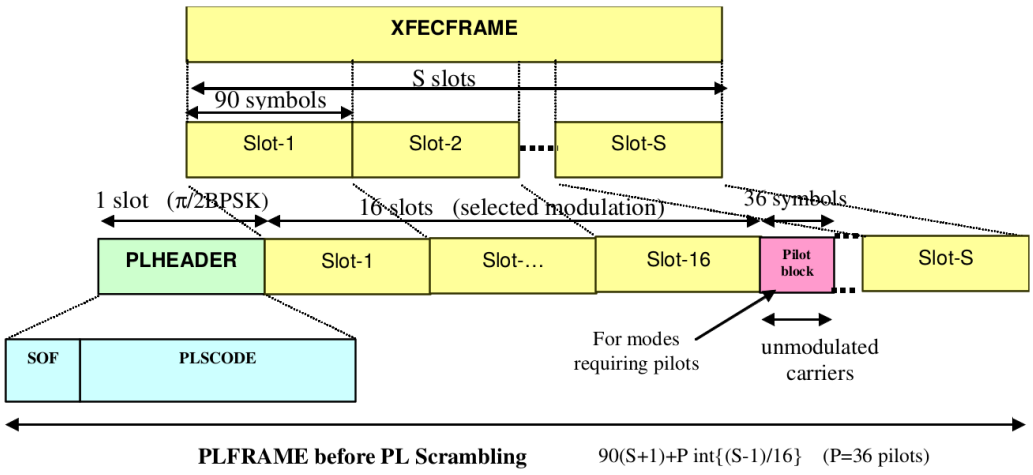


Figure 4.3: Structure of PLFRAME.

4.2.1 Generation of SOF

SOF constitute fixed sequence $18D2E82_{HEX}$ (length=26 bits) in binary format which is from right to left.

4.2.2 Generation of PLSC

- Generation of PLSC involves defining 7 symbols and multiplying first 6 symbols with the defined G matrix in [1]. First 5 symbols called as MODCOD filed and next 2 symbols as TYPE field.

- First 5 symbols represents MODCOD which specifies Frame's mapping scheme and code rate. Fig. 4.5 shows the MODCOD coding for various mapping schemes.

- Next 2 symbols represents TYPE filed which specifies Frame length and presence and absence of pilot fields. This is shown in Table 4.1.

- Similarly, Fig. 4.4 shows the generation of 64 bit using the MODCOD and TYPE bits.

- After the generation of PLS code, we will again scramble the PLS Code with the fixed SCR sequence which is defined in [1].

$$PLSC = PLSC \oplus SCR \tag{4.1}$$

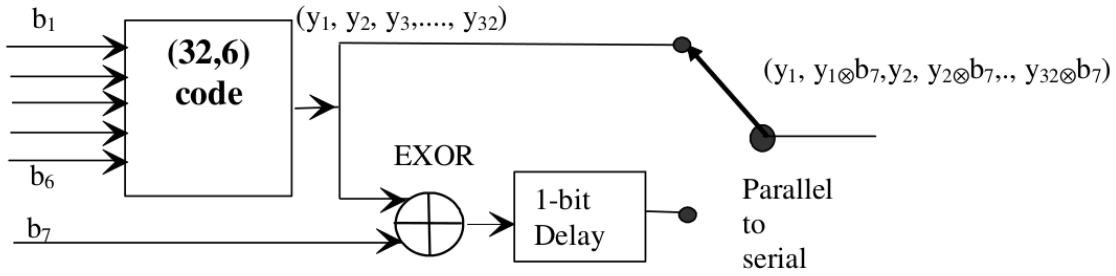


Figure 4.4: Physical Layer Signalling generation

Subscript D denotes decimal e.g. $1_D = 00001$ in binary

Bit-6	Bit-7	Frame Type	Pilots
0	0	Normal	No
0	1	Normal	Yes
1	0	Short	No
1	1	Short	Yes

Table 4.1: Frame Type

Mode	MOD COD	Mode	MOD COD	Mode	MOD COD	Mode	MOD COD
QPSK 1/4	1_D	QPSK 5/6	9_D	8PSK 9/10	17_D	32APSK 4/5	25_D
QPSK 1/3	2_D	QPSK 8/9	10_D	16APSK 2/3	18_D	32APSK 5/6	26_D
QPSK 2/5	3_D	QPSK 9/10	11_D	16APSK 3/4	19_D	32APSK 8/9	27_D
QPSK 1/2	4_D	8PSK 3/5	12_D	16APSK 4/5	20_D	32APSK 9/10	28_D
QPSK 3/5	5_D	8PSK 2/3	13_D	16APSK 5/6	21_D	Reserved	29_D
QPSK 2/3	6_D	8PSK 3/4	14_D	16APSK 8/9	22_D	Reserved	30_D
QPSK 3/4	7_D	8PSK 5/6	15_D	16APSK 9/10	23_D	Reserved	31_D
QPSK 4/5	8_D	8PSK 8/9	16_D	32APSK 3/4	24_D	DUMMY PLFRAME	0_D

Figure 4.5: MODCOD coding for various mapping schemes.

4.2.3 PLHEADER

The PLHEADER has SOF (26 symbols) and PLSC (64 symbols) sequences, that are modulated by rotating QPSK by $\frac{\pi}{4}$. This is also known as $\frac{\pi}{2}$-BPSK.

$$I_{2i-1} = Q_{2i-1} = \frac{1}{\sqrt{2}}(1 - 2y_{2i-1}) \quad i = 1, 2, \ldots, 45 \tag{4.2}$$

$$I_{2i} = -Q_{2i} = -\frac{1}{\sqrt{2}}(1 - 2y_{2i-1}) \quad 1, 2 \ldots, 45 \tag{4.3}$$

where $y_i \in \{0, 1\}, i = 1, \ldots, 90$.

4.2.4 Generation of Pilots

A Pilot block consists of $P = 36$ symbols. Each pilot is composed of un-modulated complex symbol.Where, $I = Q = \frac{1}{\sqrt{2}}$

4.2.5 PLFRAME Calculations

PLFRAME calculations are available in Tables 4.2 and 4.3. The PLFRAME length is calculated as

$$L = SOF + PLSC + (90 * S_{SLOTS}) + (36) * P_{SLOTS} \tag{4.4}$$

where,

$$S_{SLOTS} = \frac{N}{90 \times \log_2 M}, P_{SLOTS} = \left\lfloor \frac{(S_{SLOTS} - 1)}{16} \right\rfloor$$

N = Frame Type(64800/16200 bits),M = Mapping order(4/8/16/32), depending upon the modulation scheme.

	SHORT FRAME (16200 bits)								
PLFRAME	SOF	PLSC	Slot-i	S_{SLOTS}	Pilot Symbols	P_{SLOTS}	#Pilots	Without Pilots	With Pilots
QPSK	26	64	90	90	36	5	180	8190	8370
8PSK	26	64	90	60	36	3	108	5490	5598
16APSK	26	64	90	45	36	2	72	4140	4212
32APSK	26	64	90	36	36	2	72	3330	3402

Table 4.2: Short frame details.

	NORMAL FRAME (64800 bits)								
PLFRAME	SOF	PLSC	Slot-i	S_{SLOTS}	Pilot Symbols	P_{SLOTS}	#Pilots	Without Pilots	With Pilots
QPSK	26	64	90	360	36	22	792	32490	33282
8PSK	26	64	90	240	36	14	504	21690	22194
16APSK	26	64	90	180	36	11	396	16290	16686
32APSK	26	64	90	144	36	8	288	13050	13338

Table 4.3: Long frame details.

4.3 Pulse Shaping

Let X_k be the modulated symbol in the kth time slot. Then the mth sample of the transmitted symbol in the kth time slot is

$$S_k(m) = h_k(m) * X_k \tag{4.5}$$

where $h_k(m), m = 1, \ldots, M; k = 1, \ldots, N$ are samples of the pulse shaping filter [1] obtained from

$$H(f) = \begin{cases} 1 & |f| < f_N(1-\alpha) \\ \left\{\frac{1}{2} + \frac{1}{2}\sin\frac{\pi}{2}\left[\frac{f_N - |f|}{\alpha}\right]\right\}^{\frac{1}{2}} & |f| = f_N(1-\alpha) \\ 0 & |f| > f_N(1+\alpha) \end{cases} \tag{4.6}$$

And comes to be,

$$h_k(m) = \begin{cases} 1 - \alpha + \frac{4}{\pi} & m = 0 \\ \frac{\alpha}{\sqrt{(2)}}[(1 + \frac{2}{\pi})sin(\frac{\pi}{4\alpha}) + (1 - \frac{2}{\pi})cos(\frac{\pi}{4\alpha})] & m = \pm\frac{1}{4\alpha} \\ \frac{sin(\pi\frac{m}{T}(1-\alpha)) + 4\alpha\frac{m}{T}cos(\pi\frac{m}{T}(1+\alpha))}{\pi\frac{m}{T}(1-[4\alpha\frac{m}{T}]^2)} & else \end{cases} \tag{4.7}$$

where f_N is the Nyquist frequency and $\alpha = 0.35, 0.25$ or 0.2. Let the mth received sample in the kth time slot be $Y_k(m)$. At the Receiver, the symbol to be demodulated is then obtained as

$$Y_k(m) * h_k(M - m) \tag{4.8}$$

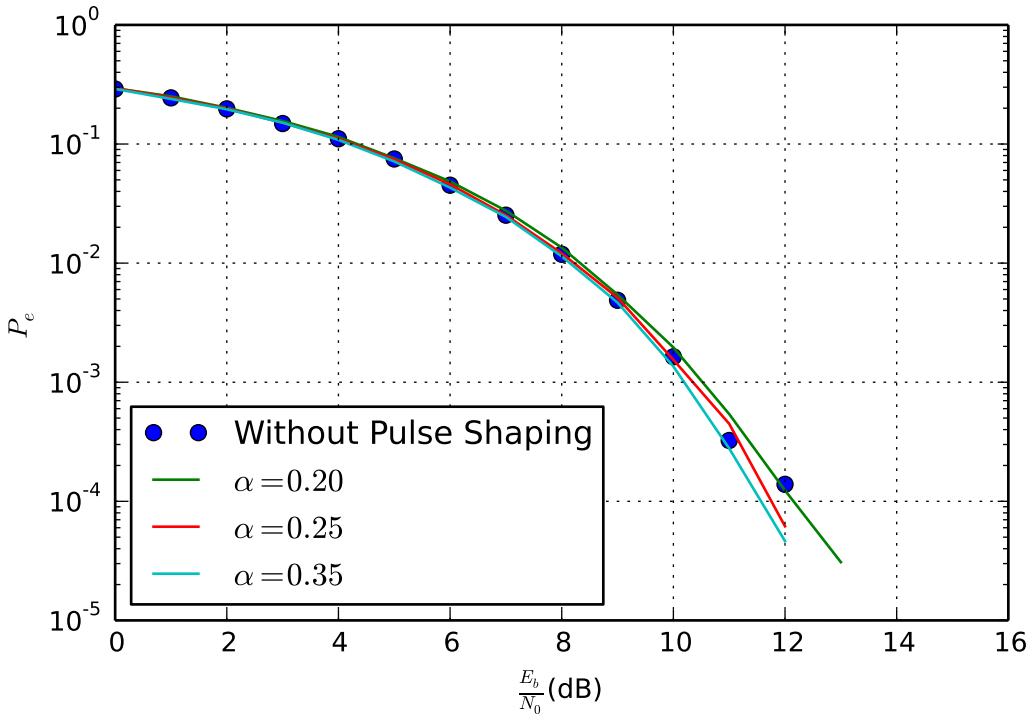


Figure 4.6: SER comparison for QPSK with and without the pulse in (4.6).

Chapter 5

Synchronization Techniques

5.1 Time Offset: Gardner TED

Let the mth sample in the rth received symbol time slot be

$$Y_k(m) = X_k + V_k(m), \quad k = 1, \ldots, N, m = 1, \ldots, M. \tag{5.1}$$

where X_k is the transmitted symbol in the kth time slot and $V_k(m) \sim \mathcal{N}(0, \sigma^2)$. The decision variable for the kth symbol is [10]

$$U_k = \frac{1}{N} \sum_{i=1}^{N} Y_{k-i} \left(\frac{M}{2} \right) \left[Y_{k-i+1}(M) - Y_{k-i}(M) \right] \tag{5.2}$$

5.1.1 Plots

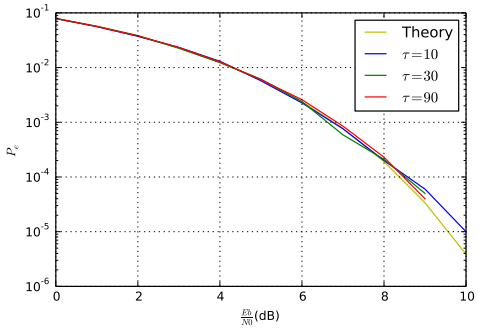


Figure 5.1: SNR vs BER for varying τ.

Fig. 5.1 and shows the variation of the BER with respect to the SNR with different

timing offsets τ for $N = 6$.

5.2 Frequency Offset: LR Technique

Let the frequency offset be Δf [11] . Then

$$Y_k = X_k e^{j2\pi\Delta f k M} + V_k, \quad k = 1, \ldots, N \tag{5.3}$$

From (5.3),

$$Y_k X_k^* = |X_k|^2 e^{j2\pi\Delta f k M} + X_k^* V_k \tag{5.4}$$

$$\implies r_k = e^{j2\pi\Delta f k M} + \bar{V}_k \tag{5.5}$$

where

$$r_k = Y_k X_k^*, \bar{V}_k = X_k^* V_k, |X_k|^2 = 1 \tag{5.6}$$

The autocorrelation can be calculated as

$$R(k) \triangleq \frac{1}{N-k} \sum_{i=k+1}^{N} r_i r_{i-k}^*, 1 \leq k \leq N-1 \tag{5.7}$$

Where N is the length of the received signal. For large centre frequency, the following yields a good approximation for frequency offset upto 40 MHz.

$$\Delta\hat{f} \approx \frac{1}{2\pi M} \frac{\sum_{k=1}^{P} \mathrm{Im}(R(k))}{\sum_{k=1}^{P} k\mathrm{Re}(R(k))}, \quad P\Delta f M << 1 \tag{5.8}$$

where P is the number of pilot symbols.

5.2.1 Plots

The number of pilot symbols is $P = 18$.

Fig. 5.2 shows the variation of the error in the offset estimate with respect to the offset Δf when the SNR = 10 dB. Similarly Fig. 5.3 shows the variation of the error with respect to the SNR for $\Delta f = 5$MHz.

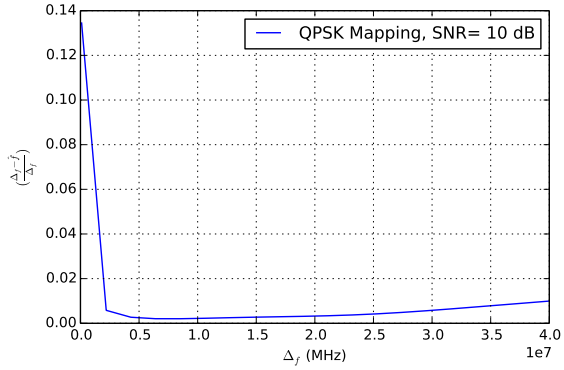


Figure 5.2: Error variation with respect to frequency offset.

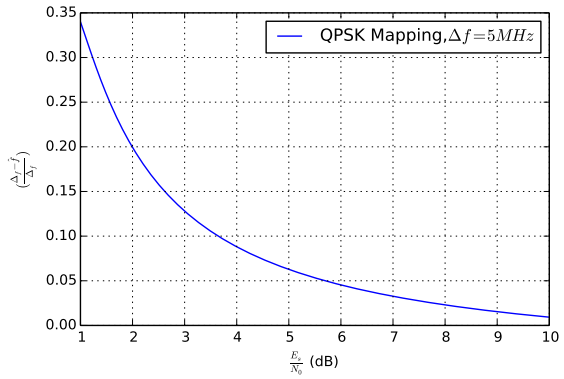


Figure 5.3: Error variation with respect to the SNR. $\Delta f = 5$ MHz $f_c = 25$ GHz

5.3 Phase Offset: Feed Forward Maximum Likelihood (FF-ML)technique

Let the phase offset be $\Delta\phi$ [12] . Then for the kth pilot,

$$Y_k = X_k e^{j\Delta\phi_k} + V_k, \quad k = 1, \ldots, P \tag{5.9}$$

From (5.9),

$$Y_k X_k^* = |X_k|^2 e^{j\Delta\phi_k} + X_k^* V_k \tag{5.10}$$

$$\implies r_k = e^{j\Delta\phi_k} + \bar{V}_k \tag{5.11}$$

where

$$r_k = Y_k X_k^*, \bar{V}_k = X_k^* V_k, |X_k|^2 = 1 \tag{5.12}$$

From (5.11), the estimate for the kth pilot is obtained as

$$\Delta\hat{\phi}_k = \arg(r_k) \tag{5.13}$$

The phase estimate is then obtained using $\Delta\hat{\phi}_k$ in the following update equation as

$$\Delta\theta_k = \Delta\theta_{k-1} + \alpha SAW\left[\Delta\hat{\phi}_k - \Delta\theta_{k-1}\right] \tag{5.14}$$

Where SAW is sawtooth non-linearity

$$SAW[\phi] = [\phi]_{-\pi}^{\pi} \tag{5.15}$$

and $\alpha \leq 1$. The estimate is then obtained as $\Delta\theta_P$.

5.3.1 Plots

Fig. 5.4 shows the variation of the phase error in the offset estimate with respect to the pilot symbols when the SNR = 10 dB and $\alpha = 0.5$.

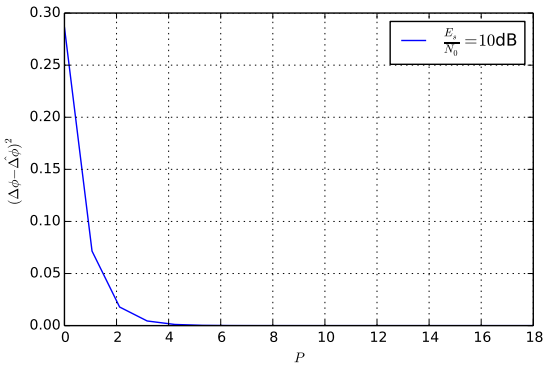


Figure 5.4: Phase error variation with respect to pilot symbols

Similarly Fig. 5.5 shows the variation of the error with respect to the SNR for pilot symbols $P = 18$ and $\alpha = 1$.

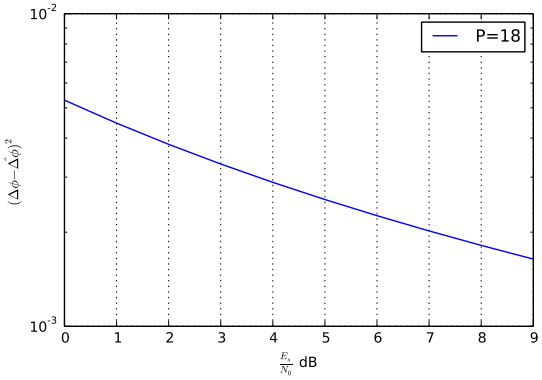


Figure 5.5: $\Delta f = 5$ MHz

5.4 Automatic Gain Controller (AGC): Data-Aided Vector-Tracker (DA-VT)

Let the random AGC offset α, then the received symbol equation with amplitude offset as

$$Y_k = \alpha X_k + V_k \quad k = 1, \ldots, P \tag{5.16}$$

where $\alpha = \alpha_I + j\alpha_Q$ is the gain parameter. According to [13], the $\hat{\alpha_k}$ estimate for the kth pilot is

$$\alpha_{k+1} = \alpha_k - \gamma \left[\alpha_k Y_k^p - X_k^p\right] \left[X_k^p\right]^*, \tag{5.17}$$

where γ is the AGC step size.

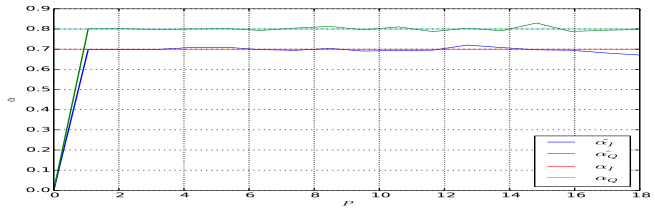


Figure 5.6: Convergence of Digital AGC with resepct to P.

5.4.1 Plots

The following code plots the real and imaginary parts of the gain parameter α with respect to the number of pilot symbols P. in Fig. 5.6. $\gamma = 10^{-3}, SNR = 10dB$.

5.5 Frame Synchronization : Global Summation of SOF/PLSC Detectors

Let the frequency offset be Δf and phase offset be $\Delta\phi$.Then,

$$Y_k = X_k e^{j(2\pi\Delta f k M + \phi_k)} + V_k, \quad k = 1, \ldots, N \tag{5.18}$$

assuming that no pilot symbols are trasmitted. Let the phase information be θ_k, and defined as

$$\theta(k) = \frac{Y_k}{|Y_k|} \tag{5.19}$$

At the receiver, the header information is available in the form of

$$g_i(l) = x_s(l)x_s(l-i), l = 0, \ldots, SOF-1 \tag{5.20}$$

$$h_i(l) = x_p(l)x_p(l-i), l = 0, \ldots, PLSC-1 \tag{5.21}$$

where x_s are the mapped SOF symbols, x_p are the scrambled PLSC symbols, both modulated using $\pi/2$ BPSK for $i = 1, 2, 4, 8, 16, 32$. A special kind of correlation is performed to obtain

$$m_i(k) = \sum_{l=0}^{PLSC-1} e^{j(\theta(k-l)-\theta(k-l-i))} h_i(l), \tag{5.22}$$

$$n_i(k) = \sum_{l=0}^{SOF-1} e^{j(\theta(k-l)-\theta(k-l-i))} g_i(l), \tag{5.23}$$

$$k = 1, \ldots, N \tag{5.24}$$

Compute

$$p_i(k) = \begin{cases} \max\left(|n_i(k-PLSC) + m_i(k)|\right., \\ |n_i(k-PLSC) - m_i(k)|) & k > PLSC \\ \max|m_i(k)| & k < 64 \end{cases} \tag{5.25}$$

GLOBAL variable $G_{R,T}(k)$ [14] defined as,

$$G_{R,T}(k) = \sum_{i \geq 1} p_i(k), \quad i = 1, 2, 4, 8, 16, 32 \tag{5.26}$$

At the receiver, let us consider we have sent two types of transmission. One is PLHEADER+DATA (Y_{k1}) and another is only DATA (Y_{k2}) and the GLOBAL variables for (Y_{k1}) and (Y_{k2}) from (5.26) are $G1_{R,T}(k)$, $G2_{R,T}(k)$ respectively.

5.5.1 Global Threshold Calculation

The Global Threshold variable is defined as

$$T = \max\left(\max\left(G1_{R,T}(k)\right), \max\left(G2_{R,T}(k)\right)\right) \tag{5.27}$$

The probability of false detection of plheader when only the DATA frame (Y_{k2}) has been sent is defined as

$$P_{FA} = \frac{\sum \frac{sign(|Y_{k2}-T|)+1}{2}}{N} \tag{5.28}$$

The probability of missed detection of plheader when PLHEADER+DATA (Y_{k1}) has been sent is defined as

$$P_{MD} = \frac{\sum \frac{sign(T-|Y_{k1}|)+1}{2}}{N + PLSC + SOF} \tag{5.29}$$

5.5.2 Plots

Fig.5.7 shows the ROC curve ($P_{FA}vsP_{MD}$) at the receiver for frame synchronization at $\frac{E_b}{N_0} = -2$ dB.

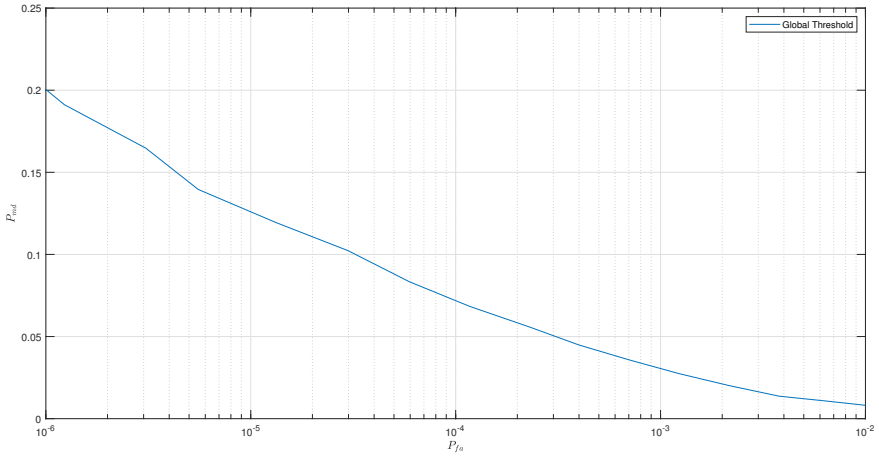


Figure 5.7: Frame Synchronization Receiver Operating Characteristcs (ROC)

Chapter 6

Conclusion and Future Scope

In this thesis, Every module and its submodules have been simulated and verified with standard results [1]. For every sub module, the key equation from various papers has been given as more readable and understandable. Especially, In the case of APSK mapping schemes rather than going for the nearest neighbourhood method by analyzing the phase and amplitude structure of constellation, we presented optimized way demapping.

6.1 Future Work/Scope

- In LDPC Coding, for APSK mapping need to generalize the Log Likelihood Ratio calculation and need to estimate the error performance for various code rates and various frame length using simulation.

- In Time Synchronization, there is a possibility of estimating the knowledge of previous symbol knowledge using the Error probability calculation and by intelligently studying the Gardner Timing Error Detector equation [10].

- In Frequency synchronization [11] and Phase Synchronization block [12], there is the possibility of differentiating the performance with pilots and without pilots scenarios.

- Cascading whole modules and its submodules in programming or implementing in the hardware platform or using any open source platform like GNU radio will give more insights and problems in every module block.

Chapter 7

AppendixA :Parity Check matrix indices of Various Code rates for Short Frame ($16200bits$)

6295	9626	304	7695	4839	4936	1660	144	11203	5567	6347	12557
10691	4988	3859	3734	3071	3494	7687	10313	5964	8069	8296	11090
10774	3613	5208	11177	7676	3549	8746	6583	7239	12265	2674	4292
11869	3708	5981	8718	4908	10650	6805	3334	2627	10461	9285	11120
7844	3079	10773									
3385	10854	5747									
1360	12010	12202									
6189	4241	2343									
9840	12726	4977									

Table 7.2: Short Frame,Code Rate $R = \frac{1}{4}$

416	8909	4156	3216	3112	2560	2912	6405	8593	4969	6723	6912
8978	3011	4339	9312	6396	2957	7288	5485	6031	10218	2226	3575
3383	10059	1114	10008	10147	9384	4290	434	5139	3536	1965	2291
2797	3693	7615	7077	743	1941	8716	6215	3840	5140	4582	5420
6110	8551	1515	7404	4879	4946	5383	1831	3441	9569	10472	4306
1505	5682	7778									
7172	6830	6623									
7281	3941	3505									
10270	8669	914									
3622	7563	9388									
9930	5058	4554									
4844	9609	2707									
6883	3237	1714									
4768	3878	10017									
10127	3334	8267									

Table 7.4: Short Frame ,Code Rate $R = \frac{1}{3}$

5650	4143	8750	583	6720	8071	635	1767	1344	6922	738	6658
5696	1685	3207	415	7019	5023	5608	2605	857	6915	1770	8016
3992	771	2190	7258	8970	7792	1802	1866	6137	8841	886	1931
4108	3781	7577	6810	9322	8226	5396	5867	4428	8827	7766	2254
4247	888	4367	8821	9660	324	5864	4774	227	7889	6405	8963
9693	500	2520	2227	1811	9330	1928	5140	4030	4824	806	3134
1652	8171	1435									
3366	6543	3745									
9286	8509	4645									
7397	5790	8972									
6597	4422	1799									
9276	4041	3847									
8683	7378	4946									
5348	1993	9186									
6724	9015	5646									
4502	4439	8474									
5107	7342	9442									
1387	8910	2660									

Table 7.6: Short Frame ,Code Rate $R = \frac{2}{5}$

20	712	2386	6354	4061	1062	5045	5158
21	2543	5748	4822	2348	3089	6328	5876
22	926	5701	269	3693	2438	3190	3507
23	2802	4520	3577	5324	1091	4667	4449
24	5140	2003	1263	4742	6497	1185	6202
0	4046	6934					
1	2855	66					
2	6694	212					
3	3439	1158					
4	3850	4422					
5	5924	290					
6	1467	4049					
7	7820	2242					
8	4606	3080					
9	4633	7877					
10	3884	6868					
11	8935	4996					
12	3028	764					
13	5988	1057					
14	7411	3450					

Table 7.8: Short Frame,Code Rate $R = \frac{1}{2}$

2765	5713	6426	3596	1374	4811	2182	544	3394	2840	4310	771
4951	211	2208	723	1246	2928	398	5739	265	5601	5993	2615
210	4730	5777	3096	4282	6238	4939	1119	6463	5298	6320	4016
4167	2063	4757	3157	5664	3956	6045	563	4284	2441	3412	6334
4201	2428	4474	59	1721	736	2997	428	3807	1513	4732	6195
2670	3081	5139	3736	1999	5889	4362	3806	4534	5409	6384	5809
5516	1622	2906	3285	1257	5797	3816	817	875	2311	3543	1205
4244	2184	5415	1705	5642	4886	2333	287	1848	1121	3595	6022
2142	2830	4069	5654	1295	2951	3919	1356	884	1786	396	4738

0	2161	2653
1	1380	1461
2	2502	3707
3	3971	1057
4	5985	6062
5	1733	6028
6	3786	1936
7	4292	956
8	5692	3417
9	266	4878
10	4913	3247
11	4763	3937
12	3590	2903
13	2566	4215
14	5208	4707
15	3940	3388
16	5109	4556
17	4908	4177

Table 7.10: Short Frame,Code Rate $R = \frac{3}{5}$

0	10491	16043	506	12826	8065	8226	2767	240	18673	9279	10579	20928	
1	17819	8313	6433	6224	5120	5824	12812	17187	9940	13447	13825	18483	
2	17957	6024	8681	18628	12794	5915	14576	10970	12064	20437	4455	7151	
3	19777	6183	9972	14536	8182	17749	11341	5556	4379	17434	15477	18532	
4	4651	19689	1608	659	16707	14335	6143	3058	14618	17894	20684	5306	
5	9778	2552	12096	12369	15198	16890	4851	3109	1700	18725	1997	15882	
6	486	6111	13743	11537	5591	7433	15227	14145	1483	3887	17431	12430	
7	20647	14311	11734	4180	8110	5525	12141	15761	18661	18441	10569	8192	
8	3791	14759	15264	19918	10132	9062	10010	12786	10675	9682	19246	5454	
9	19525	9485	7777	19999	8378	9209	3163	20232	6690	16518	716	7353	
10	4588	6709	20202	10905	915	4317	11073	13576	16433	368	3508	21171	
11	14072	4033	19959	12608	631	19494	14160	8249	10223	21504	12395	4322	
12	13800	14161		45	19644	7428		18	14777	2044			
13	2948	9647		46	16076	3521		19	13920	9900			
14	14693	16027		47	11779	21062		20	452	7374			
15	20506	11082		48	13062	9682		21	18206	9921			
16	1143	9020		49	8934	5217		22	6131	5414			
17	13501	4014		50	11087	3319		23	10077	9726			
18	1548	2190		51	18892	4356		24	12045	5479			
19	12216	21556		52	7894	3898		25	4322	7990			
20	2095	19897		53	5963	4360		26	15616	5550			
21	4189	7958		54	7346	11726		27	15561	10661			
22	15940	10048		55	5182	5609		28	20718	7387			
23	515	12614		56	2412	17295		29	2518	18804			
24	8501	8450		57	9845	20494		30	8984	2600			
25	17595	16784		58	6687	1864		31	6516	17909			
26	5913	8495		59	20564	5216		32	11148	98			
27	16394	10423		0	18226	17207		33	20559	3704			
28	7409	6981		1	9380	8266		34	7510	1569			
29	6678	15939		2	7073	3065		35	16000	11692			
30	20344	12987		3	18252	13437		36	9147	10303			
31	2510	14588		4	9161	15642		37	16650	191			
32	17918	6655		5	10714	10153		38	15577	18685			
33	6703	19451		6	11585	9078		39	17167	20917			
34	496	4217		7	5359	9418		40	4256	3391			
35	7290	5766		8	9024	9515		41	20092	17219			
36	10521	8925		9	1206	16354		42	9218	5056			
37	20379	11905		10	14994	1102		43	18429	8472			
38	4090	5838		11	9375	20796		44	12093	20753			
39	19082	17040		12	15964	6027		45	16345	12748			
40	20233	12352		13	14789	6452		46	16023	11095			
41	19365	19546		14	8002	18591		47	5048	17595			
42	6249	19030		15	14742	14089		48	18995	4817			
43	11037	19193		16	253	3045		49	16483	3536			
44	19760	11772		17	1274	19286		50	1439	16148			

51	3661	3039
52	19010	18121
53	8968	11793
54	13427	18003
55	5303	3083
56	531	16668
57	4771	6722
58	5695	7960
59	3589	14630

Table 7.12: Short Frame,Code Rate $R = \frac{2}{3}$

3	3198	478	4207	1481	1009	2616	1924	3437	554	683	1801
4	2681	2135									
5	3107	4027									
6	2637	3373									
7	3830	3449									
8	4129	2060									
9	4184	2742									
10	3946	1070									
11	2239	984									
0	1458	3031									
1	3003	1328									
2	1137	1716									
3	132	3725									
4	1817	638									
5	1774	3447									
6	3632	1257									
7	542	3694									
8	1015	1945									
9	1948	412									
10	995	2238									
11	4141	1907									
0	2480	3079									
1	3021	1088									
2	713	1379									
3	997	3903									
4	2323	3361									
5	1110	986									
6	2532	142									
7	1690	2405									
8	1298	1881									
9	615	174									
10	1648	3112									
11	1415	2808									

Table 7.14: Short Frame,Code Rate $R = \frac{3}{4}$

5	896	1565
6	2493	184
7	212	3210
8	727	1339
9	3428	612
0	2663	1947
1	230	2695
2	2025	2794
3	3039	283
4	862	2889
5	376	2110
6	2034	2286
7	951	2068
8	3108	3542
9	307	1421
0	2272	1197
1	1800	3280
2	331	2308
3	465	2552
4	1038	2479
5	1383	343
6	94	236
7	2619	121
8	1497	2774
9	2116	1855
0	722	1584
1	2767	1881
2	2701	1610
3	3283	1732
4	168	1099
5	3074	243
6	3460	945
7	2049	1746
8	566	1427
9	3545	1168

Table 7.16: Short Frame,Code Rate $R = \frac{4}{5}$

3	2409	499	1481	908	559	716	1270	333	2508	2264	1702	2805
4	2447	1926										
5	414	1224										
6	2114	842										
7	212	573										
0	2383	2112										
1	2286	2348										
2	545	819										
3	1264	143										
4	1701	2258										
5	964	166										
6	114	2413										
7	2243	81										
0	1245	1581										
1	775	169										
2	1696	1104										
3	1914	2831										
4	532	1450										
5	91	974										
6	497	2228										
7	2326	1579										
0	2482	256										
1	1117	1261										
2	1257	1658										
3	1478	1225										
4	2511	980										
5	2320	2675										
6	435	1278										
7	228	503										
0	1885	2369										
1	57	483										
2	838	1050										
3	1231	1990										
4	1738	68										
5	2392	951										
6	163	645										
7	2644	1704										

Table 7.18: Short Frame,Code Rate $R = \frac{5}{6}$

0	1558	712	805
1	1450	873	1337
2	1741	1129	1184
3	294	806	1566
4	482	605	923
0	926	1578	
1	777	1374	
2	608	151	
3	1195	210	
4	1484	692	
0	427	488	
1	828	1124	
2	874	1366	
3	1500	835	
4	1496	502	
0	1006	1701	
1	1155	97	
2	657	1403	
3	1453	624	
4	429	1495	
0	809	385	
1	367	151	
2	1323	202	
3	960	318	
4	1451	1039	
0	1098	1722	
1	1015	1428	
2	1261	1564	
3	544	1190	
4	1472	1246	
0	508	630	
1	421	1704	
2	284	898	
3	392	577	
4	1155	556	
0	631	1000	
1	732	1368	
2	1328	329	
3	1515	506	
4	1104	1172	

Table 7.20: Short Frame,Code Rate $R = \frac{8}{9}$

Chapter 8

AppendixB : Parity Check matrix indices of Various Code rates for Normal Frame ($64800 bits$)

23606	36098	1140	28859	18148	18510	6226	540	42014	20879	23802	47088
16419	24928	16609	17248	7693	24997	42587	16858	34921	21042	37024	20692
1874	40094	18704	14474	14004	11519	13106	28826	38669	22363	30255	31105
22254	40564	22645	22532	6134	9176	39998	23892	8937	15608	16854	31009
8037	40401	13550	19526	41902	28782	13304	32796	24679	27140	45980	10021
40540	44498	13911	22435	32701	18405	39929	25521	12497	9851	39223	34823
15233	45333	5041	44979	45710	42150	19416	1892	23121	15860	8832	10308
10468	44296	3611	1480	37581	32254	13817	6883	32892	40258	46538	11940
6705	21634	28150	43757	895	6547	20970	28914	30117	25736	41734	11392
22002	5739	27210	27828	34192	37992	10915	6998	3824	42130	4494	35739
8515	1191	13642	30950	25943	12673	16726	34261	31828	3340	8747	39225
18979	17058	43130	4246	4793	44030	19454	29511	47929	15174	24333	19354
16694	8381	29642	46516	32224	26344	9405	18292	12437	27316	35466	41992
15642	5871	46489	26723	23396	7257	8974	3156	37420	44823	35423	13541
42858	32008	41282	38773	26570	2702	27260	46974	1469	20887	27426	38553
22152	24261	8297									
19347	9978	27802									
34991	6354	33561									
29782	30875	29523									
9278	48512	14349									
38061	4165	43878									
8548	33172	34410									
22535	28811	23950									
20439	4027	24186									
38618	8187	30947									
35538	43880	21459									
7091	45616	15063									
5505	9315	21908									
36046	32914	11836									
7304	39782	33721									
16905	29962	12980									
11171	23709	22460									
34541	9937	44500									
14035	47316	8815									
15057	45482	24461									
30518	36877	879									
7583	13364	24332									
448	27056	4682									
12083	31378	21670									
1159	18031	2221									
17028	38715	9350									
17343	24530	29574									
46128	31039	32818									
20373	36967	18345									
46685	20622	32806									

Table 8.2: Normal Frame,Code Rate $R = \frac{1}{4}$

34903	20927	32093	1052	25611	16093	16454	5520	506	37399	18518	21120
11636	14594	22158	14763	15333	6838	22222	37856	14985	31041	18704	32910
17449	1665	35639	16624	12867	12449	10241	11650	25622	34372	19878	26894
29235	19780	36056	20129	20029	5457	8157	35554	21237	7943	13873	14980
9912	7143	35911	12043	17360	37253	25588	11827	29152	21936	24125	40870
40701	36035	39556	12366	19946	29072	16365	35495	22686	11106	8756	34863
19165	15702	13536	40238	4465	40034	40590	37540	17162	1712	20577	14138
31338	19342	9301	39375	3211	1316	33409	28670	12282	6118	29236	35787
11504	30506	19558	5100	24188	24738	30397	33775	9699	6215	3397	37451
34689	23126	7571	1058	12127	27518	23064	11265	14867	30451	28289	2966
11660	15334	16867	15160	38343	3778	4265	39139	17293	26229	42604	13486
31497	1365	14828	7453	26350	41346	28643	23421	8354	16255	11055	24279
15687	12467	13906	5215	41328	23755	20800	6447	7970	2803	33262	39843
5363	22469	38091	28457	36696	34471	23619	2404	24229	41754	1297	18563
3673	39070	14480	30279	37483	7580	29519	30519	39831	20252	18132	20010
34386	7252	27526	12950	6875	43020	31566	39069	18985	15541	40020	16715
1721	37332	39953	17430	32134	29162	10490	12971	28581	29331	6489	35383
736	7022	42349	8783	6767	11871	21675	10325	11548	25978	431	24085
1925	10602	28585	12170	15156	34404	8351	13273	20208	5800	15367	21764
16279	37832	34792	21250	34192	7406	41488	18346	29227	26127	25493	7048
39948	28229	24899									
17408	14274	38993									
38774	15968	28459									
41404	27249	27425									
41229	6082	43114									
13957	4979	40654									
3093	3438	34992									
34082	6172	28760									
42210	34141	41021									
14705	17783	10134									
41755	39884	22773									
14615	15593	1642									
29111	37061	39860									
9579	33552	633									
12951	21137	39608									
38244	27361	29417									
2939	10172	36479									
29094	5357	19224									
9562	24436	28637									
40177	2326	13504									
6834	21583	42516									
40651	42810	25709									
31557	32138	38142									
18624	41867	39296									
37560	14295	16245									
6821	21679	31570									
25339	25083	22081									
8047	697	35268									
9884	17073	19995									
26848	35245	8390									
18658	16134	14807									
12201	32944	5035									
25236	1216	38986									
42994	24782	8681									
28321	4932	34249									
4107	29382	32124									
22157	2624	14468									
38788	27081	7936									
4368	26148	10578									
25353	4122	39751									

Table 8.4: Normal Frame ,Code Rate $R = \frac{1}{3}$

31413	18834	28884	947	23050	14484	14809	4968	455	33659	16666	19008
13172	19939	13354	13719	6132	20086	34040	13442	27958	16813	29619	16553
1499	32075	14962	11578	11204	9217	10485	23062	30936	17892	24204	24885
32490	18086	18007	4957	7285	32073	19038	7152	12486	13483	24808	21759
32321	10839	15620	33521	23030	10646	26236	19744	21713	36784	8016	12869
35597	11129	17948	26160	14729	31943	20416	10000	7882	31380	27858	33356
14125	12131	36199	4058	35992	36594	33698	15475	1566	18498	12725	7067
17406	8372	35437	2888	1184	30068	25802	11056	5507	26313	32205	37232
15254	5365	17308	22519	35009	718	5240	16778	23131	24092	20587	33385
27455	17602	4590	21767	22266	27357	30400	8732	5596	3060	33703	3596
6882	873	10997	24738	20770	10067	13379	27409	25463	2673	6998	31378
15181	13645	34501	3393	3840	35227	15562	23615	38342	12139	19471	15483
13350	6707	23709	37204	25778	21082	7511	14588	10010	21854	28375	33591
12514	4695	37190	21379	18723	5802	7182	2529	29936	35860	28338	10835
34283	25610	33026	31017	21259	2165	21807	37578	1175	16710	21939	30841
27292	33730	6836	26476	27539	35784	18245	16394	17939	23094	19216	17432
11655	6183	38708	28408	35157	17089	13998	36029	15052	16617	5638	36464
15693	28923	26245	9432	11675	25720	26405	5838	31851	26898	8090	37037
24418	27583	7959	35562	37771	17784	11382	11156	37855	7073	21685	34515
10977	13633	30969	7516	11943	18199	5231	13825	19589	23661	11150	35602
19124	30774	6670	37344	16510	26317	23518	22957	6348	34069	8845	20175
34985	14441	25668	4116	3019	21049	37308	24551	24727	20104	24850	12114
38187	28527	13108	13985	1425	21477	30807	8613	26241	33368	35913	32477
5903	34390	24641	26556	23007	27305	38247	2621	9122	32806	21554	18685
17287	27292	19033									
25796	31795	12152									
12184	35088	31226									
38263	33386	24892									
23114	37995	29796									
34336	10551	36245									
35407	175	7203									
14654	38201	22605									
28404	6595	1018									
19932	3524	29305									
31749	20247	8128									
18026	36357	26735									
7543	29767	13588									
13333	25965	8463									
14504	36796	19710									
4528	25299	7318									
35091	25550	14798									
7824	215	1248									
30848	5362	17291									
28932	30249	27073									
13062	2103	16206									
7129	32062	19612									
9512	21936	38833									
35849	33754	23450									
18705	28656	18111									
22749	27456	32187									
28229	31684	30160									
15293	8483	28002									
14880	13334	12584									
28646	2558	19687									
6259	4499	26336									
11952	28386	8405									
10609	961	7582									
10423	13191	26818									
15922	36654	21450									
10492	1532	1205									
30551	36482	22153									
5156	11330	34243									
28616	35369	13322									
8962	1485	21186									
23541	17445	35561									
33133	11593	19895									
33917	7863	33651									
20063	28331	10702									
13195	21107	21859									
4364	31137	4804									
5585	2037	4830									
30672	16927	14800									

Table 8.6: Normal Frame,Code Rate $R = \frac{2}{5}$

54	9318	14392	27561	26909	10219	2534	8597
55	7263	4635	2530	28130	3033	23830	3651
56	24731	23583	26036	17299	5750	792	9169
57	5811	26154	18653	11551	15447	13685	16264
58	12610	11347	28768	2792	3174	29371	12997
59	16789	16018	21449	6165	21202	15850	3186
60	31016	21449	17618	6213	12166	8334	18212
61	22836	14213	11327	5896	718	11727	9308
62	2091	24941	29966	23634	9013	15587	5444
63	22207	3983	16904	28534	21415	27524	25912
64	25687	4501	22193	14665	14798	16158	5491
65	4520	17094	23397	4264	22370	16941	21526
66	10490	6182	32370	9597	30841	25954	2762
67	22120	22865	29870	15147	13668	14955	19235
68	6689	18408	18346	9918	25746	5443	20645
69	29982	12529	13858	4746	30370	10023	24828
70	1262	28032	29888	13063	24033	21951	7863
71	6594	29642	31451	14831	9509	9335	31552
72	1358	6454	16633	20354	24598	624	5265
73	19529	295	18011	3080	13364	8032	15323
74	11981	1510	7960	21462	9129	11370	25741
75	9276	29656	4543	30699	20646	21921	28050
76	15975	25634	5520	31119	13715	21949	19605
77	18688	4608	31755	30165	13103	10706	29224
78	21514	23117	12245	26035	31656	25631	30699
79	9674	24966	31285	29908	17042	24588	31857
80	21856	27777	29919	27000	14897	11409	7122
81	29773	23310	263	4877	28622	20545	22092
82	15605	5651	21864	3967	14419	22757	15896
83	30145	1759	10139	29223	26086	10556	5098
84	18815	16575	2936	24457	26738	6030	505
85	30326	22298	27562	20131	26390	6247	24791
86	928	29246	21246	12400	15311	32309	18608
87	20314	6025	26689	16302	2296	3244	19613
88	6237	11943	22851	15642	23857	15112	20947
89	26403	25168	19038	18384	8882	12719	7093
0	14567	24965					
1	3908	100					
2	10279	240					
3	24102	764					
4	12383	4173					
5	13861	15918					
6	21327	1046					
7	5288	14579					

8	28158	8069
9	16583	11098
10	16681	28363
11	13980	24725
12	32169	17989
13	10907	2767
14	21557	3818
15	26676	12422
16	7676	8754
17	14905	20232
18	15719	24646
19	31942	8589
20	19978	27197
21	27060	15071
22	6071	26649
23	10393	11176
24	9597	13370
25	7081	17677
26	1433	19513
27	26925	9014
28	19202	8900
29	18152	30647
30	20803	1737
31	11804	25221
32	31683	17783
33	29694	9345
34	12280	26611
35	6526	26122
36	26165	11241
37	7666	26962
38	16290	8480
39	11774	10120
40	30051	30426
41	1335	15424
42	6865	17742
43	31779	12489
44	32120	21001
45	14508	6996
46	979	25024
47	4554	21896
48	7989	21777
49	4972	20661
50	6612	2730
51	12742	4418
52	29194	595
53	19267	20113

Table 8.8: Normal Frame,Code Rate $R = \frac{1}{2}$

22422	10282	11626	19997	11161	2922	3122	99	5625	17064	8270	179	27	5743	8084
25087	16218	17015	828	20041	25656	4186	11629	22599	17305	22515	6463	28	6770	9548
11049	22853	25706	14388	5500	19245	8732	2177	13555	11346	17265	3069	29	4285	17542
16581	22225	12563	19717	23577	11555	25496	6853	25403	5218	15925	21766	30	13568	22599
16529	14487	7643	10715	17442	11119	5679	14155	24213	21000	1116	15620	31	1786	4617
5340	8636	16693	1434	5635	6516	9482	20189	1066	15013	25361	14243	32	23238	11648
18506	22236	20912	8952	5421	15691	6126	21595	500	6904	13059	6802	33	19627	2030
8433	4694	5524	14216	3685	19721	25420	9937	23813	9047	25651	16826	34	13601	13458
21500	24814	6344	17382	7064	13929	4004	16552	12818	8720	5286	2206	35	13740	17328
22517	2429	19065	2921	21611	1873	7507	5661	23006	23128	20543	19777	36	25012	13944
1770	4636	20900	14931	9247	12340	11008	12966	4471	2731	16445	791	37	22513	6687
6635	14556	18865	22421	22124	12697	9803	25485	7744	18254	11313	9004	38	4934	12587
19982	23963	18912	7206	12500	4382	20067	6177	21007	1195	23547	24837	39	21197	5133
756	11158	14646	20534	3647	17728	11676	11843	12937	4402	8261	22944	40	22705	6938
9306	24009	10012	11081	3746	24325	8060	19826	842	8836	2898	5019	41	7534	24633
7575	7455	25244	4736	14400	22981	5543	8006	24203	13053	1120	5128	42	24400	12797
3482	9270	13059	15825	7453	23747	3656	24585	16542	17507	22462	14670	43	21911	25712
15627	15290	4198	22748	5842	13395	23918	16985	14929	3726	25350	24157	44	12039	1140
24896	16365	16423	13461	16615	8107	24741	3604	25904	8716	9604	20365	45	24306	1021
3729	17245	18448	9862	20831	25326	20517	24618	13282	5099	14183	8804	46	14012	20747
16455	17646	15376	18194	25528	1777	6066	21855	14372	12517	4488	17490	47	11265	15219
1400	8135	23375	20879	8476	4084	12936	25536	22309	16582	6402	24360	48	4670	15531
25119	23586	128	4761	10443	22536	8607	9752	25446	15053	1856	4040	49	9417	14359
377	21160	13474	5451	17170	5938	10256	11972	24210	17833	22047	16108	50	2415	6504
13075	9648	24546	13150	23867	7309	19798	2988	16858	4825	23950	15125	51	24964	24690
20526	3553	11525	23366	2452	17626	19265	20172	18060	24593	13255	1552	52	14443	8816
18839	21132	20119	15214	14705	7096	10174	5663	18651	19700	12524	14033	53	6926	1291
4127	2971	17499	16287	22368	21463	7943	18880	5567	8047	23363	6797	54	6209	20806
10651	24471	14325	4081	7258	4949	7044	1078	797	22910	20474	4318	55	13915	4079
21374	13231	22985	5056	3821	23718	14178	9978	19030	23594	8895	25358	56	24410	13196
6199	22056	7749	13310	3999	23697	16445	22636	5225	22437	24153	9442	57	13505	6117
7978	12177	2893	20778	3175	8645	11863	24623	10311	25767	17057	3691	58	9869	8220
20473	11294	9914	22815	2574	8439	3699	5431	24840	21908	16088	18244	59	1570	6044
8208	5755	19059	8541	24924	6454	11234	10492	16406	10831	11436	9649	60	25780	17387
16264	11275	24953	2347	12667	19190	7257	7174	24819	2938	2522	11749	61	20671	24913
3627	5969	13862	1538	23176	6353	2855	17720	2472	7428	573	15036	62	24558	20591
0	18539	18661		9	11296	18655		18	19775	4247		63	12402	3702
1	10502	3002		10	15046	20659		19	1660	19413		64	8314	1357
2	9368	10761		11	7300	22140		20	4403	3649		65	20071	14616
3	12299	7828		12	22029	14477		21	13371	25851		66	17014	3688
4	15048	13362		13	11129	742		22	22770	21784		67	19837	946
5	18444	24640		14	13254	13813		23	10757	14131		68	15195	12136
6	20775	19175		15	19234	13273		24	16071	21617		69	7758	22808
7	18970	10971		16	6079	21122		25	6393	3725		70	3564	2925
8	5329	19982		17	22782	5828		26	597	19968		71	3434	7769

Table 8.10: Normal Frame,Code Rate $R = \frac{3}{5}$

0	10491	16043	506	12826	8065	8226	2767	240	18673	9279	10579	20928		51	3661	3039
1	17819	8313	6433	6224	5120	5824	12812	17187	9940	13447	13825	18483		52	19010	18121
2	17957	6024	8681	18628	12794	5915	14576	10970	12064	20437	4455	7151		53	8968	11793
3	19777	6183	9972	14536	8182	17749	11341	5556	4379	17434	15477	18532		54	13427	18003
4	4651	19689	1608	659	16707	14335	6143	3058	14618	17894	20684	5306		55	5303	3083
5	9778	2552	12096	12369	15198	16890	4851	3109	1700	18725	1997	15882		56	531	16668
6	486	6111	13743	11537	5591	7433	15227	14145	1483	3887	17431	12430		57	4771	6722
7	20647	14311	11734	4180	8110	5525	12141	15761	18661	18441	10569	8192		58	5695	7960
8	3791	14759	15264	19918	10132	9062	10010	12786	10675	9682	19246	5454		59	3589	14630
9	19525	9485	7777	19999	8378	9209	3163	20232	6690	16518	716	7353				
10	4588	6709	20202	10905	915	4317	11073	13576	16433	368	3508	21171				
11	14072	4033	19959	12608	631	19494	14160	8249	10223	21504	12395	4322				
12	13800	14161		45	19644	7428		18	14777	2044						
13	2948	9647		46	16076	3521		19	13920	9900						
14	14693	16027		47	11779	21062		20	452	7374						
15	20506	11082		48	13062	9682		21	18206	9921						
16	1143	9020		49	8934	5217		22	6131	5414						
17	13501	4014		50	11087	3319		23	10077	9726						
18	1548	2190		51	18892	4356		24	12045	5479						
19	12216	21556		52	7894	3898		25	4322	7990						
20	2095	19897		53	5963	4360		26	15616	5550						
21	4189	7958		54	7346	11726		27	15561	10661						
22	15940	10048		55	5182	5609		28	20718	7387						
23	515	12614		56	2412	17295		29	2518	18804						
24	8501	8450		57	9845	20494		30	8984	2600						
25	17595	16784		58	6687	1864		31	6516	17909						
26	5913	8495		59	20564	5216		32	11148	98						
27	16394	10423		0	18226	17207		33	20559	3704						
28	7409	6981		1	9380	8266		34	7510	1569						
29	6678	15939		2	7073	3065		35	16000	11692						
30	20344	12987		3	18252	13437		36	9147	10303						
31	2510	14588		4	9161	15642		37	16650	191						
32	17918	6655		5	10714	10153		38	15577	18685						
33	6703	19451		6	11585	9078		39	17167	20917						
34	496	4217		7	5359	9418		40	4256	3391						
35	7290	5766		8	9024	9515		41	20092	17219						
36	10521	8925		9	1206	16354		42	9218	5056						
37	20379	11905		10	14994	1102		43	18429	8472						
38	4090	5838		11	9375	20796		44	12093	20753						
39	19082	17040		12	15964	6027		45	16345	12748						
40	20233	12352		13	14789	6452		46	16023	11095						
41	19365	19546		14	8002	18591		47	5048	17595						
42	6249	19030		15	14742	14089		48	18995	4817						
43	11037	19193		16	253	3045		49	16483	3536						
44	19760	11772		17	1274	19286		50	1439	16148						

Table 8.12: Normal Frame ,Code Rate $R = \frac{2}{3}$

0	6385	7901	14611	13389	11200	3252	5243	2504	2722	821	7374
1	11359	2698	357	13824	12772	7244	6752	15310	852	2001	11417
2	7862	7977	6321	13612	12197	14449	15137	13860	1708	6399	13444
3	1560	11804	6975	13292	3646	3812	8772	7306	5795	14327	7866
4	7626	11407	14599	9689	1628	2113	10809	9283	1230	15241	4870
5	1610	5699	15876	9446	12515	1400	6303	5411	14181	13925	7358
6	4059	8836	3405	7853	7992	15336	5970	10368	10278	9675	4651
7	4441	3963	9153	2109	12683	7459	12030	12221	629	15212	406
8	6007	8411	5771	3497	543	14202	875	9186	6235	13908	3563
9	3232	6625	4795	546	9781	2071	7312	3399	7250	4932	12652
10	8820	10088	11090	7069	6585	13134	10158	7183	488	7455	9238
11	1903	10818	119	215	7558	11046	10615	11545	14784	7961	15619
12	3655	8736	4917	15874	5129	2134	15944	14768	7150	2692	1469
13	8316	3820	505	8923	6757	806	7957	4216	15589	13244	2622
14	14463	4852	15733	3041	11193	12860	13673	8152	6551	15108	8758
15	3149	11981		0	1022	1264		30	9584	13123	
16	13416	6906		1	12604	9965		31	13987	9597	
17	13098	13352		2	8217	2707		32	15409	12110	
18	2009	14460		3	3156	11793		33	8754	15490	
19	7207	4314		4	354	1514		34	7416	15325	
20	3312	3945		5	6978	14058		35	2909	15549	
21	4418	6248		6	7922	16079		36	2995	8257	
22	2669	13975		7	15087	12138		37	9406	4791	
23	7571	9023		8	5053	6470		38	11111	4854	
24	14172	2967		9	12687	14932		39	2812	8521	
25	7271	7138		10	15458	1763		40	8476	14717	
26	6135	13670		11	8121	1721		41	7820	15360	
27	7490	14559		12	12431	549		42	1179	7939	
28	8657	2466		13	4129	7091		43	2357	8678	
29	8599	12834		14	1426	8415		44	7703	6216	
30	3470	3152		15	9783	7604		0	3477	7067	
31	13917	4365		16	6295	11329		1	3931	13845	
32	6024	13730		17	1409	12061		2	7675	12899	
33	10973	14182		18	8065	9087		3	1754	8187	
34	2464	13167		19	2918	8438		4	7785	1400	
35	5281	15049		20	1293	14115		5	9213	5891	
36	1103	1849		21	3922	13851		6	2494	7703	
37	2058	1069		22	3851	4000		7	2576	7902	
38	9654	6095		23	5865	1768		8	4821	15682	
39	14311	7667		24	2655	14957		9	10426	11935	
40	15617	8146		25	5565	6332		10	1810	904	
41	4588	11218		26	4303	12631		11	11332	9264	
42	13660	6243		27	11653	12236		12	11312	3570	
43	8578	7874		28	16025	7632		13	14916	2650	
44	11741	2686		29	4655	14128		14	7679	7842	

15	6089	13084
16	3938	2751
17	8509	4648
18	12204	8917
19	5749	12443
20	12613	4431
21	1344	4014
22	8488	13850
23	1730	14896
24	14942	7126
25	14983	8863
26	6578	8564
27	4947	396
28	297	12805
29	13878	6692
30	11857	11186
31	14395	11493
32	16145	12251
33	13462	7428
34	14526	13119
35	2535	11243
36	6465	12690
37	6872	9334
38	15371	14023
39	8101	10187
40	11963	4848
41	15125	6119
42	8051	14465
43	11139	5167
44	2883	14521

Table 8.14: Normal Frame ,Code Rate $R = \frac{3}{4}$

0	149	11212	5575	6360	12559	8108	8505	408	10026	12828	27	2139	8757
1	5237	490	10677	4998	3869	3734	3092	3509	7703	10305	28	12004	5948
2	8742	5553	2820	7085	12116	10485	564	7795	2972	2157	29	8704	3191
3	2699	4304	8350	712	2841	3250	4731	10105	517	7516	30	8171	10933
4	12067	1351	11992	12191	11267	5161	537	6166	4246	2363	31	6297	7116
5	6828	7107	2127	3724	5743	11040	10756	4073	1011	3422	32	616	7146
6	11259	1216	9526	1466	10816	940	3744	2815	11506	11573	33	5142	9761
7	4549	11507	1118	1274	11751	5207	7854	12803	4047	6484	34	10377	8138
8	8430	4115	9440	413	4455	2262	7915	12402	8579	7052	35	7616	5811
9	3885	9126	5665	4505	2343	253	4707	3742	4166	1556	0	7285	9863
10	1704	8936	6775	8639	8179	7954	8234	7850	8883	8713	1	7764	10867
11	11716	4344	9087	11264	2274	8832	9147	11930	6054	5455	2	12343	9019
12	7323	3970	10329	2170	8262	3854	2087	12899	9497	11700	3	4414	8331
13	4418	1467	2490	5841	817	11453	533	11217	11962	5251	4	3464	642
14	1541	4525	7976	3457	9536	7725	3788	2982	6307	5997	5	6960	2039
15	11484	2739	4023	12107	6516	551	2572	6628	8150	9852	6	786	3021
16	6070	1761	4627	6534	7913	3730	11866	1813	12306	8249	7	710	2086
17	12441	5489	8748	7837	7660	2102	11341	2936	6712	11977	8	7423	5601
18	10155	4210		9	4846	4435		0	5647	4935	9	8120	4885
19	1010	10483		10	4157	9228		1	4219	1870	10	12385	11990
20	8900	10250		11	12270	6562		2	10968	8054	11	9739	10034
21	10243	12278		12	11954	7592		3	6970	5447	12	424	10162
22	7070	4397		13	7420	2592		4	3217	5638	13	1347	7597
23	12271	3887		14	8810	9636		5	8972	669	14	1450	112
24	11980	6836		15	689	5430		6	5618	12472	15	7965	8478
25	9514	4356		16	920	1304		7	1457	1280	16	8945	7397
26	7137	10281		17	1253	11934		8	8868	3883	17	6590	8316
27	11881	2526		18	9559	6016		9	8866	1224	18	6838	9011
28	1969	11477		19	312	7589		10	8371	5972	19	6174	9410
29	3044	10921		20	4439	4197		11	266	4405	20	255	113
30	2236	8724		21	4002	9555		12	3706	3244	21	6197	5835
31	9104	6340		22	12232	7779		13	6039	5844	22	12902	3844
32	7342	8582		23	1494	8782		14	7200	3283	23	4377	3505
33	11675	10405		24	10749	3969		15	1502	11282	24	5478	8672
34	6467	12775		25	4368	3479		16	12318	2202	25	4453	2132
35	3186	12198		26	6316	5342		17	4523	965	26	9724	1380
0	9621	11445		27	2455	3493		18	9587	7011	27	12131	11526
1	7486	5611		28	12157	7405		19	2552	2051	28	12323	9511
2	4319	4879		29	6598	11495		20	12045	10306	29	8231	1752
3	2196	344		30	11805	4455		21	11070	5104	30	497	9022
4	7527	6650		31	9625	2090		22	6627	6906	31	9288	3080
5	10693	2440		32	4731	2321		23	9889	2121	32	2481	7515
6	6755	2706		33	3578	2608		24	829	9701	33	2696	268
7	5144	5998		34	8504	1849		25	2201	1819	34	4023	12341
8	11043	8033		35	4027	1151		26	6689	12925	35	7108	5553

Table 8.16: Normal Frame,Code Rate $R = \frac{4}{5}$

0	4362	416	8909	4156	3216	3112	2560	2912	6405	8593	4969	6723
1	2479	1786	8978	3011	4339	9313	6397	2957	7288	5484	6031	10217
2	10175	9009	9889	3091	4985	7267	4092	8874	5671	2777	2189	8716
3	9052	4795	3924	3370	10058	1128	9996	10165	9360	4297	434	5138
4	2379	7834	4835	2327	9843	804	329	8353	7167	3070	1528	7311
5	3435	7871	348	3693	1876	6585	10340	7144	5870	2084	4052	2780
6	3917	3111	3476	1304	10331	5939	5199	1611	1991	699	8316	9960
7	6883	3237	1717	10752	7891	9764	4745	3888	10009	4176	4614	1567
8	10587	2195	1689	2968	5420	2580	2883	6496	111	6023	1024	4449
9	3786	8593	2074	3321	5057	1450	3840	5444	6572	3094	9892	1512
10	8548	1848	10372	4585	7313	6536	6379	1766	9462	2456	5606	9975
11	8204	10593	7935	3636	3882	394	5968	8561	2395	7289	9267	9978
12	7795	74	1633	9542	6867	7352	6417	7568	10623	725	2531	9115
13	7151	2482	4260	5003	10105	7419	9203	6691	8798	2092	8263	3755
14	3600	570	4527	200	9718	6771	1995	8902	5446	768	1103	6520
15	6304	7621		15	4506	3491		15	9027	3415		
16	6498	9209		16	8191	4182		16	1690	3866		
17	7293	6786		17	10192	6157		17	2854	8469		
18	5950	1708		18	5668	3305		18	6206	630		
19	8521	1793		19	3449	1540		19	363	5453		
20	6174	7854		20	4766	2697		20	4125	7008		
21	9773	1190		21	4069	6675		21	1612	6702		
22	9517	10268		22	1117	1016		22	9069	9226		
23	2181	9349		23	5619	3085		23	5767	4060		
24	1949	5560		24	8483	8400		24	3743	9237		
25	1556	555		25	8255	394		25	7018	5572		
26	8600	3827		26	6338	5042		26	8892	4536		
27	5072	1057		27	6174	5119		27	853	6064		
28	7928	3542		28	7203	1989		28	8069	5893		
29	3226	3762		29	1781	5174		29	2051	2885		
0	7045	2420		0	1464	3559		0	10691	3153		
1	9645	2641		1	3376	4214		1	3602	4055		
2	2774	2452		2	7238	67		2	328	1717		
3	5331	2031		3	10595	8831		3	2219	9299		
4	9400	7503		4	1221	6513		4	1939	7898		
5	1850	2338		5	5300	4652		5	617	206		
6	10456	9774		6	1429	9749		6	8544	1374		
7	1692	9276		7	7878	5131		7	10676	3240		
8	10037	4038		8	4435	10284		8	6672	9489		
9	3964	338		9	6331	5507		9	3170	7457		
10	2640	5087		10	6662	4941		10	7868	5731		
11	858	3473		11	9614	10238		11	6121	10732		
12	5582	5683		12	8400	8025		12	4843	9132		
13	9523	916		13	9156	5630		13	580	9591		
14	4107	1559		14	7067	8878		14	6267	9290		

15	3009	2268
16	195	2419
17	8016	1557
18	1516	9195
19	8062	9064
20	2095	8968
21	753	7326
22	6291	3833
23	2614	7844
24	2303	646
25	2075	611
26	4687	362
27	8684	9940
28	4830	2065
29	7038	1363
0	1769	7837
1	3801	1689
2	10070	2359
3	3667	9918
4	1914	6920
5	4244	5669
6	10245	7821
7	7648	3944
8	3310	5488
9	6346	9666
10	7088	6122
11	1291	7827
12	10592	8945
13	3609	7120
14	9168	9112
15	6203	8052
16	3330	2895
17	4264	10563
18	10556	6496
19	8807	7645
20	1999	4530
21	9202	6818
22	3403	1734
23	2106	9023
24	6881	3883
25	3895	2171
26	4062	6424
27	3755	9536
28	4683	2131
29	7347	8027

Table 8.18: Normal Frame , Code Rate $R = \frac{5}{6}$

0	6235	2848	3222
1	5800	3492	5348
2	2757	927	90
3	6961	4516	4739
4	1172	3237	6264
5	1927	2425	3683
6	3714	6309	2495
7	3070	6342	7154
8	2428	613	3761
9	2906	264	5927
10	1716	1950	4273
11	4613	6179	3491
12	4865	3286	6005
13	1343	5923	3529
14	4589	4035	2132
15	1579	3920	6737
16	1644	1191	5998
17	1482	2381	4620
18	6791	6014	6596
19	2738	5918	3786
0	5156	6166	
1	1504	4356	
2	130	1904	
3	6027	3187	
4	6718	759	
5	6240	2870	
6	2343	1311	
7	1039	5465	
8	6617	2513	
9	1588	5222	
10	6561	535	
11	4765	2054	
12	5966	6892	
13	1969	3869	
14	3571	2420	
15	4632	981	
16	3215	4163	
17	973	3117	
18	3802	6198	
19	3794	3948	
0	3196	6126	
1	573	1909	
2	850	4034	
3	5622	1601	
4	6005	524	

5	5251	5783
6	172	2032
7	1875	2475
8	497	1291
9	2566	3430
10	1249	740
11	2944	1948
12	6528	2899
13	2243	3616
14	867	3733
15	1374	4702
16	4698	2285
17	4760	3917
18	1859	4058
19	6141	3527
0	2148	5066
1	1306	145
2	2319	871
3	3463	1061
4	5554	6647
5	5837	339
6	5821	4932
7	6356	4756
8	3930	418
9	211	3094
10	1007	4928
11	3584	1235
12	6982	2869
13	1612	1013
14	953	4964
15	4555	4410
16	4925	4842
17	5778	600
18	6509	2417
19	1260	4903
0	3369	3031
1	3557	3224
2	3028	583
3	3258	440
4	6226	6655
5	4895	1094
6	1481	6847
7	4433	1932
8	2107	1649
9	2119	2065

10	4003	6388
11	6720	3622
12	3694	4521
13	1164	7050
14	1965	3613
15	4331	66
16	2970	1796
17	4652	3218
18	1762	4777
19	5736	1399
0	970	2572
1	2062	6599
2	4597	4870
3	1228	6913
4	4159	1037
5	2916	2362
6	395	1226
7	6911	4548
8	4618	2241
9	4120	4280
10	5825	474
11	2154	5558
12	3793	5471
13	5707	1595
14	1403	325
15	6601	5183
16	6369	4569
17	4846	896
18	7092	6184
19	6764	7127
0	6358	1951
1	3117	6960
2	2710	7062
3	1133	3604
4	3694	657
5	1355	110
6	3329	6736
7	2505	3407
8	2462	4806
9	4216	214
10	5348	5619
11	6627	6243
12	2644	5073
13	4212	5088
14	3463	3889

15	5306	478
16	4320	6121
17	3961	1125
18	5699	1195
19	6511	792
0	3934	2778
1	3238	6587
2	1111	6596
3	1457	6226
4	1446	3885
5	3907	4043
6	6839	2873
7	1733	5615
8	5202	4269
9	3024	4722
10	5445	6372
11	370	1828
12	4695	1600
13	680	2074
14	1801	6690
15	2669	1377
16	2463	1681
17	5972	5171
18	5728	4284
19	1696	1459

Table 8.20: Normal Frame ,Code Rate $R = \frac{8}{9}$

0	5611	2563	2900
1	5220	3143	4813
2	2481	834	81
3	6265	4064	4265
4	1055	2914	5638
5	1734	2182	3315
6	3342	5678	2246
7	2185	552	3385
8	2615	236	5334
9	1546	1755	3846
10	4154	5561	3142
11	4382	2957	5400
12	1209	5329	3179
13	1421	3528	6063
14	1480	1072	5398
15	3843	1777	4369
16	1334	2145	4163
17	2368	5055	260
0	6118	5405	
1	2994	4370	
2	3405	1669	
3	4640	5550	
4	1354	3921	
5	117	1713	
6	5425	2866	
7	6047	683	
8	5616	2582	
9	2108	1179	
10	933	4921	
11	5953	2261	
12	1430	4699	
13	5905	480	
14	4289	1846	
15	5374	6208	
16	1775	3476	
17	3216	2178	
0	4165	884	
1	2896	3744	
2	874	2801	
3	3423	5579	
4	3404	3552	
5	2876	5515	
6	516	1719	
7	765	3631	
8	5059	1441	

9	5629	598
10	5405	473
11	4724	5210
12	155	1832
13	1689	2229
14	449	1164
15	2308	3088
16	1122	669
17	2268	5758
0	5878	2609
1	782	3359
2	1231	4231
3	4225	2052
4	4286	3517
5	5531	3184
6	1935	4560
7	1174	131
8	3115	956
9	3129	1088
10	5238	4440
11	5722	4280
12	3540	375
13	191	2782
14	906	4432
15	3225	1111
16	6296	2583
17	1457	903
0	855	4475
1	4097	3970
2	4433	4361
3	5198	541
4	1146	4426
5	3202	2902
6	2724	525
7	1083	4124
8	2326	6003
9	5605	5990
10	4376	1579
11	4407	984
12	1332	6163
13	5359	3975
14	1907	1854
15	3601	5748
16	6056	3266
17	3322	4085

0	1768	3244
1	2149	144
2	1589	4291
3	5154	1252
4	1855	5939
5	4820	2706
6	1475	3360
7	4266	693
8	4156	2018
9	2103	752
10	3710	3853
11	5123	931
12	6146	3323
13	1939	5002
14	5140	1437
15	1263	293
16	5949	4665
17	4548	6380
0	3171	4690
1	5204	2114
2	6384	5565
3	5722	1757
4	2805	6264
5	1202	2616
6	1018	3244
7	4018	5289
8	2257	3067
9	2483	3073
10	1196	5329
11	649	3918
12	3791	4581
13	5028	3803
14	3119	3506
15	4779	431
16	3888	5510
17	4387	4084
0	5836	1692
1	5126	1078
2	5721	6165
3	3540	2499
4	2225	6348
5	1044	1484
6	6323	4042
7	1313	5603
8	1303	3496

9	3516	3639
10	5161	2293
11	4682	3845
12	3045	643
13	2818	2616
14	3267	649
15	6236	593
16	646	2948
17	4213	1442
0	5779	1596
1	2403	1237
2	2217	1514
3	5609	716
4	5155	3858
5	1517	1312
6	2554	3158
7	5280	2643
8	4990	1353
9	5648	1170
10	1152	4366
11	3561	5368
12	3581	1411
13	5647	4661
14	1542	5401
15	5078	2687
16	316	1755
17	3392	1991

Table 8.22: Normal Frame,Code Rate $R = \frac{9}{10}$

References

[1] "Digital Video Broadcasting (DVB);Second generation framing structure, channel coding and modulation systems for Broadcasting, Interactive Services, News Gathering and other broadband satellite applications," ETSI, Tech. Rep., 2014. [Online]. Available: http://www.etsi.org/deliver/etsi_en/302300_302399/30230701/01.04.01_60/en_30230701v010401p.pdf

[2] R.Gallager, "Low-density parity check codes," *IRE Trans.Information Theory*, pp. 21–28, Jan. 1962.

[3] N. Wiberg, "Codes and Decoding on General Graphs," *Phd Dissertation,Linkoping University,Sweden*, 1996.

[4] D. J. C. MacKay and R. M. Neal, "Near Shannon limit performance of low density parity check codes," *Electronics Letters*, vol. 32, no. 18, pp. 1645–, Aug 1996. [Online]. Available: https://doi.org/10.1049/el:19961141

[5] T. J. Richardson and R. L. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 599–618, Feb 2001. [Online]. Available: https://doi.org/10.1109/18.910577

[6] P. Savvopoulos, N. Papandreou, and T. Antonakopoulos, "Architecture and dsp implementation of a dvb-s2 baseband demodulator," in *2009 12th Euromicro*

Conference on Digital System Design, Architectures, Methods and Tools, Aug 2009, pp. 441–448. [Online]. Available: https://doi.org/10.1109/DSD.2009.228

[7] M. A. Smadi, "Performance analysis of bpsk system with hard decision (63 , 36) bch code," 2010.

[8] L. L. Joiner and J. J. Komo, "Decoding binary BCH codes," in *Proceedings IEEE Southeastcon '95. Visualize the Future*, March 1995, pp. 67–73. [Online]. Available: https://doi.org/10.1109/SECON.1995.513059

[9] V. B. Olivatto, R. R. Lopes, and E. R. de Lima, "Simplified LLR calculation for DVB-S2 LDPC decoder," in *2015 IEEE International Conference on Communication, Networks and Satellite (COMNESTAT)*, Dec 2015, pp. 26–31. [Online]. Available: https://doi.org/10.1109/COMNETSAT.2015.7434300

[10] F. Gardner, "A BPSK/QPSK Timing-Error Detector for Sampled Receivers," *IEEE Transactions on Communications*, vol. 34, no. 5, pp. 423–429, May 1986. [Online]. Available: https://doi.org/10.1109/TCOM.1986.1096561

[11] M. Luise and R. Reggiannini, "Carrier frequency recovery in all-digital modems for burst-mode transmissions," *IEEE Transactions on Communications*, vol. 43, no. 2/3/4, pp. 1169–1178, Feb 1995. [Online]. Available: https://doi.org/10.1109/26.380149

[12] E. Casini, R. D. Gaudenzi, and A. Ginesi, "DVB-S2 modem algorithms design and performance over typical satellite channels," *International Journal of Satellite Communications and Networking*, vol. 22, no. 3, pp. 281–318, 2004. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/sat.791

[13] R. De Gaudenzi and M. Luise, "Analysis and design of an all-digital demodulator for trellis coded 16-QAM transmission over a nonlinear satellite

channel," *IEEE Transactions on Communications*, vol. 43, no. 2/3/4, pp. 659–668, Feb 1995. [Online]. Available: https://doi.org/10.1109/26.380085

[14] H. Miyashiro, E. Boutillon, C. Roland, J. Vilca, and D. Díaz, "Improved Multiplierless Architecture for Header Detection in DVB-S2 Standard," in *2016 IEEE International Workshop on Signal Processing Systems (SiPS)*, Oct 2016, pp. 248–253. [Online]. Available: https://doi.org/10.1109/SiPS.2016.51