Implementation And System Level Performance Analysis Of NOMA

MTech. Thesis

By

SURABHI SINHA

DEPARTMENT OF ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY INDORE

June 2023

Implementation And System Level Performance Analysis Of NOMA

A THESIS

Submitted in partial fulfilment of the requirements for the award of the degree ***of***
Master of Technology

by

SURABHI SINHA

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY INDORE

JUNE 2023

INDIAN INSTITUTE OF TECHNOLOGY INDORE

CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in the thesis entitled **Implementation And System Level Performance Analysis Of NOMA** in the partial fulfilment of the requirements for the award of the degree of **MASTER OF TECHNOLOGY** and submitted in the **DEPARMENT OF ELECTRICAL ENGINEERING, Indian Institute of Technology Indore**, is an authentic record of my own work carried out during the time period from August 2021 to May 2023 under the supervision of Prof. Vimal Bhatia. The matter presented in this thesis has not been submitted by me for the award of any other degree of this or any other institute.

Surabhi Sinha
25/6/23

Signature of the student with date
(Surabhi Sinha)

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

25.06.2023

Signature of the Supervisor of M.Tech. thesis (with date)
(Prof. Vimal Bhatia)

Surabhi Sinha has successfully given her M.Tech. Oral Examination held on 16th May 2023.

25.06.2023

Signature(s) of Supervisor(s) of M.Tech. thesis
Date:

Convener, DPGC
Date: 30/06/2023

M. Shaikh

Signature of PSPC Member #1
Date: 27/06/2023

Signature of PSPC Member #1
Date:

ACKNOWLEDGEMENTS

I would like to express my gratitude to my supervisor Prof. Vimal Bhatia for his guidance and valuable feedback during the course of my MTech project. I would also like to thank my PSPC members Prof. Mukesh Kumar and Dr. Shaikh M Mobin for their insightful comments and suggestions during my project presentations.

I sincerely acknowledge the encouragement and tremendous support provided by my parents and my brother which helped me stay positive and overcome every hurdle throughout my college journey.

Last but not the least, I would like to thank all my friends in the IITI community, IIT Indore teaching and non-teaching staff and MHRD for supporting my M.Tech. project directly or indirectly. To them, I will forever be grateful.

Abstract

Utilizing non-orthogonal multiple access (NOMA) in the downlink leads to enhancement in the capacity of the communication network and improves its spectral efficiency. It can theoretically service any number of users in each time/frequency slot, in contrast to orthogonal multiple access (OMA) techniques that only serve one user per slot. This technique is of great importance when it comes to 5G and beyond 5G for mobile communication as there is a need for massive connectivity, high spectral efficiency, and high system throughput. Hence the aim of this thesis is to implement NOMA using software defined radio (SDR). We also observe the execution time and memory usage for three distinct channel encoders/decoders used for implementing NOMA namely hamming code, convolutional code and LDPC code. Calculation of the bit error rate of each type of encoder is done to determine the more efficient encoder.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

Introduction

In the context of cellular radio, multiple access refers to a technique by which multiple users can share a common radio resource to establish a communication link with a base station (BS). Some of the widely used multiple access techniques in the past generations of cellular networks include time division multiple access (TDMA), frequency division multiple access (FDMA), and code division multiple access (CDMA) [1][2]. These are referred to as orthogonal multiple access (OMA) techniques. In these techniques the resources are divided orthogonally with no interference between adjacent users. For example in TDMA, the total time slot is divided amongst the total number of users using the channel. However, it poses a problem because the number of orthogonal resources such as time or frequency is limited and can exhaust over time. Moreover, 5G needs to support massive connectivity of users and/or devices to meet the demand for low latency, low-cost devices, and diverse service types. To satisfy these requirements, enhanced technologies are necessary. So far, some potential technologies have been proposed to address challenges of 5G, such as massive MIMO, millimeter wave communications, ultra-dense network, and non-orthogonal multiple access (NOMA) [3].

Non-Orthogonal Multiple Access (NOMA) is an emerging and innovative multiple access technique that has garnered significant interest and research attention in recent years with regard to 5G and beyond 5G wireless communication. The fundamental principle behind NOMA is to exploit the power domain or code domain multiplexing to serve multiple users within the same time and frequency resources[4].

In power domain NOMA, the multiple users are allocated different power coefficients and then superimposed on each other. In order to apply power domain NOMA effectively, an optimum power allocation mechanism is required. In [5], the authors proposed methods for power optimization such that it maximizes the ergodic capacity and proved that NOMA provides significant performance improvement compared to OMA techniques.

Another advantage of using NOMA compared to OMA systems is that NOMA is able to provide higher system throughput. This is because a transmitting modem in NOMA can simultaneously transmit multiple data signals to different users with different power levels while each user, at the same time, occupies the entire available frequency band [6], [7]. However, the receiver complexity increases as we need to separate multiple users by suppressing the interference. Recently, several methods have been developed for cancelling the interference obtained between multiple users in power domain NOMA such as successive interference cancellation (SIC) algorithm, parallel interference cancellation (PIC) and a joint interference cancellation (JIC) [8-11].

1.1 Contribution

The main aim of this thesis is to implement NOMA on Software Defined Radio (SDR) and to measure its over-the-air performance. Here we present profiling results for three distinct channel encoders/decoders used for implementing NOMA namely hamming code, convolutional codes and LDPC code. The profiling results comprise of execution time of each encoder module, execution time of all the sub modules and total execution time. Profiling results for memory usage of each encoder/decoder is also documented which include average bytes used, minimum bytes used, and maximum bytes used. Lastly, bit error rate (BER) for each of the encoding techniques is calculated considering both near and far user in order to determine the more efficient encoder for NOMA.

1.2 Literature Survey

In [12] the authors experimentally evaluated the PHY performance of the IEEE 802.22 communication system. They demonstrated how the USRP can be used to realize software-defined IEEE 802.22 transceiver for prototyping and profiling of components for practical deployment. In [13] the authors implemented all physical layer processing steps at BS, near user and far user by suitably modifying an existing point-to-point wireless testbed. They have made use of Orthogonal Frequency Division Multiplexing (OFDM), and the design parameters are similar to those of the IEEE 802.11a standard. The testbed runs on GNU Radio (revision 10923) on a Linux PC. GNU Radio provides driver functions that interface the PC with the USRP board that functions as the analog frontend. They have been successful to experimentally determine the set of rate-pairs achieved by this transmission scheme under a packet-error constraint. The paper has a "rate-centric" view and focuses on improving spectral efficiency. Their results also suggest that superposition coding can provide substantial gains in spectral efficiencies over those achieved by orthogonal schemes such as Time Division Multiplexing.

In [14], [15] authors implement a few SISO-NOMA testbeds. They have considered both offline and real-time experiments. In the offline experiment, USRP is just
treated as the radio frequency (RF) terminal while the decoding and SIC are processes are done offline. In a real-time experiment, the time complexity of real-time SIC is evaluated.

In [16] authors proposed a NOMA-based wireless power transfer system using the SDR platform. The proposed system was implemented on a USRP platform and evaluated in terms of power transfer efficiency and user fairness. The results showed that the proposed scheme can improve the power transfer efficiency by up to 50 percent compared to conventional schemes while maintaining user fairness.

Furthermore, in [17] authors proposed a joint NOMA and beamforming scheme for the SDR platform. The proposed scheme was implemented on a USRP platform and evaluated in terms of system capacity and energy efficiency. The results showed that the proposed scheme can achieve up to 20 percent improvement in system capacity and up to 50 percent improvement in energy efficiency compared to conventional OMA and beamforming schemes.

1.2 Thesis Outline

This chapter has given a basic introduction to the need for NOMA, the basic principles of NOMA, and the objective of the work in brief.

The remaining contents are organized as follows:

Chapter 2: This chapter contains a review of past work done in the domain of NOMA with special focus on the hardware implementation of NOMA using SDRs.

Chapter 3: This chapter provides the fundamentals of two-user NOMA, details of the hardware and software used for the SDR platform and the LDPC encoding used in 5G standards.

Chapter 4: This chapter covers the system architecture, the block diagrams for the transmitter and receiver and the description of the individual modules.

Chapter 5: This chapter covers experimental results and explanation of these results.

Chapter 6: This chapter covers additional experiments done on 5G testbed at CEWiT, IIT Madras to test the compatibility of NOMA with industry standards of 5G. The experimental set-up, description of 5G NR frame structure and results are presented in this chapter.

Chapter 7: In this chapter, conclusions on the complete thesis work is provided and a discussion on the possibility of future work is presented.

CHAPTER 2

Background

2.1 Two user NOMA

In our experiment we have used the 2-user NOMA. We have considered downlink transmission using one base station and two users, namely user 1 and user 2. We consider user 1 as the near user from the base station and is situated at X distance. The user 2 is the far user from the base station and is situated at X+D distance. The distance between user 1 and user 2 is D. As illustrated in figure 1, both the users have been allocated the entire frequency band. However, the power allocated to each user is different. Let the channel between the base station and user 1 is h1, and the channel between the base station and user 2 is h_2. As user 2 is far from the base station compared to user 1, we assume its channel gain is smaller than user 1, that is mod of h_2 it is smaller than mod of h_1. The UE placed closer to the transmitter can be termed as a strong user with the assumption that it experiences a better channel with lesser distortion and lesser path loss and hence will be allocated a lower transmit power. The UE placed further away from the transmitter can be termed as the weak user which experiences more distortion and more path loss and is allocated more power.

To understand the transmission strategy, let's assume x_1 and x_2 are two distinct transmitted symbols to be transmitted to user 1 and user 2 respectively. Consider A_1 and A_2 are their respective power allocation factors such that A_1 plus A_2 is equal to one. In NOMA to promote user fairness, more power is given to the far user and less power to the near user. So A_2 should be greater than A_1. Thus, the superposition-coded NOMA signal transmitted by the base station can be expressed as follows,

$$y = A_1 x_1 + A_2 x_2 \quad \text{- (1)}$$

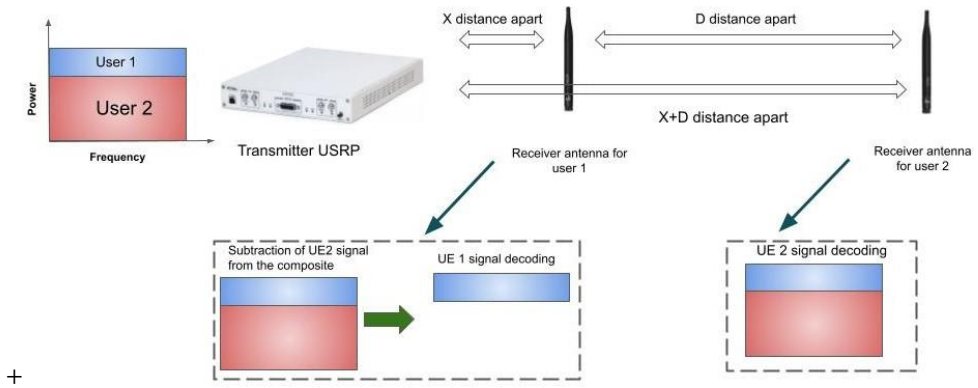


Figure 1: - System model for two-user NOMA

At user 2, which is the far user a direct decoding of y would yield x_2. The second term containing the x_1 component will be treated as an interference at user 2. At the near user, that is user 1 first the user 2 data is decoded and the subtracted from y. Then the remaining symbols are decoded to get the data for x_1.

2.2 Software Defined Radio

Software Defined Radio (SDR) is a radio communication system that uses reconfigurable software-based components for processing and conversion of digital signals. Unlike traditional radio communication systems, these radio devices are highly flexible and versatile. Utilising a software-based radio has the benefit of allowing us to readily make any modifications without having to alter the SDR platform, which is the hardware and operating system on which the waveform programme is running. For instance, modifying the physical layer functions just requires changing the software; redesigning the hardware is not necessary.

2.2.1 Hardware

The USRP 2952R is used as the peripheral equipment for this project. This USRP is built on the LabVIEW reconfigurable I/O (RIO) architecture, hence it delivers an integrated hardware and software solution. It can prototype a range of advanced research applications such as multiple input, multiple output (MIMO); synchronization of heterogeneous networks; LTE relaying; RF compressive sampling; spectrum sensing; cognitive radio; beamforming; and direction finding.

The USRP-2952 is equipped with a GPS-disciplined 10 MHz oven-controlled crystal oscillator (OCXO) reference clock. GPS disciplining delivers improved frequency accuracy and synchronization capabilities. It operates in the 400 MHz to 4.4 GHz frequency band. Using an MXI-Express x4 connection and PCIe-8371 interface card, the digital baseband signal is transmitted from the host computer to the USRP.

2.2.2 Software

All baseband processing of the signal is done using NI LabVIEW 2021. LabVIEW uses an object-oriented programming structure which uses the concept of encapsulation and inheritance to create a hierarchy of VIs and subVIs. Each VI or virtual instrument can be run on its own which helps to easily scale and modularize the programs. LabVIEW also has a vast function palette containing thousands of built-in functions and IP including analysis and I/O and a drag and drop feature which helps create to create applications with ease. It is also compatible to run real-time modules and embedded devices such as FPGAs, microprocessors, microcontrollers, PDAs, and touch panels.

2.2.3 Profiling

The aim is to profile and get the execution time and memory usage for each of the channel encoding techniques namely hamming code, convolutional codes, LDPC code to determine the more efficient encoder for 5G.

The Profile Performance and Memory window in LabVIEW is a powerful tool for determining the execution time of each of the VIs and SubVIs and the usage of memory.

This window has an interactive tabular display of time and memory usage for each VI in your system.

The Profile Performance and Memory window calculates the minimum, maximum, and average time spent per run of a VI. It also shows the minimum, maximum and average number of bytes used by the VI.

2.3 Quasi Cyclic LDPC Encoding

Quasi-Cyclic Low-Density Parity-Check (QC-LDPC) codes are a type of linear block error-correcting codes that have become popular in recent years, particularly in wireless communication and storage applications. QC-LDPC codes are a subclass of LDPC codes, which have low encoding and decoding complexity, and high coding gain.

The QC-LDPC codes are constructed by applying a permutation to a base matrix, which is a sparse matrix that has a quasi-cyclic structure. The base matrix is a circulant matrix whose rows are cyclic shifts of each other. The permutation applied to the base matrix rearranges its rows and columns, resulting in a new matrix that still has the quasi-cyclic structure. The QC-LDPC codes are characterized by their degree distribution, which determines the distribution of ones in the parity check matrix.

QC-LDPC codes have several advantages over traditional LDPC codes. One of the main advantages is their low encoding and decoding complexity, which makes them suitable for real-time applications. Moreover, QC-LDPC codes have a high coding gain, which means that they can correct a large number of errors with a relatively low redundancy. They are also resistant to burst errors, which makes them suitable for wireless communication systems that are prone to fading and interference.

In addition, QC-LDPC codes have been shown to outperform other error-correcting codes, such as turbo codes and convolutional codes, in terms of error-correction performance. They have been adopted in various wireless communication standards, such as WiMAX and DVB-S2, and are being considered for the upcoming 5G and 6G wireless communication standards.

2.3.1 LDPC Base Graph generation

For transmission of a DL transport block, a transport block CRC is first appended to provide error detection, followed by a LDPC base graph selection. NR supports two LDPC base graphs, one for small transport blocks and one for larger transport blocks.

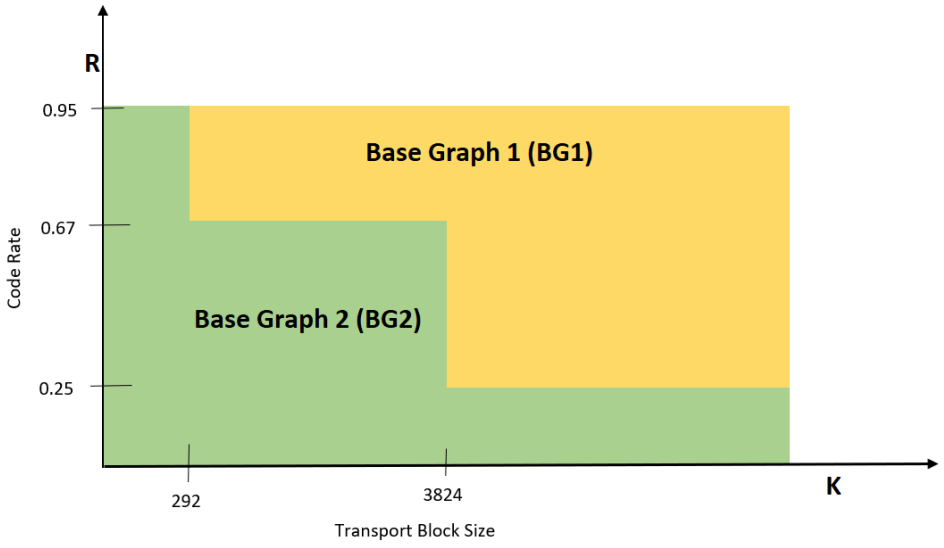


Figure 2: - LDPC base graph selection based on transport block size and code rate.

Then transport block is segmented into code blocks and code block CRC attachment is performed. Each code block is individually LDPC encoded. The LDPC coded blocks are then individually rate matched. Finally, code block concatenation is performed to create a codeword for transmission on the PDSCH. Up to 2 code words can be transmitted simultaneously on the PDSCH.

There are two types of Base Graphs standardized in the specification, 38.212 (Multiplexing and channel coding). Base Graph is a Matrix where each of the entries can be further expanded based on the expansion factor Zc.

Base Graph 1 (BG1): With Matrix size 46X68 entries --> For Large Transport Block

Base Graph 2(BG2): With matrix size 42X52 entries--> For Smaller Transport Block

Let us consider the following example to understand the construction of the LDPC parity check matrix for a given information block size K and code rate R = K/N. For simplicity let us have consider a small TBS of size 20 bits to illustrate below example, K=20 & R=0.25

Step 1: Obtain the base graph BG1 or BG2 for the given K (Transport Block)and R (Code Rate) according to the figure

As per the specification,

- if K<=3824 and R<=0.67 then BG2 is selected.
- If K<= 292 then BG2 is selected
- if R<=0.25 then BG2 is selected.
- Else BG1 is selected

Now since in our example K<292 , BG2 is selected.

Step 2: Determine the value of Kb for the given K (Transport Block) and R (Code Rate). Kb denotes the number of information bit columns for the lifting size Zc. As per Specification,

- For LDPC BG1, Kb = 22
- For LDPC BG2

a. if K > 640 then Kb = 10
b. if K is between 560 < K <= 640 then Kb = 9
c. if K is between 192 < K <= 560 then Kb =8
d. If K is <= 192 then Kb is = 6

- Since K<192 , Kb = 6

Step 3: Determine the base matrix expansion factor Zc by selecting the minimum Zc value in below Table, such that Kb × Zc ≥K

a

j \ Zc	2	3	5	7	9	11	13	15
0	2	3	5	7	9	11	13	15
1	4	6	10	14	18	22	26	30
2	8	12	20	28	36	44	52	60
3	16	24	40	56	72	88	104	120
4	32	48	80	112	144	176	208	240
5	64	96	160	224	288	352		
6	128	192	320					
7	256	384						

Table 1:- Selection of expansion factor based on the value of a and j

For K=20 , Zc = 4, this satisfies the condition Kb*Zc >= K , 6*4 = 24 , 24> 20 and this is the minimum Zc value from the above table that satisfies this condition.

Step 4: After Zc is determined, the corresponding shift coefficient matrix set need to be selected from below Table.

Set index (i_{LS})	Set of lifting sizes (Z)
0	{2, 4, 8, 16, 32, 64, 128, 256}
1	{3, 6, 12, 24, 48, 96, 192, 384}
2	{5, 10, 20, 40, 80, 160, 320}
3	{7, 14, 28, 56, 112, 224}
4	{9, 18, 36, 72, 144, 288}
5	{11, 22, 44, 88, 176, 352}
6	{13, 26, 52, 104, 208}
7	{15, 30, 60, 120, 240}

Table 2 : - Selection of lifting size based on set index

Since Zc = 4, Set Index (iLS) "0" is considered.

Step 5: Determine the entries values in the base matrix based on the Zc, Calculate the shifting coefficient value P(i,j) by the modular Z operation. P(i,j) = f(Vi,j , z) = mod(Vi,j , z)

$\mathbf{H}_{BG}$		$V_{i,j}$							
Row index i	Column index j	Set index i_{LS}							
		0	1	2	3	4	5	6	7
0	0	250	307	73	223	211	294	0	135
	1	69	19	15	16	198	118	0	227
	2	226	50	103	94	188	167	0	126
	3	159	369	49	91	186	330	0	134
	5	100	181	240	74	219	207	0	84
	6	10	216	39	10	4	165	0	83
	9	59	317	15	0	29	243	0	53
	10	229	288	162	205	144	250	0	225
	11	110	109	215	216	116	1	0	205
	12	191	17	164	21	216	339	0	128
	13	9	357	133	215	115	201	0	75
	15	195	215	298	14	233	53	0	135
	16	23	106	110	70	144	347	0	217
	18	190	242	113	141	95	304	0	220
	19	35	180	16	198	216	167	0	90
	20	239	330	189	104	73	47	0	105
	21	31	346	32	81	261	188	0	137
	22	1	1	1	1	1	1	0	1
	23	0	0	0	0	0	0	0	0

$\mathbf{H}_{BG}$		$V_{i,j}$							
Row index i	Column index j	Set index i_{LS}							
		0	1	2	3	4	5	6	7
15	1	96	2	290	120	0	348	6	138
	10	65	210	60	131	183	15	81	220
	13	63	318	130	209	108	81	182	173
	18	75	55	184	209	68	176	53	142
	25	179	269	51	81	64	113	46	49
	37	0	0	0	0	0	0	0	0
16	1	64	13	69	154	270	190	88	78
	3	49	338	140	164	13	293	198	152
	11	49	57	45	43	99	332	160	84
	20	51	289	115	189	54	331	122	5
	22	154	57	300	101	0	114	182	205
	38	0	0	0	0	0	0	0	0
17	0	7	260	257	56	153	110	91	183
	14	164	303	147	110	137	228	184	112
	16	59	81	128	200	0	247	30	106
	17	1	358	51	63	0	116	3	219
	21	144	375	228	4	162	190	155	129
	39	0	0	0	0	0	0	0	0
	1	42	130	260	199	161	47	1	183

Table 3: - Calculation of set index value based on row index and column index

For K=20 , Base Graph = 2, Zc= 4 & SetIndex " iLS" = 0, from above Table 5.3.2-3: LDPC base graph 2 Using the equation "P(i,j) = f(Vi,j , z) = mod(Vi,j , z)" all the possible base graph matrix entries with the shifting coefficient are determined,

From Step 3 & 4 Set Index iLS = 0 & Zc =4

Using the above principal remaining all entries of the matrix can be populated based on the Row index “i" and Column Index “j” & iLS SetIndex.

Step 6: Each entry in the final exponent matrix should be replaced with the equivalent zero matrix or circulant permutation matrix of size Z x Z. After completing the LDPC code construction, a parity check matrix H of dimension mbZ x nbZ is obtained.

As per LDCP construction model each entry in the base graph is further expanded based on the expansion factor "Zc" with a shifting coefficient. For example, all entries which are 2 in the base graph are replaced with a "ZxZ matrix with 1's of identity matrix shifted to right by two position's" and all entries which are 3 in the base graph are replaced with a "4x4 matrix with 1's shifted to right by three position"

In our project we have considered a base graph with only 3 entries which are 1,-1 and 0. First we read a base graph file of size 46x68 containing elements 1,0 and -1 Here in our project we have used base graph 1 with an expansion factor of 2. Each element of the base graph is considered one by one. All -1 values in the Base Matrix are replaced with a "2x2 All Zeros matrix". All 0 values in the Base Matrix are replaced with a "2x2 Identity matrix with 1's located diagonal". All 1 values in the Base Matrix are replaced with a "2x2 matrix with 1's shifted to right by one position". Hence now the expanded base graph is a 92x136 matrix.

K denotes the data bits per packet.
For Base graph 1, K is given by (22 x Expansion Factor)
Hence here K= 2 x 22 = 44 bits
Length of the input bit stream is given as B. In our project the length of input is 3200.

If B <= K then no segmentation is required and the input bit stream forms the codeword
If B > K the the whole input bit stream is segmented into smaller blocks of K size.
The total number of codewords are C=B/K = 3200/44 = 72.73 = 73

If B is not exactly divisible by K the remaining bits in the last codeword will be filled by filler bits which are zeroes. Now the codewords blocks are multiplied by the parity check matrix of LDPC generated in step 3. This gives us the LDPC codes.

2.3.2 LDPC Decoding

A hard-decision decoder operates on data that takes a fixed set of possible values (typically 0 or 1 in a binary code). Suppose the threshold voltage chosen by the hard decision decoder is 2.2V. Any voltage received above 2.2V is considered as 1 bit and any voltage received below 2.2V is considered 0 bit.

A soft-decision decoder takes on a whole range of in-between values as the input. Soft-decision decoding technique provides better error correction capability than hard decision decoding. To enhance the decoder gain, modern LDPC decoders work with soft decision algorithms. To decode faster the whole Base Matrix is not required to be transmitted, only the required part of the matrix can be transmitted this can be achieved by puncturing and rate matching mechanisms specified in specification 38.212.

Soft decoding is a technique that utilizes soft information, which includes the likelihood or reliability of each bit's value, rather than relying solely on hard decisions (0 or 1). By incorporating soft information into the decoding process, LDPC codes can effectively handle and correct errors, especially in scenarios with high noise levels. Soft decoding algorithms, such as the belief propagation algorithm, efficiently utilize the soft information to iteratively update the likelihood values of the transmitted bits, leading to improved error correction performance in LDPC-based 5G systems. The utilization of soft decoding techniques in LDPC codes is a significant factor in achieving high reliability and efficient data transmission in 5G networks.

There are different types of soft decoding algorithms used for decoding LDPC (Low-Density Parity-Check) codes. Here are a few commonly employed soft decoding techniques:

1. Sum-Product Algorithm (SPA): The Sum-Product Algorithm, also known as the belief propagation algorithm, is a widely used soft decoding method for LDPC codes. It utilizes the concept of message passing between variable and check nodes to iteratively update the likelihood information. SPA calculates the messages by computing the sum of logarithmic likelihood ratios (LLRs) received from neighboring nodes.
2. Log-Likelihood Ratio (LLR) Decoding: LLR decoding involves representing the likelihood of each bit being 0 or 1 as a real-valued LLR. Soft decoding algorithms leverage LLRs to refine the estimates of the transmitted codeword. LLR decoding can be performed using various techniques, including the Sum-Product Algorithm and its variants.
3. Min-Sum Algorithm: The Min-Sum Algorithm is a simplified version of the Sum-Product Algorithm, aiming to reduce the computational complexity while maintaining decent decoding performance. Instead of summing the LLRs, the Min-Sum Algorithm calculates the minimum absolute value of the accumulated LLRs, which reduces the computational complexity at the expense of some performance degradation.
4. Offset Min-Sum Algorithm: The Offset Min-Sum Algorithm is an enhancement to the Min-Sum Algorithm. It introduces an offset to the accumulated LLRs, which helps in improving the error floor performance of the decoder.
5. Approximate Message Passing (AMP): AMP is an iterative soft decoding algorithm that combines message passing with estimation techniques. It approximates the posterior distribution of the transmitted codeword by iteratively updating the estimates using a denoising function.

These are just a few examples of soft decoding algorithms used for LDPC codes. The choice of algorithm depends on factors such as the desired trade-off between complexity and performance, specific requirements of the application, and available computational resources. Researchers continue to explore and develop new soft decoding techniques to enhance the efficiency and performance of LDPC decoders.

Log-Likelihood Ratio (LLR) decoding is a popular technique used in the decoding of Low-Density Parity-Check (LDPC) codes. LDPC codes are linear error-correcting codes with sparse parity-check matrices, which make them suitable for efficient implementation and high-performance error correction.

LLR decoding operates on the soft decision values of received symbols. Instead of making hard decisions (0 or 1) on the received bits, LLR decoding assigns a likelihood ratio value to each bit, representing the likelihood of it being a 0 or 1. The LLR value is defined as the logarithm of the ratio of the probability of the bit being 0 to the probability of it being 1.

In the context of LDPC decoding, LLR decoding is typically performed using iterative algorithms such as belief propagation, also known as the sum-product algorithm. The algorithm takes as input the received LLR values and computes updated LLR values for each bit in an iterative manner.

The LLR values are passed between variable nodes and check nodes in the LDPC graph, where variable nodes represent the bits and check nodes represent the parity-check equations. At each iteration, the algorithm computes new LLR values at the variable nodes based on the information received from the connected check nodes, and vice versa.

The computation of updated LLR values involves message passing and soft decision combining. Messages are exchanged between variable and check nodes, and LLR values are combined using the log-likelihood ratio addition formula. The process continues until a stopping criterion is met, such as reaching a maximum number of iterations or achieving a desired level of error correction.

Once the decoding process is completed, the LLR values can be used to make hard decisions on the decoded bits. Typically, a threshold is applied to the LLR values to determine the most likely bit value.

LLR decoding in LDPC provides excellent error correction performance, especially when combined with powerful iterative decoding algorithms. It is widely used in various communication systems, including wireless, satellite, and optical communication, where LDPC codes are employed to achieve reliable data transmission.

The hard-decision demodulation computes the minimum Hamming distance for each received sample and selects the symbol with minimum distance. When the hard decision output for a symbol has an equal Hamming distance for multiple codewords, one of those codewords is randomly selected. Using soft-decision demodulation can reduce the probability of decision error, but it is more computationally intensive.

Two soft-decision algorithms are available: exact log-likelihood ratio (LLR) and approximate LLR. Exact LLR provides the greatest accuracy but is slower, while approximate LLR is less accurate but more efficient. The exact LLR algorithm computes exponentials using finite precision arithmetic. For computations involving very large positive or negative magnitudes, the exact LLR algorithm yields:

- Inf or -Inf if the noise variance is a very large value
- NaN if the noise variance and signal power are both very small values

The approximate LLR algorithm does not compute exponentials. You can avoid Inf, -Inf, and NaN results by using the approximate LLR algorithm.

The log-likelihood ratio (LLR) is the logarithm of the ratio of probabilities of a 0 bit being transmitted versus a 1 bit being transmitted for a received signal. The LLR for a bit, b, using exact LLR algorithm is defined as:

$$L(b) = \log\left(\frac{P_r(b=0|r=(x,y))}{P_r(b=1|r=(x_1y))}\right) \quad \text{- (2)}$$

Assuming equal probability for all symbols, the LLR for an AWGN channel can be expressed as

$$L(b) = \log\left(\frac{\sum_{s\in S_0} e^{-\frac{1}{\sigma^2}\left((x-s_x)^2+(y-s_y)^2\right)}}{\sum_{s\in S_1} e^{-\frac{1}{\sigma^2}\left((x-s_x)^2+(y-s_y)^2\right)}}\right) \quad \text{- (3)}$$

CHAPTER 3

System Model

3.1 Transmitter

The various blocks comprising the transmitter system model are illustrated in figure 2. The description of the blocks are as follows,

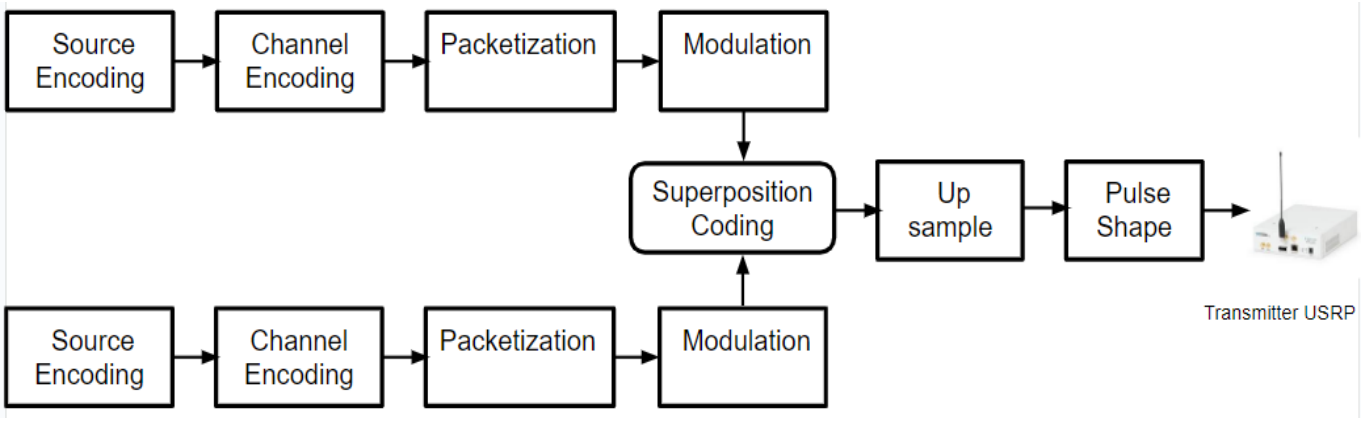


Figure 3: Transmitter system model

1)Source Encoding

Here we are considering a model consisting of 2 UEs sending text data. Each users' text data is in the form of a string which needs to be concatenated together. The text in this appended array is then converted to bits.

As bandwidth of any channel is limited it is necessary to do redundancy reduction to remove unwanted bits. This is accomplished by a source encoder, which operates in conjunction with an analog-to-digital converter.

2)Channel Encoding

The reduced bits are then transferred to the channel encoding module. Unlike the source encoder which reduces redundancy the channel encoder, on the other hand adds redundancy to the transmitted signal so that errors caused by noise during transmission can be corrected at the receiver. This process of encoding for protection against channel errors is called error-control coding. In this project we have used three types of channel encoders namely hamming, LDPC and convolutional.

3)Packetization

Many bits are accumulated after the channel encoding process. To transmit this large data accurately and in a way that the channel doesn't get over-flooded, the total bits are divided into small packets before modulation. This process is called packetizing.

4)Modulation

The modulation schemes used on the data bits are BPSK, QPSK and 8-PSK.

5)Upsample

Upsampling of the baseband signal is important to increase resolution as well as to prevent aliasing of the signal when it is passed through the digital to analog(DAC) converter present in the USRP. Upsampling also improves the anti-aliasing filter performance and reduces the noise. It is also a prerequisite for pulse shaping.

For practical digital communication to work we need some excess bandwidth above minimum. This is because we need timing and frequency synchronization to be done which need excess bandwidth.

6)Pulse Shape

Pulse shaping is the process of changing the waveform of transmitted pulses. Its purpose is to make the transmitted signal better suited to its communication channel, typically by limiting the effective bandwidth of the transmission. Transmitting a signal at high modulation rate through a band-limited channel can create inter-symbol interference. A band-limited signal corresponds to an infinite time signal which leads to overlap of neighbouring pulses.

An increase in modulation rate leads to increase in the signal bandwidth. As soon as the spectrum of the signal is a sharp rectangular, it leads to a sinc shape in the time domain. This happens if the bandwidth of the signal is larger than the channel bandwidth which leads to a distortion known as intersymbol interference (ISI). To avoid this from occuring we go for pulse shaping of the signal before transmitting it through the channel.
For upsampling we first generate root raised cosine (RRC) filter coefficients which are then convolved with the upsampled symbol stream.

Once pulse shaping is completed, all the data is stored in a queue. The transmission through the VERT2450 antenna is initiated once the number of elements in the queue exceeds the set buffer threshold.

3.2 Receiver

The various blocks comprising the receiver system model are illustrated in figure 3.
The description of the blocks are as follows,

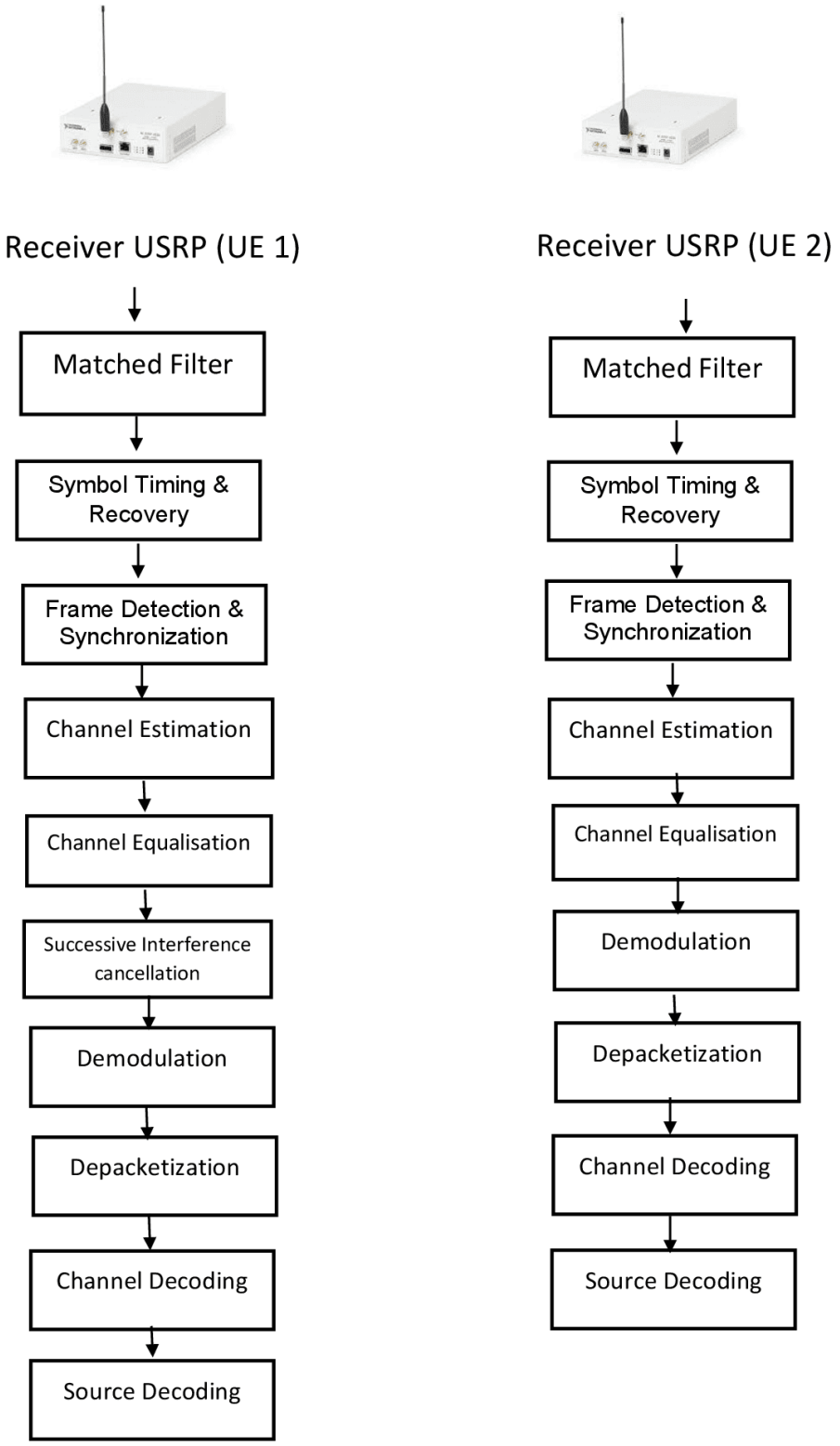


Figure 4: Receiver system model

1) Matched filter

It is the optimal linear filter used to increase the signal-to-noise ratio (SNR) of the signal received at the antenna of the receiver USRP. For this we generate root raised cosine (RRC) filter coefficients which are then convolved with the received symbols to get the filtered signal.

2)Symbol Timing and recovery

The receiver will then decide which symbols were sent by sampling the message signals at the symbol rate. Although the receiver is aware of the symbol rate, it is unaware of the ideal time to sample the signal in order to minimise noise. The symbol-timing recovery loop seeks to determine the most advantageous moment to sample the received signal.

3)Frame detection & synchronisation

The output of matched filter is downsampled to and then cross-correlated with a pre-determined sync sequence. We square the absolute value of the bit stream obtained by cross-correlation and find out the position where the maximum sync occurs.

4)Channel Estimation

The channel estimation algorithm extracts the reference signals for a transmit/receive antenna pair from the received grid.

In our system model we have used the linear least squares algorithm to estimate the channel.

Estimation is done considering a linear system,

$$Ay = B \quad \text{- (4)}$$

where A is the known matrix, y is an unknown vector or channel coefficients, and B is the observation vector or the preloaded training sequence. Assuming A is the full-rank matrix, the least-square error solution is,

$$\hat{y} = (A^H A)^{-1} A^H B \quad \text{- (5)}$$

where A^H is the Hermitian or conjugate transpose of A . The squared error (or the square of the L2 norm of the error) is
defined as,

$$SE = \|Ay - B\|_2^2 \quad \text{- (6)}$$

5)Channel Equalisation

In telecommunication, equalization is the reversal of distortion incurred by a signal transmitted through a channel. The channel estimate coefficients obtained earlier are converted into a toeplitz matrix D and then converted into a penrose-equaliser matrix. This is done by taking a pseudoinverse of matrix D and then multiplied with equalizer delay constants.

The equalizer coefficients are convolved with the received symbols to get the equalized symbols.

6)De-modulation and De-packetization

Finally, we do demodulation of the signal and re-group the different packets received. The packets are concatenated together to get a single stream of bits.

7)Channel Decoding

A channel decoder is used on the receiver side to return the binary information back to its original form by removing the parity bits.

8)Source Decoding

The source decoder adds back the redundant bits which were removed during source encoding to save bandwidth. This is done to ensure that the original information is restored.

3.3 System Configuration and Parameters

Figure 5 : Experimental set-up

The experiments are carried out at our research lab using the set up shown in figure . Two GPP are connected to two NI-USRP 2952R via an MXI-Express x4 cable and a PCIe-8371 interface card. Each GPP has an Intel(R) Core(TM)i7-4790 processor and operates at a maximum clock frequency of 3.60 GHz.

Due to its maximum sample rates of 400 MS/s at the transmitter and 120 MS/s at the receiver, which can be adjusted as required, the USRP 2952R is the perfect choice for a peripheral equipment. Additionally, a broad frequency range of 400 MHz to 4.4 GHz is supported by the USRP 2952R, with a maximum instantaneous real-time bandwidth of 120 MHz. One USRP is used as the transmitter, while the other is used as the receiver. The same 1.4 GHz carrier frequency for transmission and reception is used, together with an IQ rate of 200 kilosamples per second.

All baseband signal processing is implemented on NI LabVIEW 21.0 Development System software. The transmit and receive antenna is 30cm apart in case of near user and 100 cm apart in case of far user. The antennas used for experiment are VERT2450 and all antennas have line of sight transmission. VERT2450 is a dual band (2.4 to 2.48 GHz and 4.9 to 5.9 GHz) omni-directional vertical antenna with a 3dBi gain. The modulation schemes used for the data are BPSK, QPSK and 8-PSK. The other system parameters are summarized in figure 5.

Parameters	Value
USRP	2952R
CPU	Intel(R) Core(TM)i7- 4790 CPU @ 3.60GHz
IQ rate	200 kS/sec
Channel Bandwidth	120 MHz
Carrier Frequency	1.4GHz
Data bits per packet	1000
Modulation scheme	BPSK, QPSK, 8-PSK
Channel Estimation	Linear Least Square Estimation
Channel coding/decoding	Hamming, Convolutional, LDPC
Transmitter Gain	10

Table 4 : System Configurations

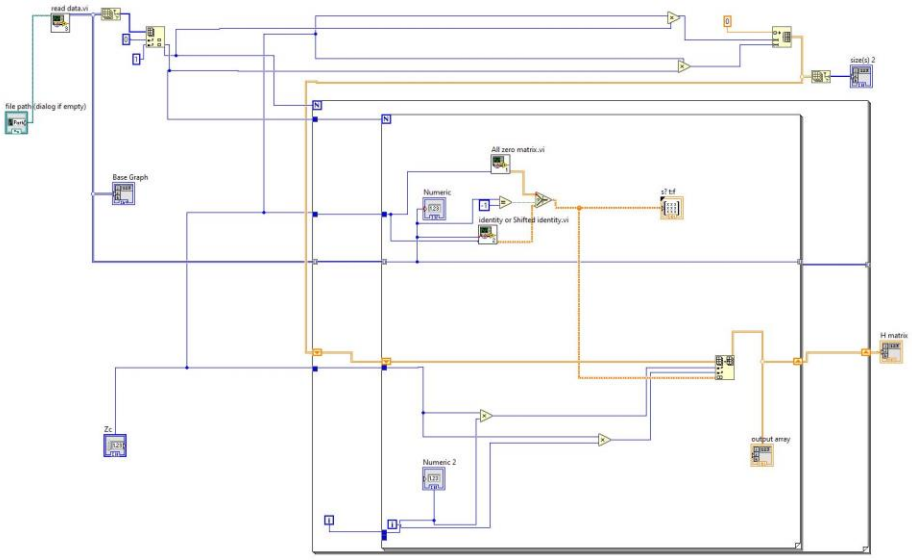


Figure 6 : SubVI to generate parity check matrix from base graph

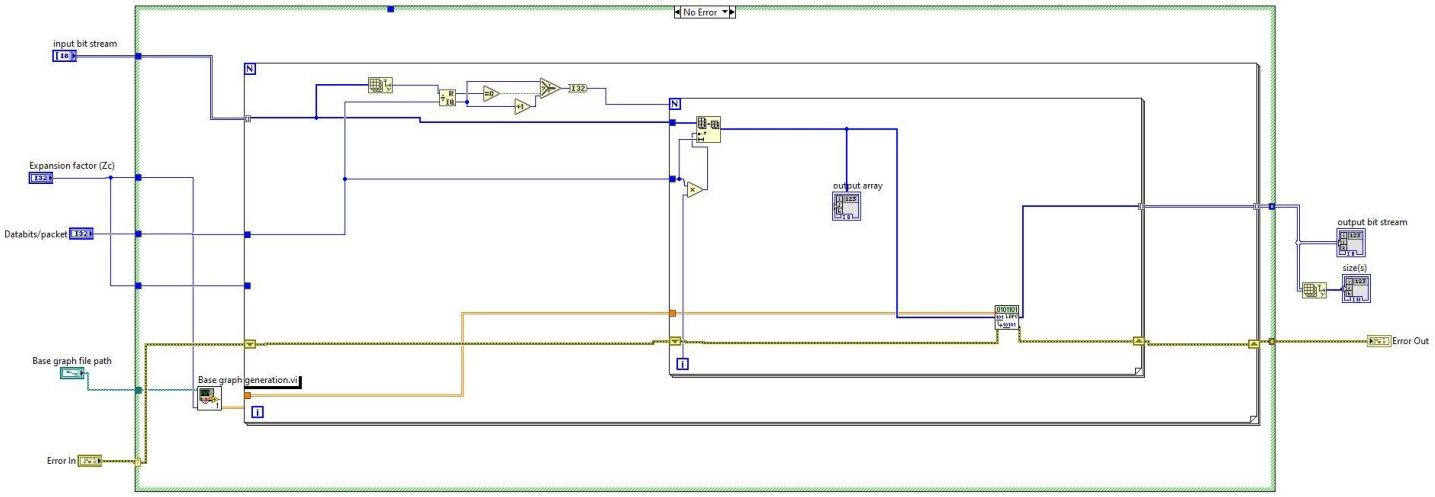


Figure 7 : SubVI to generate LDPC codes by using input bit stream and parity check matrix

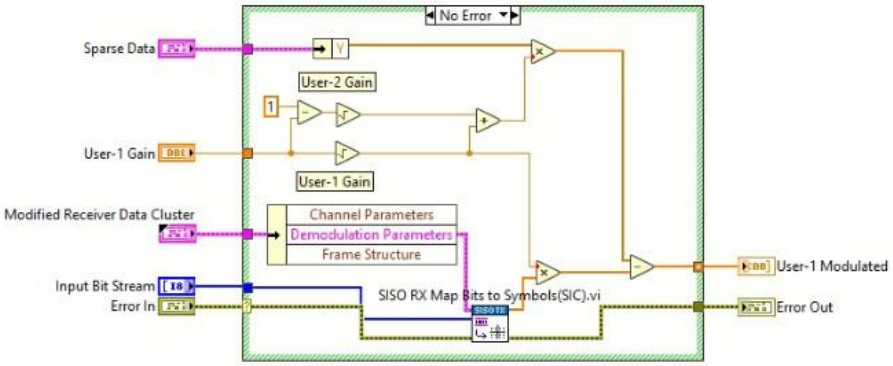


Figure 8 : SubVI to implement SIC for near user

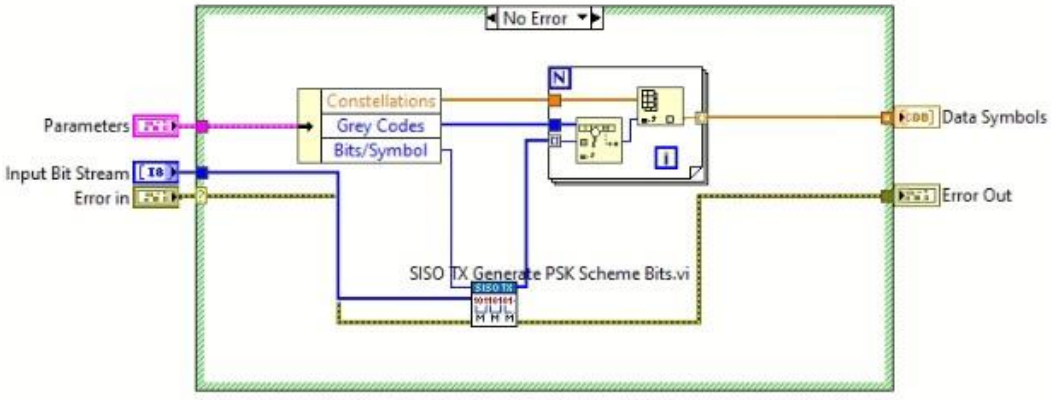


Figure 9 : SubVI to regenerate far user signal for SIC

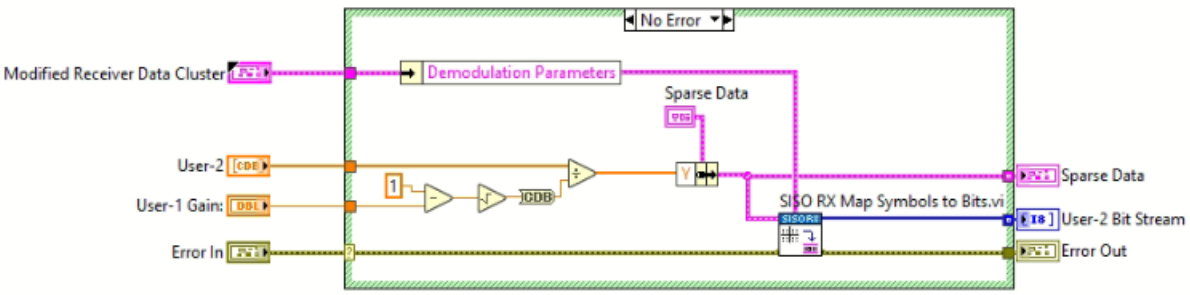


Figure 10 : SubVI to decode near user

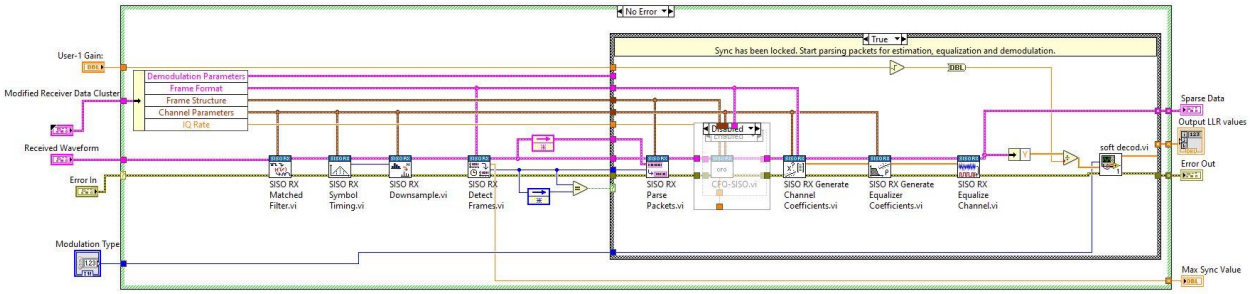


Figure 11 : SubVI for soft decoding of LDPC codes and obtaining LLR values

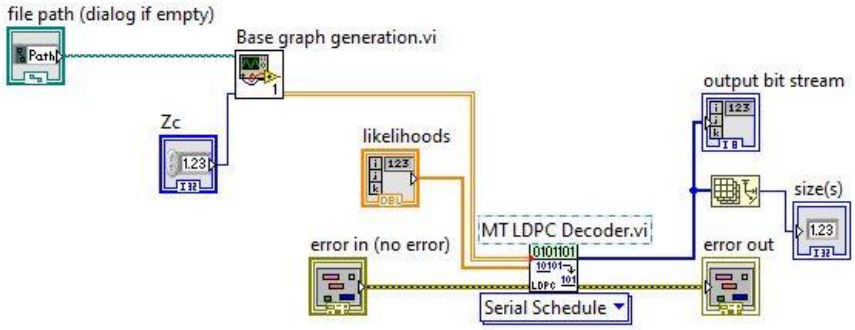


Figure 12 : SubVI to obtain original input data

CHAPTER 4

Results and Discussions

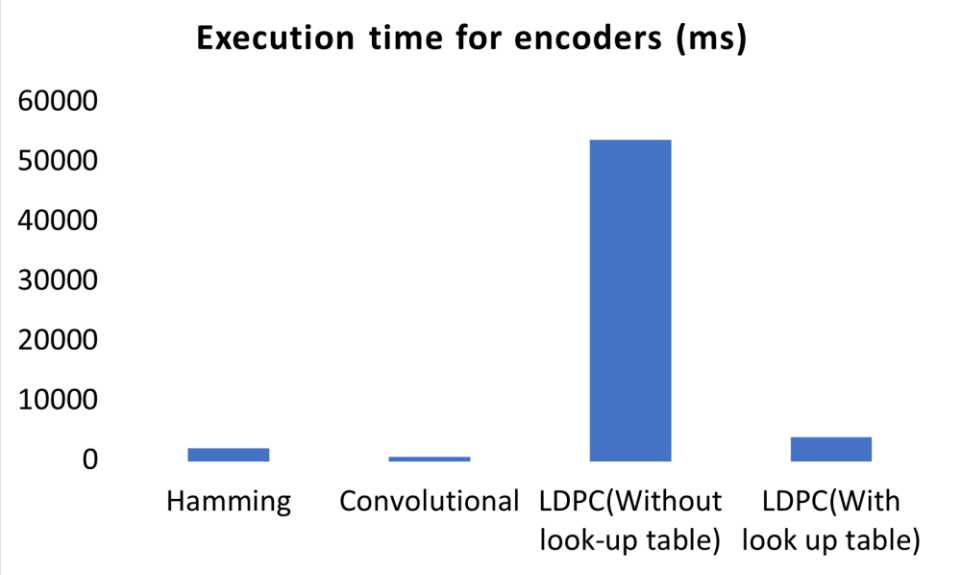
4.1 Execution time and memory usage for channel encoders of NOMA

The profiling results for the three types of channel encoders are depicted in table 8 and the comparative bar graphs are shown in figure 7 and figure 8. These results are for when the number of cycles is equal to 10. We observe that the convolutional encoder takes the least amount of execution time (8.6 ms) while LDPC takes the maximum amount of execution time (53091.8 ms). For average memory usage we observe that hamming encoder consumes the least memory (2.25 kilobytes) while LDPC encoder takes the most memory (501.99 kilobytes).

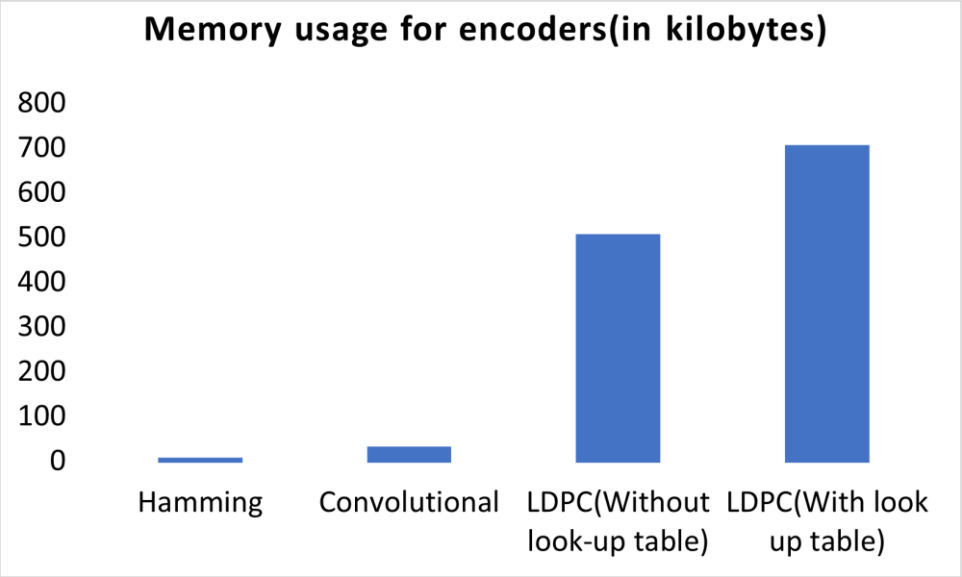
However, we can improve upon the execution time of LDPC encoder by using an alternative approach. It has been observed that a major portion of the total execution time is used up in calculating the parity check matrix from the base graph depending on the chosen expansion factor. Hence this process can be done offline before the beginning of the transmission. The corresponding parity check matrix for each expansion factor can be stored in a look-up beforehand and only the required parity check matrix be loaded into the transmitter at the time of transmission. This reduces the total execution time of LDPC encoder from 53091.8 ms to 3356 ms which is a 93.78 percent improvement in the execution time. Here we also observe that the memory requirement goes up by around 200 kilobytes. Hence there is a trade-off between the execution time and memory usage.

Type of Channel Encoder	VI Time (ms)	SubVIs Time (ms)	Total Execution Time (ms)	Min Bytes Used (kilobyte)	Max Bytes Used (kilobyte)	Average Bytes Used (kilobyte)
Hamming	0.1	1552.6	1552.7	2.25	2.25	2.25
Convolutional	5.5	3.1	8.6	26.19	26.19	26.19
LDPC(Without look-up table)	54.6	53039.1	53093.7	501.98	502.00	501.99
LDPC(With look up table)	54.6	3296.9	3351.5	700.62	700.64	700.63

Table 5: The profiling results of execution time and memory usage for the channel encoders when number of cycles is equal to 10



(a)



(b)

Figure 13 : (a) The comparison of execution time of the three types of channel encoders when number of cycles is equal to 10 (b) The comparison of memory usage of the three types of channel encoders when number of cycles is equal to 10

4.2 Execution time and memory usage for channel decoders of NOMA

The profiling results for the three types of channel decoders in near user are depicted in figure 9 and the comparative bar graph is shown in figure 11. These results are for when the number of cycles is equal to 10. We observe that the convolutional decoder takes the least amount of execution time (43.1 ms) while LDPC takes the maximum amount of execution time(1051.9 ms). For average memory usage, we observe that the convolutional encoder consumes the least memory (4.22 kilobytes) while the LDPC encoder takes the most memory (101.29 kilobytes).

The profiling results for the three types of channel decoders in far user are depicted in figure 10 and the comparative bar graph is shown in figure 12. These results are for when the number of cycles is equal to 10. We observe that the convolutional decoder takes the least amount of execution time (43.3 ms) while LDPC takes the maximum amount of execution time (1055.7 ms).

The comparative bar graph for average memory usage is depicted in figure 13. We observe that the memory usage in both the users is the same. The convolutional encoder consumes the least memory (4.22 kilobytes) while the LDPC encoder takes the most memory (101.29 kilobytes).

In all three encoding techniques we observe that the decoding takes significantly lesser time than the corresponding encoding process.

Type of Channel Decoder	VI Time (ms)	SubVIs Time (ms)	Total Execution Time (ms)	Min Bytes Used (kilobyte)	Max Bytes Used (kilobyte)	Average Bytes Used (kilobyte)
Hamming	0.3	354.4	354.7	4.45	4.45	4.45
Convolutional	0.2	42.9	43.1	4.22	4.22	4.22
LDPC	12.8	1039.1	1051.9	101.28	101.30	101.29

Table 6: The profiling results of execution time and memory usage for the channel decoders in near users when number of cycles is equal to 10

Type of Channel Decoder	VI Time (ms)	SubVIs Time (ms)	Total Execution Time (ms)	Min Bytes Used (kilobyte)	Max Bytes Used (kilobyte)	Average Bytes Used (kilobyte)
Hamming	0.3	360.5	360.8	4.45	4.45	4.45
Convolutional	0.2	43.1	43.3	4.22	4.22	4.22
LDPC	12.8	1042.9	1055.7	101.28	101.30	101.29

Table 7: The profiling results of execution time and memory usage for the channel decoders in far users when number of cycles is equal to 10

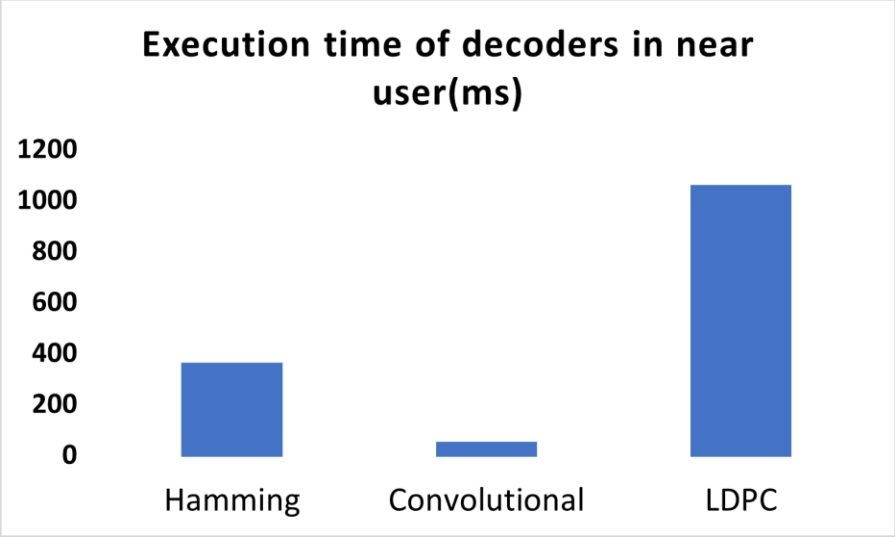


Figure 14 : The comparison of execution time of the three types of channel decoders in near user when number of cycles is equal to 10

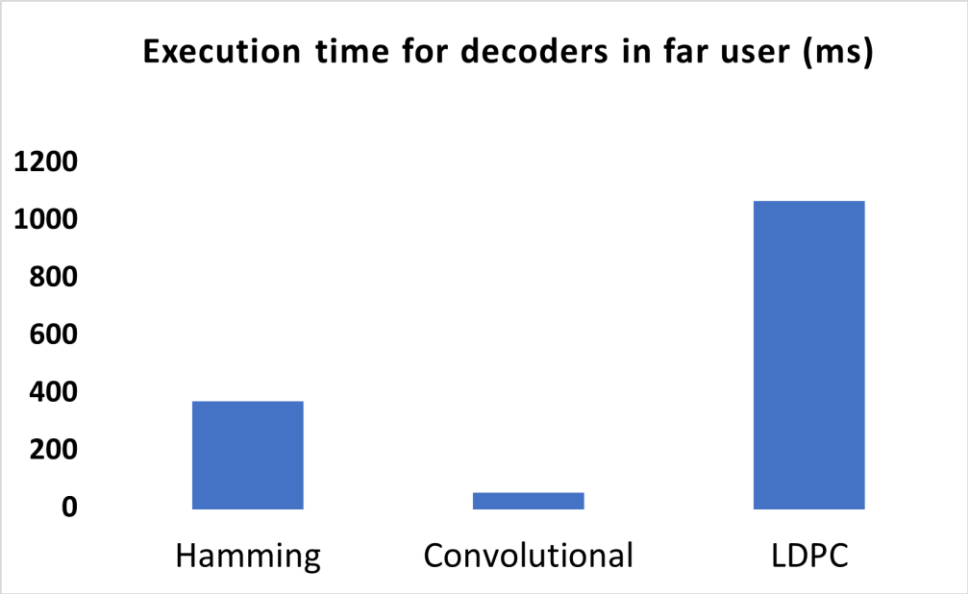


Figure 15 : The comparison of execution time of the three types of channel decoders in far user when number of cycles is equal to 10

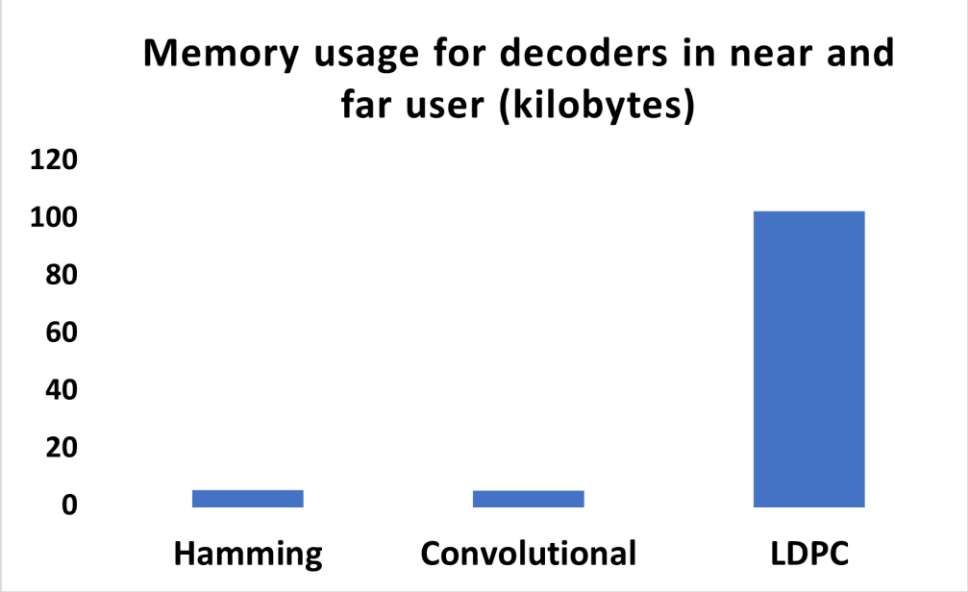


Figure 16: The comparison of memory usage of the three types of channel decoders in near user and far user when number of cycles is equal to 10

4.3 Bit error rate for different channel encoding techniques

Figure 18 and figure 19 demonstrates the BER versus SNR graphs for near and far user respectively. The figure shows the graph obtained by theoretical analysis, simulation as well as by over-the-air implementation of NOMA using convolutional, LDPC and hamming channel encoding techniques. The theoretical analysis has been done using the equation for BER calculation for the m^{th} user in an M-user NOMA system considering AWGN as given below,

$$P_{b,m} = \frac{1}{2^{M-m}} \sum_{j=1}^{2^{M-m}} Q\left(\beta_{m,j}^{+}\sqrt{\gamma}\right) \quad \text{- (7)}$$

In equation(7), $\beta_{m,j}^{+} = \lambda_{m,j}^{+}/\sqrt{P}$ and $\gamma = P/\sigma^2$. Here P denotes the signal power, σ^2 denotes the noise power and γ denotes the SNR. The calculation for $\beta_{m,j}^{+}$ requires understanding of the term $\lambda_{m,j}^{+}$. For calculating λ_m we use equation (8) where α_m is the power coefficient of m^{th} user, Si in λ_m are assigned as +1 or −1 when considering BPSK transmission. Accordingly, we get λ_m^{+} and λ_m^{-} by substituting value of Si as shown in equation (9) and (10). Once we have values for λ_m^{+} and λ_m^{-} we can obtain $\lambda_{m,j}^{+}$ and $\lambda_{m,j}^{-}$ which denote the j^{th} combination of λ_m^{+} and λ_m^{-}, respectively, where $j = 1, \ldots, 2^{M-m}$

$$\lambda_m = \alpha_m S_m + \sum_{i=m+1}^{M} \alpha_i S_i \quad \text{- (8)}$$

$$\lambda_m^{+} = \alpha_m + \sum_{i=m+1}^{M} \alpha_i S_i \quad \text{- (9)} \quad , \quad \lambda_m^{-} = -\alpha_m + \sum_{i=m+1}^{M} \alpha_i S_i \quad \text{- (10)}$$

The figures shows that for low SNR the performance of LDPC with soft decoding is comparable to hamming code and better than convolutional code. At 5dB and 6dB the performance of hamming and convolutional is comparable. However, after 6dB the performance of convolutional is better than that of hamming.

We see that in the near user, for LDPC the maximum deviation of experimentally obtained BER value from the theoretical value occurs at 7dB. The difference between the two is 3x10−5. For the same user the maximum deviation while using hamming coding occurs at 10dB (0.4x10−6) and while using convolutional coding occurs at 11dB (2x10−9). For far user, the maximum deviation occurs at 8dB for LDPC (1.2x10−6), at 9 dB for hamming (1.3x10−5) and at 11dB for convolutional coding (2x10−9). For higher SNR we see that LDPC performs better than both the convolutional and hamming codes. This is possible because we are making use of a soft decoding technique for LDPC which performs better than the conventional hard decoding LDPC.

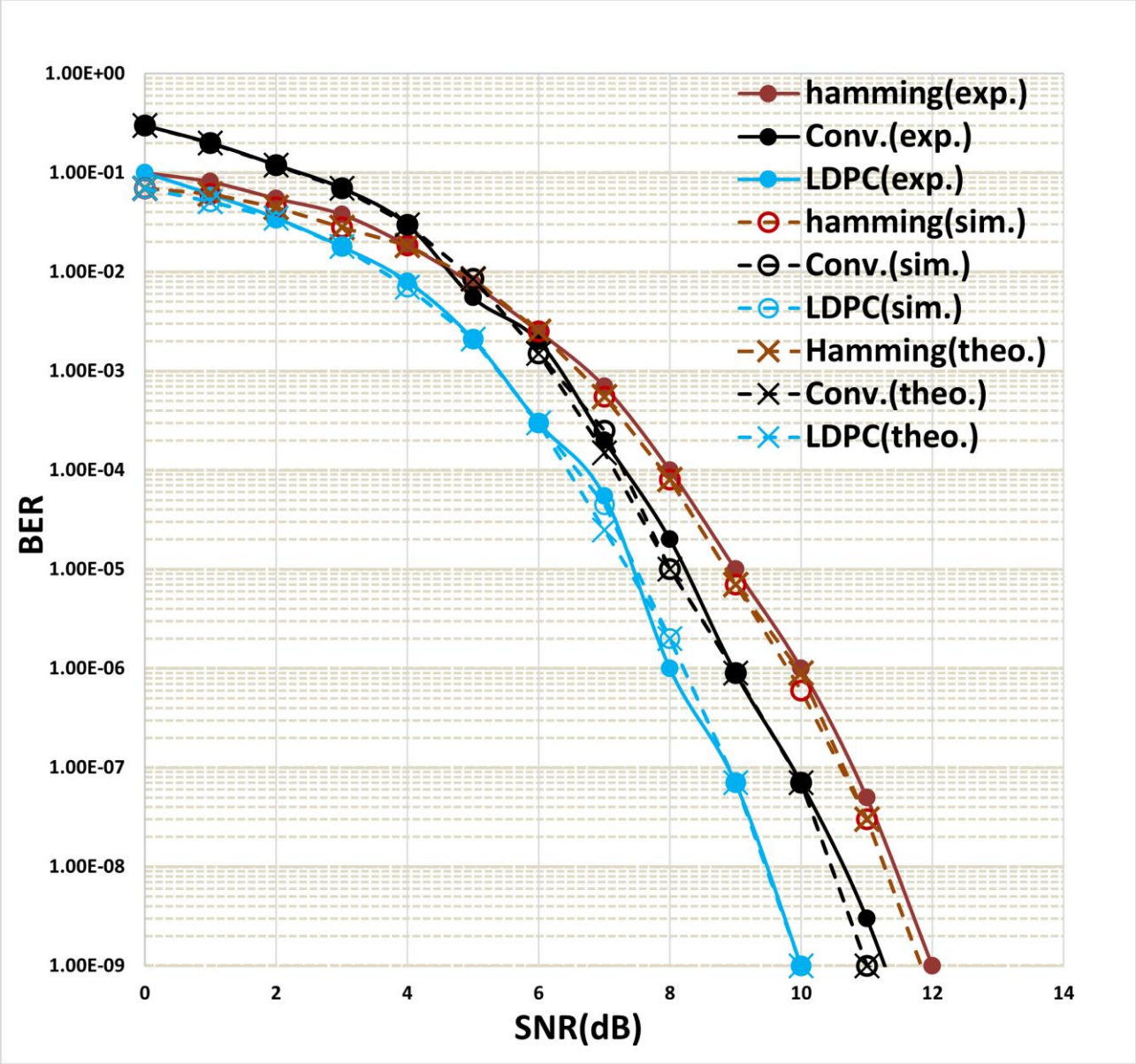


Figure 17: SNR versus BER plot for the near user using the three channel encoding techniques. Graph shows a comparison between the theoretical, simulated, and experimental results of each of the encoding technique.

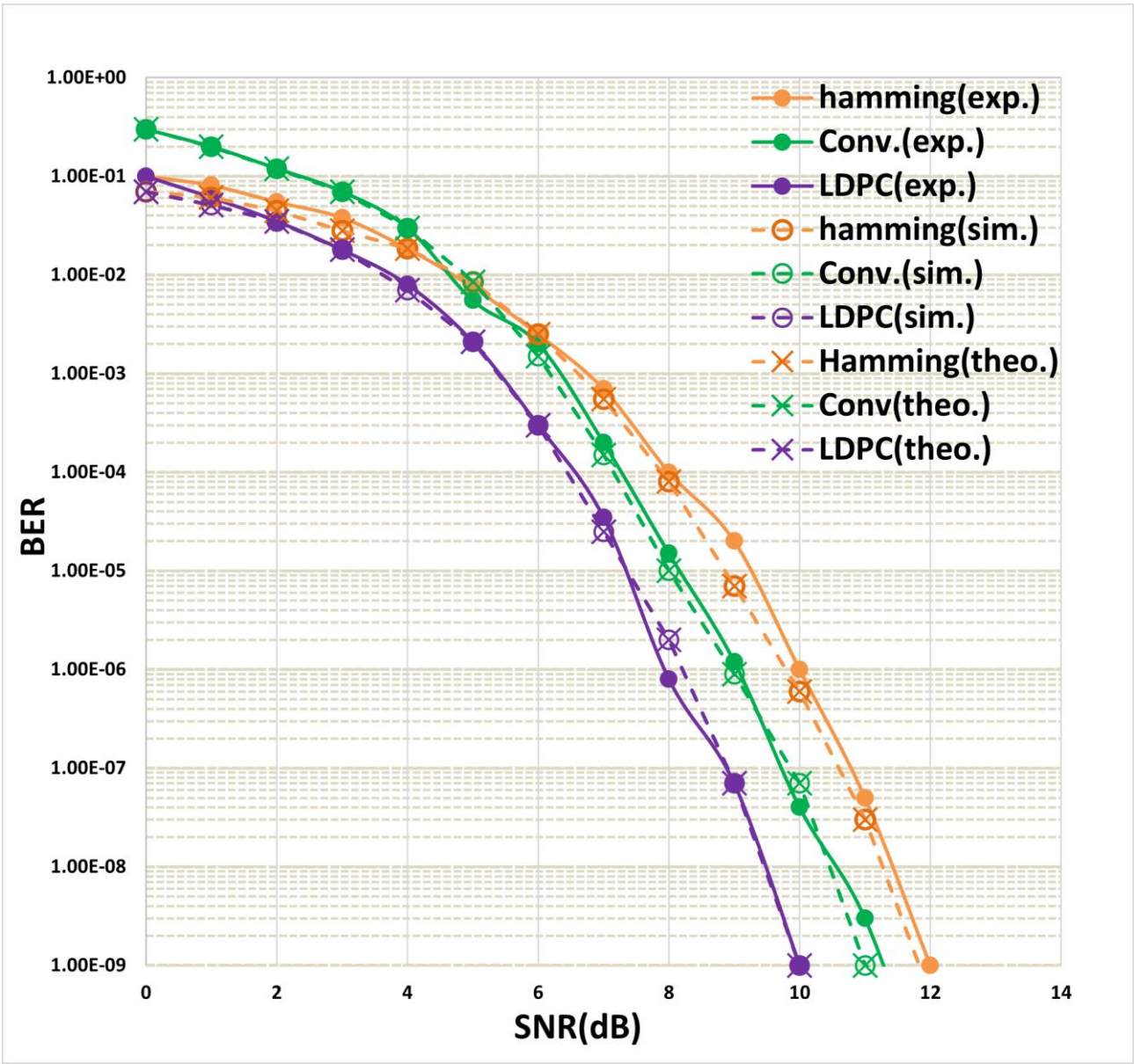


Figure 18: SNR versus BER plot for the far user using the three channel encoding techniques. Graph shows a comparison between the theoretical, simulated, and experimental results of each of the encoding technique.

CHAPTER 5

NOMA implementation at CEWiT

We have already established the advantages of NOMA and its requirement for future generations of cellular networks. To test the compatibility of NOMA technique with the standard 5G specifications set by the industry we undertake implementation of NOMA on the 5G testbed at the Centre of Excellence in Wireless Technology (CEWiT) at IIT Madras Research Park.

5.1 Experimental Set-up at CEWiT

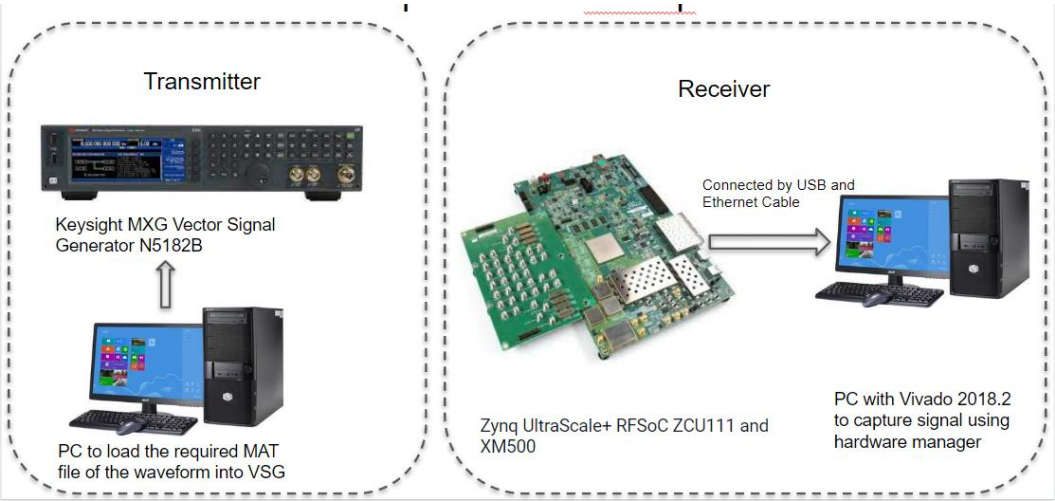


Figure 19 : Experimental set-up at CEWiT, IIT Madras

At CEWiT our aim was to test NOMA on the 5G testbed as well as to conduct its performance analysis. The testbed consisted of a transmitter and a receiver side as depicted in figure 10.

On the transmitter side we used Keysight MXG Vector Signal Generator N5182B preloaded with the corresponding MAT file of the required waveform. The experiment is conducted in an offline mode with transmitted data being pre-recorded in order to comply with the Vector Signal Generator. The Keysight MXG Vector Signal Generator is used as a gNB to transmit the signal. The required parameters of the waveform are set in MATLAB R2022b and the corresponding MAT file is loaded to the VSG over Wi-Fi.

Parameters	Value
Channel Bandwidth	100 MHz
Carrier Frequency	3.3501 GHz
Subcarrier Spacing	30 KHz
Size of Grid	273
Number of frames transmitted	1
Number of subframes transmitted	10
Total number of slots transmitted	20
Number of OFDM symbols transmitted per slot	14
Data Block Size	20
Data source	PN9

Table 8 : System Configurations for CEWiT experiments

On the receiver side we use Xilinx Zynq UltraScale+ RFSoC zcu111 board along with vivado 2018.2 software to capture a single frame which is then decoded and analysed. The receiver for UE1 and UE2 is the zcu111 board along with the XM500 balun board for the connection of antenna. The zcu111 board is connected to the PC by the means of USB and ethernet cable. The received signal is analysed in the Vivado 2018.2 hardware manager with trigger set up signifying the frame, subframe and slot number to be chosen. The selected portion of the signal is captured in the trigger window. The received data is in hex form which has to be converted to binary form. After we get the received grid decoding and analysis part is done with the help of MATLAB R2022b.

4.2 5G frame structure

A 5G frame has duration of 10 ms which consists of 10 subframes having 1ms duration each. Each subframe can have 2μ slots. The value of μ depends upon the subcarrier spacing (SCS) chosen. For example, if SCS is 15khz then the μ value is 1 and if SCS is 30khz then μ value is 2 and so on. Subframe is of fixed duration of 1ms where as slot length varies as the value of μ is different for different subcarrier spacing. For SCS of 15 KHz there is only one slot in the sub frame so the slot duration is 1 ms and for 30KHz there are 2 slots in the subframe so each slot is 500 μs. Each slot typically consists of 14 OFDM symbols.

A resource element is the smallest part of a resource block or resource grid. It has dimensions of 1 subcarrier in the frequency domain and 1 symbol in the time domain. The frequency domain bandwidth of a resource element is equal to the subcarrier spacing. A single resource element can accommodate a single modulation symbol, for example, a single QPSK, a single 16QAM or 64QAM or 256QAM modulation symbol. QPSK symbol represents 2 bits of information, so the capacity of a resource element corresponds to 2 bits when transferring a QPSK symbol.

A set of contiguous common resource blocks consists of a bandwidth part. They can be used to provide services to UE which do not support the full channel bandwidth. A UE can be configured with up to 4 downlink bandwidth parts per carrier and up to 4 uplink bandwidth parts per carrier. Only a single bandwidth part per carrier can be active in each direction. A UE receives the PDCCH and PDSCH only within an active downlink bandwidth part.

The resources allocated for PDSCH are within the bandwidth part (BWP) of the carrier. The symbol allocation of PDSCH indicates the OFDM symbol locations used by the PDSCH transmission in a slot. DM-RS symbol locations lie within the PDSCH symbol allocation. The positions of DM-RS OFDM symbols depend on the mapping type. The mapping type of PDSCH is either slot-wise (type A) or non-slot-wise (type B). PT-RS always occurs in combination with DM-RS and only when the network has configured PT-RS to be present.

Firstly, DL-SCH encoding is done which includes LDPC encoding, rate matching, code block segementation etc. After that there is PDSCH encoding in which we set the modulation parameters, the DM-RS parameters as well as the PT-RS parameters. Apart from this various parameters related to symbol allocation, alloted slot indices and allocated period in slots can also be set. The DMRS indices and values are kept the same for both signals. The channels and signals that form a synchronization signal block (SSB) are the primary and secondary synchronization signals and the physical broadcast channel. This block is created and mapped into a matrix representation and is added to the PDSCH of one of the users. The PDSCH for the UE1 signal and UE2 signal is generated separately and then added together to get the NOMA signal.

On the receiver side we receive the resource grid as a matrix. On this firstly channel estimation is done. After this we extract the PDSCH symbols from the received grid and associated channel estimates. Following this equalization of the PDSCH is done. Here we have made use of minimum mean squared error criterion for equalisation. MMSE criterion helps to minimize the intersymbol interference, additive noise effects and optimizes the equalizer coefficients. Hence the equalizer uses the channel estimate to compensate for the distortion introduced by the channel. PDSCH Decoding is done on the equalized PDSCH symbols to obtain the soft bit codewords. The channel state information (CSI) is a measure of the channel conditions for each equalised PDSCH symbol. CSI is used to weight the decoded soft bits after PDSCH decoding.

For UE1 successive interference cancellation needs to take place. So the soft bits obtained by PDSCH decoding are once again PDSCH encoded at UE1. These are then subtracted from the received symbol. The PDSCH of the signal obtained after this subtraction is once again decoded following the same steps as stated above.

4.3 Procedure

4.3.1 Setting up VSG

➢ VSG is switched on from the power button.

➢ Frequency button is used to set the centre frequency to 3.3501 GHz by
manually entering it. Ensure REF freq is OFF

➢ After entering frequency values select GHz option using the buttons on the right hand side of the screen.

➢ Once frequency has been selected press the AMPTD button to set amplitude. Select dBm after entering the numerical value.

➢ After that press MOD button. Ensure that you have transmitted the required MAT file for the waveform you wish to transmit.

➢ Now press “Select waveform”. Use ↓ arrow to go down and highlight required waveform. Press SELECT.

➢ Ensure ARB is shown as “ON”.

➢ Press MOD ON/OFF button and RF ON/OFF button to make them both ON.

4.3.2 Receiving data on zcu111 board using Vivado 2018.2

➢ The Xilinx ZYNQ Ultrascale zcu111 board is connected to the PC through the USB port and ethernet cable.

➢ Open HyperTerminal app and set the baud rate to “115200”, parity to “none”, stop bits “1”, data bits to “8”, flow control to “none”.

➢ Set the desired ip address of the board using “ifconfig” command.

➢ Load the required binaries into the WinSCP after opening a session with the corresponding IP address.

➢ Open the hardware manager on Vivado 2018.2. Add the corresponding .ltx file under trigger set-up.

➢ Select the timing probe from the list of probes. Select the radix as “binary” and set the values as from bits 23 to 0 to signify which frame, sub frame and slot number to capture.

➢ Under trigger mode settings set the trigger position in window as 1000 and the window data depth as 1,31,072.

➢ From the “tdata” select the probes 0 to 23 and convert it into a virtual bus named “real coefficients” and set the waveform type to analog and the radix to signed decimal.

➢ Capture the samples by pressing trigger on ila_1 and then export it as a .csv file.

➢ Further processing is done on the receiver chain created in MATLAB and constellation diagrams are obtained.

➢ Repeat the steps by selecting different modulation schemes for the 2 users and the compare the constellation diagrams.

4.4 Results

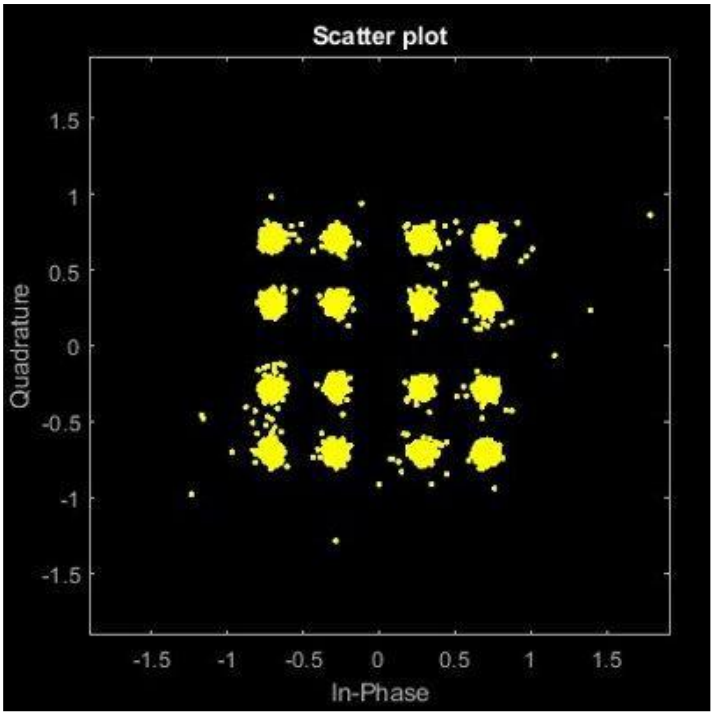


Figure 20 : UE1 at 10dBm transmit power with QPSK superimposed on QPSK(A1=0.8, A2=0.2)

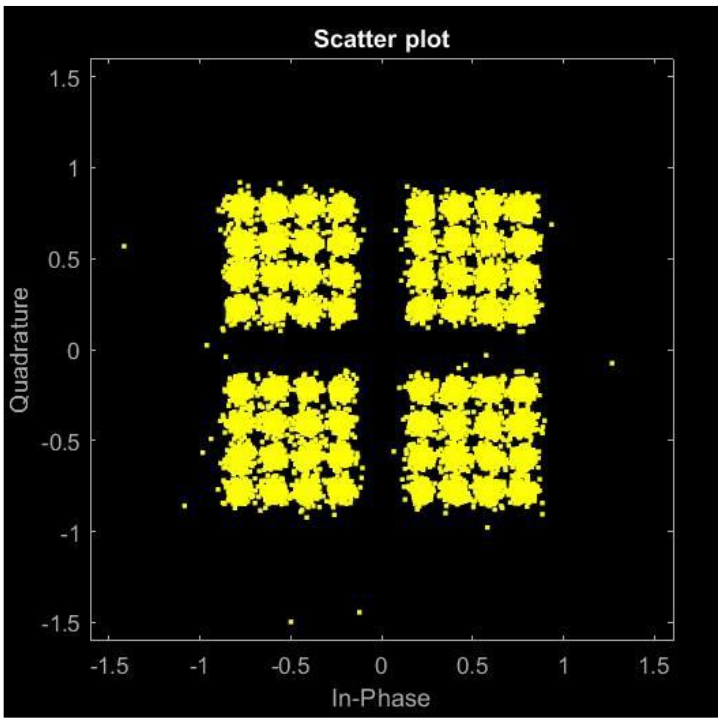


Figure 21 : For UE1 at 10dBm transmit power with 16 QAM superimposed on QPSK (A1=0.7, A2=0.3)

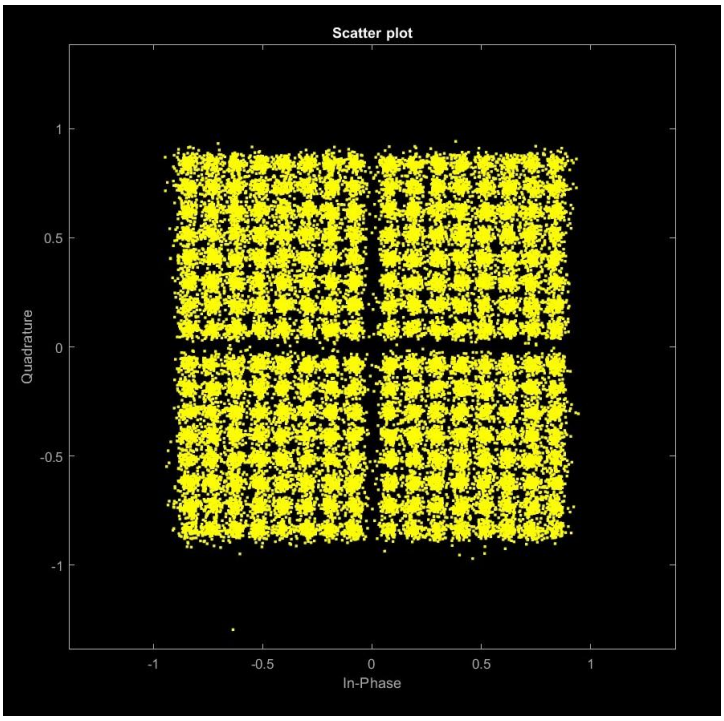


Figure 22 : For UE1 at 10dBm transmit power with 64QAM superimposed on QPSK (A1=0.65, A2=0.35)

4.5 Data Rate Calculation

The following formula is used to calculate the maximum achievable rate for the 5G NR signal of various modulation types,

$$\text{data rate (in Mbps)} = 10^{-6} \cdot v_{Layers} \cdot Q_m \cdot f \cdot R_{max} \cdot \frac{N_{PRB}^{BW,\mu} \cdot 12}{T_s^{\mu}} \cdot (1 - OH) \quad \text{- (11)}$$

Here, V_{layers} denotes the number of layers, Q_m is the modulation order, f is the scaling factor and R_{max} is the maximum LDPC coding rate given by 948/1024.

The number 12 refers to the number of sub-carriers in a resource block (RB). This is multiplied with $N^{BW,\mu}_{PRB}$ which is the maximum number of RBs that can be allocated to the UE for a given sub-carrier spacing (SCS) and bandwidth. OH is the overhead that has a fixed value of 0.14 for FR1 downlink.T^{μ}_{s} is the symbol time for a given value of μ. The formula assumes normal cyclic prefix. The following table depicts the results,

Modulation Type	Max. achievable data rate (in Mbps)
QPSK	146.06251
16QAM	292.12501
64QAM	438.18752

Table 9 : Maximum achievable data rate for different modulation types

Chapter 6

Conclusion and Future work

We have implemented NOMA for a two-user model and also conducted its performance analysis using NI-USRP 2952R. We have done profiling using the LabVIEW programming software and found out the execution time and memory usage for three different encoding techniques. When considering execution time, convolutional codes perform better than both hamming and LDPC. However, with respect to average kilobytes of memory used hamming out-performs LDPC and convolutional codes. LDPC is observed to have the highest execution time among the three. However, this execution time can be reduced by 93.78% if we make use of the look-up table while designing the LDPC encoder. This leads to the requirement of an additional 200 kilobytes of memory for the LDPC encoder.

While calculating bit error rate (BER) we have considered SNR values ranging from 0 to 12dB. Contrary to general observation, here we observed that LDPC performs better as compared to convolutional and hamming. Generally, convolutional codes are seen to have the lowest BER among the three encoders. However, the opposite is true here because we have made use of the soft-decision-based decoding algorithm for LDPC which is known to significantly reduce the BER.

The future work for this project can be undertaken in one of the following paths as follows,

- Increasing efficiency of channel encoders by adopting a different design approach in LabVIEW
- Testing for BER for each of the channel encoding techniques by considering different channel models apart from AWGN
- Extending this work for n-user NOMA and verifying if the results for 2-user NOMA holds true

REFERENCES

[1] R. De Gaudenzi, T. Garde, F. Giannetti, and M. Luise, "A performance comparison of orthogonal code division multiple-access techniques for mobile satellite communications," IEEE J. Sel. Areas Commun., vol. 13, no. 2, pp. 325-332, Feb. 1995.

[2] A. W. Scott and R. Frobenius, "Multiple Access Techniques: FDMA, TDMA, AND CDMA," in RF Measurements for Cellular Phones and Wireless Data Systems, IEEE, 2008, pp.413-429.

[3] F. Boccardi, R. W. Heath, A. Lozano, T. L. Marzetta, and P. Popovski, "Five disruptive technology directions for 5G," IEEE Commun. Mag., vol. 52, no. 2, pp. 74-80, Feb. 2014.

[4] L. Dai, B. Wang, Z. Ding, Z. Wang, S. Chen, and L. Hanzo, "A Survey of Non-Orthogonal Multiple Access for 5G," IEEE Commun. Surveys Tuts., vol. 20, no. 3, pp. 2294-2323, May 2018.

[5] Q. Sun, S. Han, C. -L. I, and Z. Pan, "On the Ergodic Capacity of MIMO NOMA Systems," IEEE Wireless Commun. Lett., vol. 4, no. 4, pp. 405-408, Aug. 2015.

[6] Y. Liu, G. Pan, H. Zhang, and M. Song, "On the Capacity Comparison Between MIMO-NOMA and MIMO-OMA," IEEE Access, vol. 4, pp. 2123-2129, 2016.

[7] P. Xu, Z. Ding, X. Dai, and H. V. Poor, "A New Evaluation Criterion for Non-Orthogonal Multiple Access in 5G Software Defined Networks," IEEE Access, vol. 3, pp. 1633-1639, 2015.

[8] S. M. R. Islam, N. Avazov, O. A. Dobre and K. -s. Kwak, "Power-Domain Non-Orthogonal Multiple Access (NOMA) in 5G Systems: Potentials and Challenges," IEEE Commun. Surveys Tuts., vol. 19, no. 2, pp. 721-742, 2017.

[9] G. Wang, Z. Zhao, and T. Xu, "A Joint Interference Cancellation Method For Non-Orthogonal Multiple Access Uplink Signals," in 14th IEEE Int. Conf. Signal Process. (ICSP), Beijing, China, 2018, pp. 694-697.

[10] Z. Ding, X. Lei, G. K. Karagiannidis, R. Schober, J. Yuan, and V. K. Bhargava, "A Survey on Non-Orthogonal Multiple Access for 5G Networks: Research Challenges and Future Trends," IEEE J. Sel. Areas Commun., vol. 35, no. 10, pp. 2181-2195, Oct. 2017.

[11] Y. Cai, Z. Qin, F. Cui, G. Y. Li, and J. A. McCann, "Modulation and Multiple Access for 5G Networks," IEEE Commun. Surveys Tuts., vol. 20, no. 1, pp. 629-646, 2018.

[12] A. Bishnu and V. Bhatia, "An IEEE 802.22 transceiver framework and its performance analysis on software-defined radio for TV white space," Telecommun. Syst., vol. 68, pp. 657-668, 2018.

[13] S. Vanka, S. Srinivasa, Z. Gong, P. Vizi, K. Stamatiou, and M. Haenggi, "Superposition Coding Strategies: Design and Experimental Evaluation," IEEE Trans. Wireless Commun. , vol. 11, no. 7, pp. 2628-2639, July 2012.

[14] X. Xiong, W. Xiang, K. Zheng, H. Shen, and X. Wei, "An open source SDR-based NOMA system for 5G networks," IEEE Wireless Commun., vol. 22, no. 6, pp. 24-32, Dec. 2015.

[15] X. Wei, Z. Geng, H. Liu, K. Zheng, and R. Xu. "A portable SDR nonorthogonal multiple access testbed for 5G networks." in IEEE 85th Veh. Technol. Conf., Sydney, Australia, 2017, pp. 1-5.

[16] Y. Liu, W. Yi, Z. Ding, X. Liu, O. A. Dobre, and N. Al-Dhahir, "Developing NOMA to Next Generation Multiple Access: Future Vision and Research Opportunities," IEEE Wireless Commun., vol. 29, no. 6, pp. 120-127, Dec. 2022.

[17] Z. Xiao, H. Dong, L. Bai, D. O. Wu and X. -G. Xia, "Unmanned Aerial Vehicle Base Station (UAV-BS) DeploymentWith Millimeter-Wave Beamforming," IEEE Internet Things J., vol. 7, no. 2, pp. 1336-1349, Feb. 2020.

[18] Y. Saito, A. Benjebbour, Y. Kishiyama, and T. Nakamura, “System level performance evaluation of downlink non-orthogonal multiple access (NOMA),” in Proc. IEEE Annu. Symp. PIMRC, London, U.K., Sep. 2013, pp. 611–615.

[19] X. Wei, H. Liu, Z. Geng, K. Zheng, R. Xu, Y. Liu, and P. Chen. "Software Defined Radio Implementation of a Non-Orthogonal Multiple Access System Towards 5G," IEEE Access, vol. 4, pp. 9604-9613, 2016.

[20] P. Demestichas, A. Georgakopoulos, D. Karvounas, K. Tsagkaris, V. Stavroulaki, J. Lu, C. Xiong, and J. Yao, “5G on the horizon: Key challenges for the radio-access network,” IEEE Veh. Technol. Mag., vol. 8, no. 3, pp. 47–53, Sep. 2013.

[21] Y. Qi, X. Zhang, and M. Vaezi, "Over-the-Air Implementation of NOMA: New Experiments and Future Directions,"IEEE Access, vol. 9, pp. 135828-135844, 2021.

[22] Z. Ding, F. Adachi, and H. V. Poor, "The Application of MIMO to Non-Orthogonal Multiple Access," IEEE Trans. Wireless Commun., vol. 15, no. 1, pp. 537-552, Jan. 2016.

[23] Z. Ding, M. Peng and, H. V. Poor, "Cooperative Non-Orthogonal Multiple Access in 5G Systems,"IEEE Commun. Lett., vol. 19, no. 8, pp. 1462-1465, Aug. 2015.

[24] Z. Ding, Y. Liu, J. Choi, Q. Sun, M. Elkashlan, I. Chih-Lin, and H. V. Poor, "Application of non-orthogonal multiple access in LTE and 5G networks," IEEE Commun. Mag., vol. 55, no. 2, pp. 185–191, Feb. 2017.

[25] S. Sharma, K. Deka, V. Bhatia and A. Gupta, "Joint Power-Domain and SCMA-Based NOMA System for Downlink in 5G and Beyond," in IEEE Commun. Lett., vol. 23, no. 6, pp. 971-974, June 2019.

[26] A. S. Parihar, P. Swami, V. Bhatia and Z. Ding, "Performance Analysis of SWIPT Enabled Cooperative-NOMA in Heterogeneous Networks Using Carrier Sensing," in IEEE Trans. Veh. Technol., vol. 70, no. 10, pp. 10646-10656, Oct. 2021.

[27] Idris, Mohd & mohd ali, Darmawaty & Abdul Razak, Nur Idora & Idris, Azlina & Ahmad, Hasyimah. " Performance analysis of NOMA using different coding techniques", Journal of Physics: Conference Series, 2020.

[28] Hsu, C., Wang, N., Chan, WY. et al., " Improving a Power Line Communications Standard with LDPC Codes", EURASIP J. Adv. Signal Process., 2007.

[29] Z. Ding, P. Fan and H. V. Poor, "Impact of User Pairing on 5G Nonorthogonal Multiple-Access Downlink Transmissions," in IEEE Transactions on Vehicular Technology, vol. 65, no. 8, pp. 6010-6023, Aug. 2016.

[30] M.P.C. Fossorier, "Quasi-cyclic Low-Density Parity-Check Codes from Circulant Permutation Matrices," IEEE Trans. Inf. Theory, vol. 50, no. 8, pp. 1788-1793, Aug. 2004.

[31] M. Aldababsa, C. Göztepe, G. K. Kurt and O. Kucur, "Bit Error Rate for NOMA Network," in IEEE Commun. Lett., vol. 24, no. 6, pp. 1188-1191, June 2020.