Multi-User MIMO-OFDM MODEM

Thesis submitted in partial fulfilment of Master of Technology (Hons)

to

Electronics and Electrical Communication Engineering

Indian Institute of Technology, Kharagpur

by

ABHINAV KUMAR VERMA

(14EC34002)

Under the guidance of

DR. SUVRA SEKHAR DAS

Department of Electronics and Electrical Communication Engineering

Indian Institute of Technology, Kharagpur-721302

April, 2019

Department of Electronics and Electrical Communication Engineering

Indian Institute of Technology, Kharagpur, India

CERTIFICATE

This is to certify that the thesis titled "**Multi-User MIMO-OFDM MODEM**" submitted by **Abhinav Kumar Verma** as M. Tech. Project for the academic year 2018-2019 Roll No. 14EC34002, to Indian Institute of Technology, Kharagpur, for the partial fulfilment of the Master of Technology, is a record of work carried out by him under my supervision and guidance. The report, in my opinion, is worthy of consideration for the partial fulfilment of award of the degree of Master of Technology in Engineering in accordance with the rules and regulations of the Institute.

Dr. Prof. Suvra Sekhar Das,
G S Sanyal School of Telecommunication
Indian Institute of Technology
Kharagpur, 721302

Date:
Place: Kharagpur

DECLARATION

I certify that

a) This work is done by me under the guidance of my supervisor while in candidature for a Master of Technology degree at this Institute.

b) This work has not been submitted to any other Institute for any degree or diploma.

c) I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.

d) Whenever I have used materials from other sources, I have given due credit to them by citing them in the text of the thesis and giving their details in the references. Further, I have taken permission from the copyright owners of the sources, whenever necessary.

Date: 25/04/2019	Abhinav Kumar Verma
Place: Kharagpur	(14EC34002)

ACKNOWLEDGEMENT

Every project is a sum of efforts of many, and this project too is a no exception. I wish to express my sincere gratitude to the people below who participated in the fulfilment of this project.

I take this opportunity to express my deepest gratitude to my project guide Dr. Suvra Sekhar Das for giving me an opportunity to work under his esteemed guidance. I am greatly indebted to him for his invaluable advice and support.

I would also like to thank Dr. Sandeep Mukherjee (Assistant Professor, Techno India University), Mr Shashank Tiwari (PhD, GSSST, IIT Kharagpur) and Mr. Sourav Chatterjee (PhD, GSSST, IIT Kharagpur), for their constant help and guidance during the course of my project work.

Date: 25/04/2019 Abhinav Kumar Verma
Place: Kharagpur (14EC34002)

ABSTRACT

The aim of this project to study and implement physical layer architecture of the 802.11n and 802.11ac on MATLAB. This has been achieved completely and the results have been shown to match those present in the literature. Crucial blocks like Forward Error Correction using Low Density Parity Check code was studied and implemented with Sum Product Algorithm for decoding. MIMO schemes such as Maximal Ratio Codebook, Space Time Block Coding, Codebook, and SVD were implemented to support all possible scenarios with up to four transmit chains. The preambles compliant with respective standards were generated and appended to the Physical layer protocol data unit (PPDU).

To replicate real-life scenarios, TGn channel models were implemented apart from Rayleigh and Additive White Gaussian Noise. On the receiver side, an algorithm for packet detection was developed based on double sliding window method. A theoretical study for optimal detection based on Likelihood Ratio Test was performed to find thresholds. The same was implemented and integrated into the simulator. Algorithms for synchronisation, MIMO channel estimation and equalisation were implemented. A brief study of Channel State Information feedback mechanism in implicit and explicit form was conducted.

The 802.11n modem was then extended to multi-user scenario compliant with IEEE 802.11ac standards. Two channel equalisation techniques for MU MIMO were implemented and tested. The modem was also extended to incorporate massive MIMO scenario that is not yet a part of 802.11 standards. The extension to massive MIMO led to huge performance gains.

A GUI based executable simulator was created after packaging all the relevant work to get modular and user-friendly IEEE simulator. This has been utilised as a teaching aid and for development of familiarity with wireless communication.

INDEX

CHAPTER 1

ARCHITECTURE OF 802.11n PHYSICAL LAYER

The demands for the wireless connections in high data rates are increasing rapidly. This has led to the development of Wireless Local Area Network (WLAN) technology. Unlike the cellular network that offers more coverage and mobility, the core marketing value of WLAN is to cater to the high-density demand with higher throughput and speed than the cellular network.

IEEE 802.11 Standards:

IEEE 802.11 provides specifications for implementation of the WLAN. It is concerned with the Physical and the Media Access Control layer of the communication link. The first 802.11 standard (base version) was released in 1997, and then after, there have been subsequent amendments to it. These standards provide the basis for products that claim Wi-Fi brand.

802.11 Protocol	**Frequency (GHz)**	**Bandwidth (Mhz)**	**Data Rate (Mbps)**	**MIMO streams**	**Modulation**
802.11 – 1997	2.4	22	1, 2	NA	DSSS, FSSS
A	5	20	6, 9, 12, 18, 24, 36, 48, 54	NA	OFDM
B	2.4	22	1, 2, 5.5, 11	NA	OFDM
G	2.4	20	6, 9, 12, 18, 24, 36, 48, 54	NA	OFDM
N	2.4/5	20	Up to 288.8	4	MIMO – OFDM
		40	Up to 600		
Ac	5	20	Up to 346.8	8	MIMO – OFDM
		40	Up to 800		
		80	Up to 1733.2		
		160	Up to 3466.8		
Ax	2.4/5		Up to 10.53 Gbps		MIMO – OFDM

Table 1.1: *802.11 protocols and their key features.*

IEEE 802.11n:

802.11n is an amendment to the 802.11 series with the amendment in 2009 being the most popular one. One of the major change in Wi-Fi came with the 802.11n as it brought MIMO (Multiple Input Multiple Output). The impact of introducing MIMO was an improvement of network throughput over previous standards. 802.11n also introduces 40MHz bandwidth for high throughput. MAC layer feature of Channel Bonding was also introduced to support the high throughput.

Physical Layer Architecture of 802.11n:

The 802.11n standards support 802.11a in its legacy mode, and hence the architecture of the physical layer for 802.11n is influenced by the 802.11a's (Clause 17) architecture.

It will be helpful to have a look at the 802.11n frame format, and that shall give us an idea about the physical layer needed.

Source: IEEE 802.11n Part11, Amendment 5, Fig. 20.1, Page 259

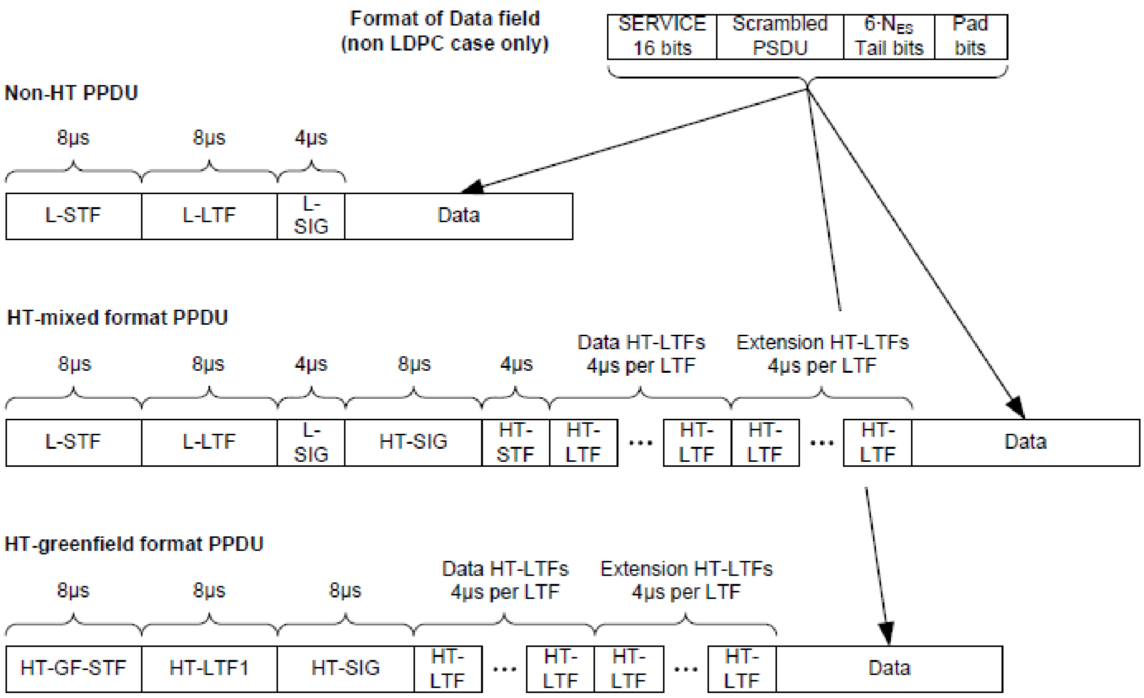


***Fig 1.1:** Physical layer frame format*

The Non - High Throughput PPDU is same as the PPDU of 802.11a. This is ensured to have legacy support so that devices working on older 802.11a can still communicate with newer devices working on 802.11n protocol.

To generate the L-STF (Legacy Short Training Field), L-LTF (Legacy Long Training Field) and L-SIG (Legacy Signal) and HT-SIG (High Throughput Signal), the transmitter design is as shown below. It must be noted that the high throughput portion of the PPDU is generated using another transmitter than the one given below, hence it is important to distinguish the two transmitters. We shall call the transmitter for the non-HT part as Transmitter 1 and the transmitter for HT portion of the PPDU, transmitter 2.

Source: IEEE 802.11n Part11, Amendment 5, Fig. 20.2, Page 261

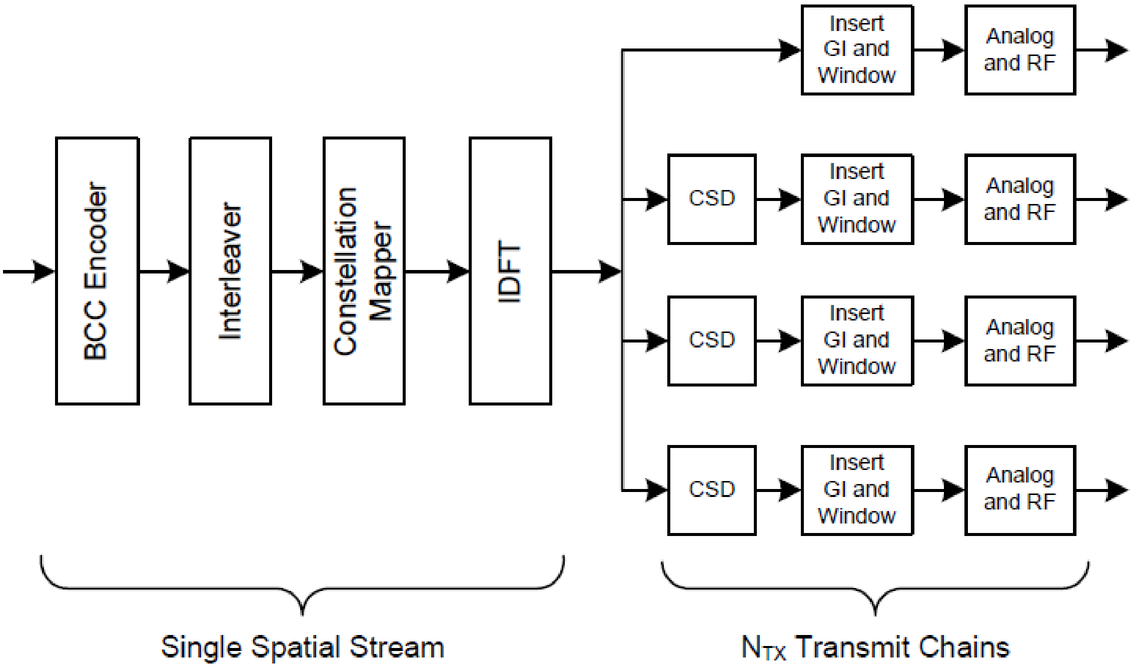


***Fig 1.2:** Transmitter 1 (for the non-HT part)*

From the above architecture, it is evident that the data transmitted through various transmit chains are cyclically shifted versions in the time domain. In fact, the first transmit chain produces preamble of the PPDU that is exactly same as the preamble of 802.11a protocol and this is necessary to ensure legacy support.

The crucial blocks of the transmitter one are explained below:

BCC encoders: It encodes the data to enable error correction. The encoder uses binary convolutional encoder followed by a puncturing device.

Interleaver: It interleaves the bits from BCC encoder (changes the order of bits) to prevent long sequences of adjacent noisy bits from entering the BCC decoder receiver.

Constellation mapper: It maps the sequence of bits in each spatial stream to constellation points (complex numbers).

IDFT: Inverse discrete Fourier transform converts a block of constellation points to a time domain block.

CSD: Cyclic shift insertion is where the insertion of the cyclic shifts prevents unintentional beamforming. CSD insertion may occur before or after the IDFT. However the effect is a cyclic shift in the time domain.

GI: Guard Interval insertion prepends to the symbol a circular extension of itself. For the high throughput portions, GI of 400ns is proposed apart from existing 800ns.

Windowing: It optionally smooths the edges of each symbol to increase spectral decay.

The following transmitter is used to generate high throughput part of the preamble and the data for PPDU.

Source: IEEE 802.11n Part11, Amendment 5, Fig. 20.3, Page 262

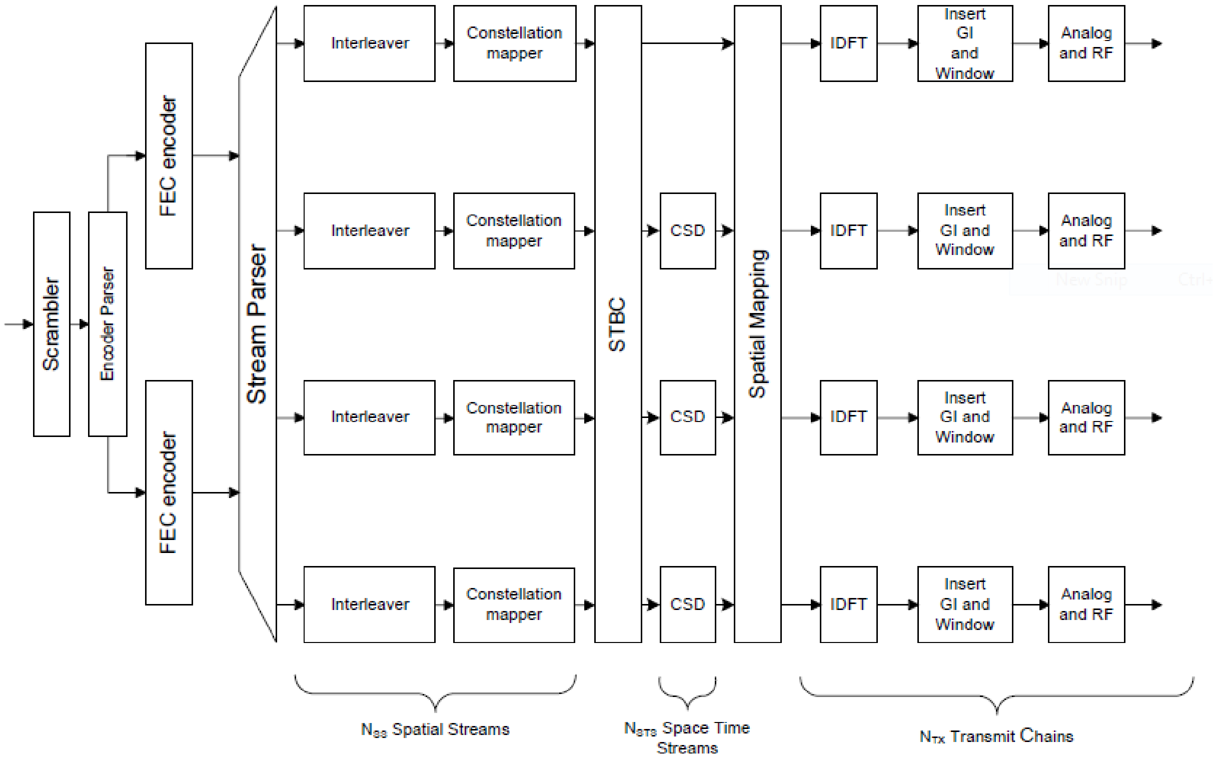


***Fig 1.3:** Transmitter 2 (for HT part)*

The crucial blocks of transmitter 2 are responsible for enabling high throughput in 802.11n. Many of the blocks used by transmitter 2 are described above, and other blocks are as followed:

Scrambler: It scrambles the data to reduce the probability of long sequences of zeros or ones.

Encoder parser: The encoder parser is used to demultiplexes the scrambled bits among N_{ES} (number of BCC encoders for the Data field) BCC encoders, in a round robin manner, if BCC encoder is used. For LDPC encoder the encoder parser is overlooked.

Stream parser: It divides the outputs of the encoders into blocks that are sent to different interleaver and mapping devices. The sequence of the bits sent to an interleaver is called a spatial stream. The number of stream parser is decided by MCS index which is discussed later.

STBC: Space Time Block Coding encoder spreads constellation points from N_{SS} spatial streams into N_{STS} space-time streams using a space-time block code.

Spatial Stream, Space Time Stream and Transmit Chain:

Spatial Stream: It refers to the number of data streams that have to be modulation using constellation mapper. Essentially N_{SS} spatial streams mean an increment of N_{SS} times in the data rate.

Space Time Stream: It refers to the number of spatial streams on which Alamouti scheme has been applied. So, if N_{STBC} spatial stream undergoes Alamouti scheme, then we have $N_{SS} = N_{STS} + N_{STBC}$.

Transmit Chain: The transmit chain refers to the number of antennae that will be used to transmit space time streams. A precoder matrix is used to map the space time stream on transmit chains.

Receiver for 802.11n:

A receiver with focus on algorithms running on the receiver side looks like the one given below.

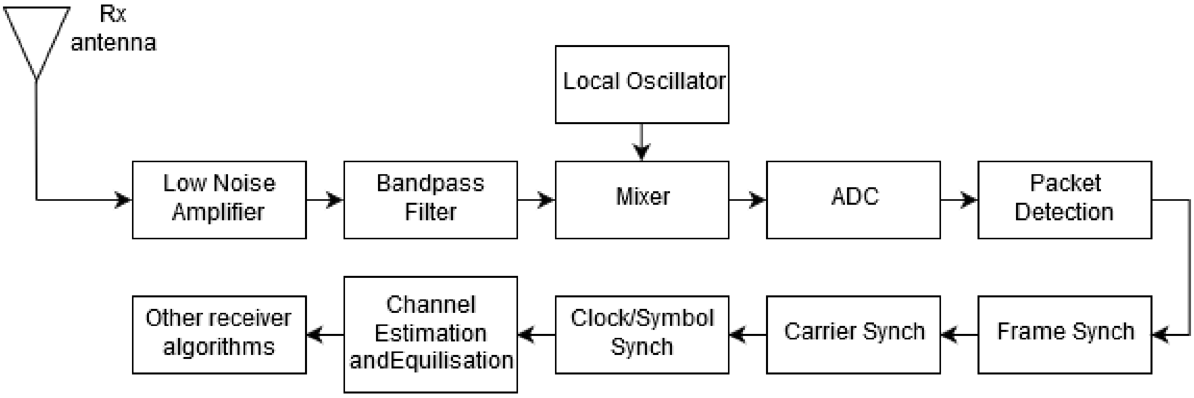


Fig 1.4: *Receiver with detection and synchronisation algorithms.*

Primary receiver sections:

RF section: The RF section consists of receiver antenna, LNA, bandpass filter, mixer, and low pass filter. This section is responsible for bringing down the frequency of the received signal by demodulation with minimal distortion.

ADC and AGC: The analog to digital converter is used to convert the down converted signal from the RF section to digital samples. The sampling time of the ADC must be equal to that of bandwidth of the signal or higher. The automatic gain control is used to set the dynamic range of the ADC. The received signal generally has huge dynamic range that needs to be corrected on the fly and AGC does that job.

Packet Detection: Whenever a packet is available at the receiver, it needs to be detected or the receiver will not know about the presence of packet. A packet detection algorithm acts on available samples to detect presence of packet and gives an estimate of start of packet.

Frame synchronisation: Frame synchronisation algorithm is used to detect the exact starting point of received packet. Autocorrelation and cross correlation of L-STF is generally used for frame synchronisation.

Carrier synchronisation: Due to offset in the local oscillator, there is usually some offset in the carrier frequency and that leads to phase error that accumulated with time. To avoid the same, carrier synchronisation is needed. The pilot subcarriers are utilised for this purpose.

Channel estimation: The MIMO channel between transmitter and the receiver must be known to decode transmitted data. Channel estimation using the LTF portion of the preamble is used for this purpose.

MIMO Channel equalisation: The MIMO channel distorts the transmitted signal that must be corrected using the estimated channel. Zero forcing, MMSE, and MLE are popular channel equalisation techniques.

MIMO equalisation: The transmitter and equalisation techniques of 802.11 is developed in a manner that MIMO equalisation is taken care in the channel equalisation itself as the spatial mapping is treated as a part of channel.

Other receiver algorithms consist of inverse of all the operations that are performed on the transmitter as shown in Figure 1.2 and 1.3. These consist of window removal, GI removal, FFT, MIMO processing, CSD removal, STBC processing etc. It should also be noted that there is no specific receiver specified in the standards and in fact the receiver design is a valuable intellectual property that organisations tend to preserve. As a result, the receiver design is an open challenge and various algorithms and methods can be tried for optimisation.

It should also be noted that some of the receiver algorithms described as a part of “other receiver algorithms” may fall before mentioned and it is subject to receiver design.

CHAPTER 2

PREAMBLE GENERATION

Overview:

The transmitter architecture described in the previous chapter has been implemented on MATLAB to generate Legacy mode and High Throughput portions of the preamble. The Greenfield PPDU is not targeted in this study.

Components of Preamble:

All the components of Preamble, their description and duration are given below

Component	**Duration (in μs)**	**Description**
L-STF	8	Legacy (non-HT) Short Training Field
L-LFT	8	Legacy (non-HT) Long Training Field
L-SIG	4	Legacy (non-HT) Signal Field
HT-SIG	8	High throughput Signal Field
HT-STF	4	High Throughput Short Training Field
HT-DLTF	4	High Throughput Data Long Training Field
HT-ELTF	4	High Throughput Extended Long Training Field

Table 2.1: *Components of Preamble and their description.*

Non High Throughput Portion:

L-STF: L-STF is the first signal of the PPDU. It is used for Packet Detection, Frame Synchronisation, Frequency Offset Synchronisation. The duration of L-STF is $8\mu s$ which corresponds to two OFDM symbols. The L-STF contains of ten 16-symbol repeated sequence. The repetition in the L-SFT makes it perfect for synchronisation using cross correlation.

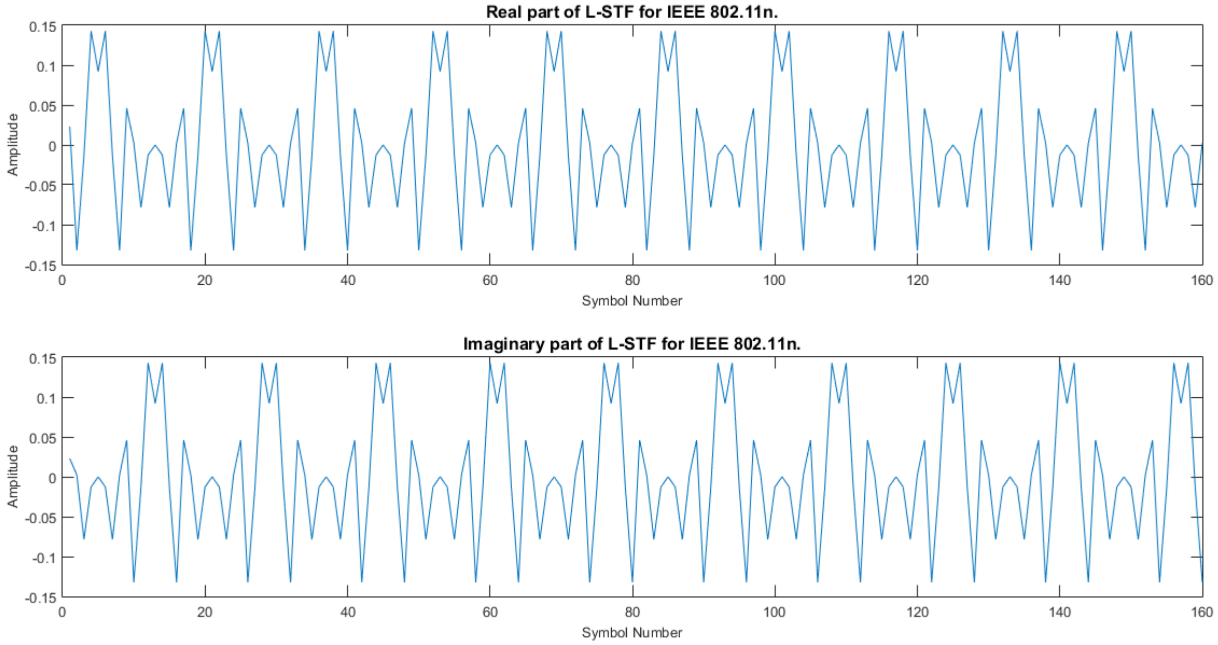


Fig 2.1: *Real and Imaginary part of L-STF.*

L-LTF: L-LTF is the signal portion of the preamble that is used to estimate the channel and hence it is helpful for successful demodulation. A pre-known sequence of data is passed through the transmitter 2 to generate L-LTF. The known signal is then compared to the received signal to obtain channel estimates.

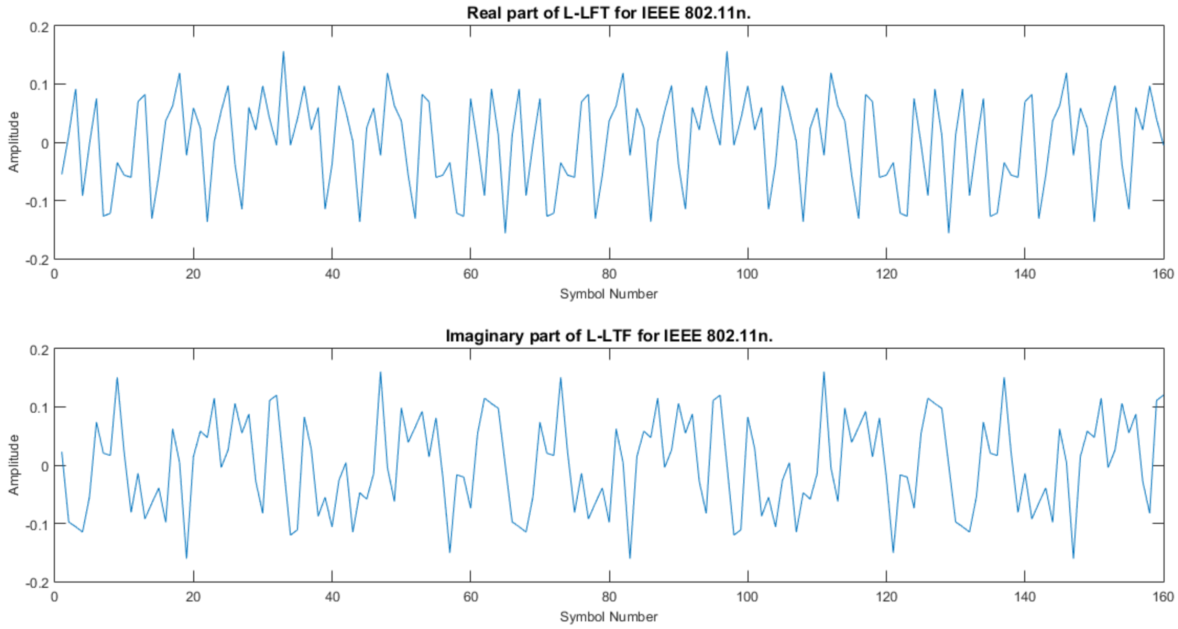


Fig 2.2: *Real and Imaginary part of L-LTF.*

L-SIG: L-SIG is the part of the preamble that informs the receiver about the rate and length of the packet. Structure of the L-SIG is given below:

Source: IEEE 802.11n Part11, Amendment 5, Page 262

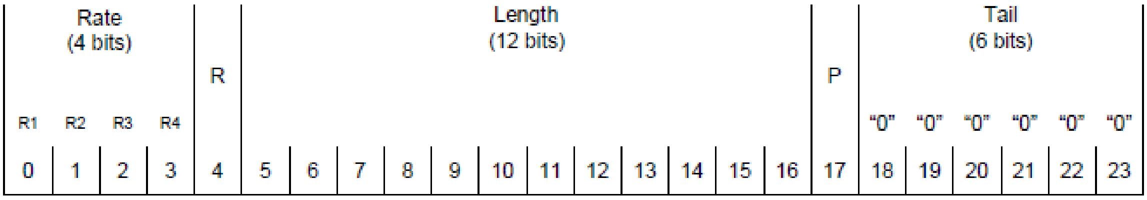


Fig 2.3: *Structure of L-SIG.*

The R bit is '0' and the Parity bit is set to the even parity check of bits 0 to 16. The bit sequence is passed through transmitter 1 using BPSK modulation.

High Throughput Portion:

The HT part of the preamble has been generated for the following configuration. It should be noted that the preamble changes with the configuration and MIMO Precoder matrix.

Component	Value
MCS	8
N_{SS}	2
N_{STBC}	1
N_{STS}	3
N_{TX}	4
N_{RX}	4
$N_{LT-DLTF}$	4
Q (Precoder Matrix)	$\sqrt{\frac{3}{4}}\begin{bmatrix}1 & 0 & 0\\0 & 1 & 0\\0 & 0 & 1\\1 & 0 & 0\end{bmatrix}$

Table 2.2: *Configuration for generation of Preamble.*

HT-STF: The STF of HT portion of the preamble is primarily used for Automatic Gain Control of MIMO streams. Different MIMO streams might have different power levels due to unintended beamforming. Hence it is important to have the HT-STF.

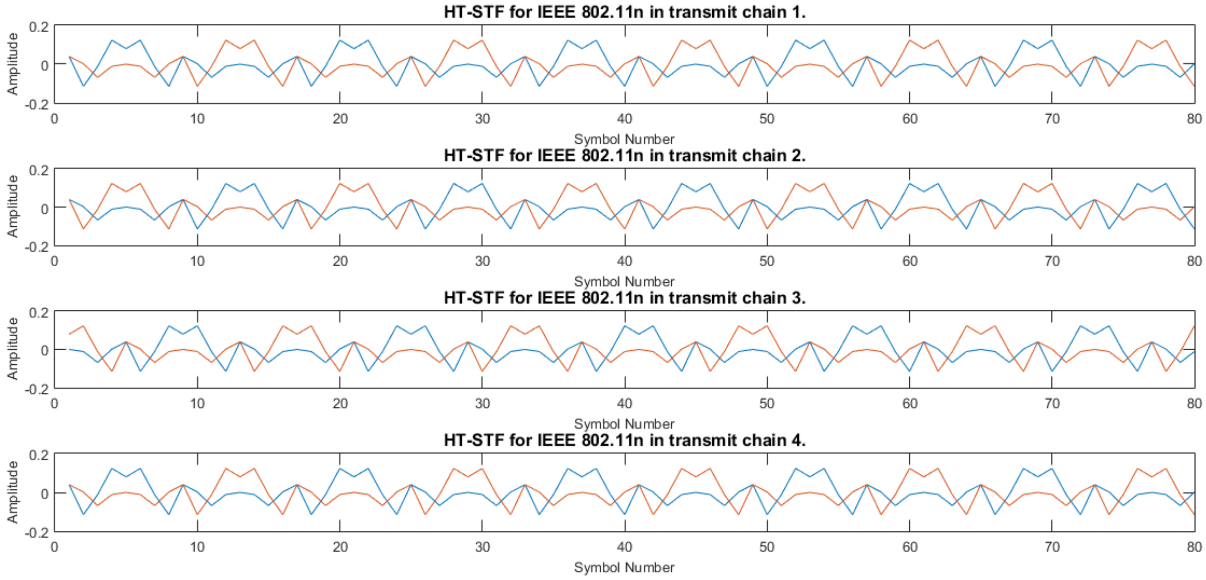


Fig 2.4: *Real and Imaginary part of HT-STF.*

HT-DLTF: The LTF of HT portion of the preamble is primarily used for MIMO channel estimation. Different MIMO streams will always lead to the formation of a different channel between the transmitter and the receiver antenna. These channels are estimated using the HT-DLTFs transmitted. It must be noted that the number of HT-DLTF depends on the number of Space-Time Streams present. For N_{STS} = 1, 2, or 4, we have $N_{HT\text{-}DLFT} = N_{STS}$. However, for N_{STS} = 3; we have N_{STS} = 4. This is done to ensure invertibility of the pre-known sequence matrix.

The HT-DLTF generated for the configuration given above is of length 320 symbols which cannot be displayed properly in short space. Hence the length has been truncated to 80 for display

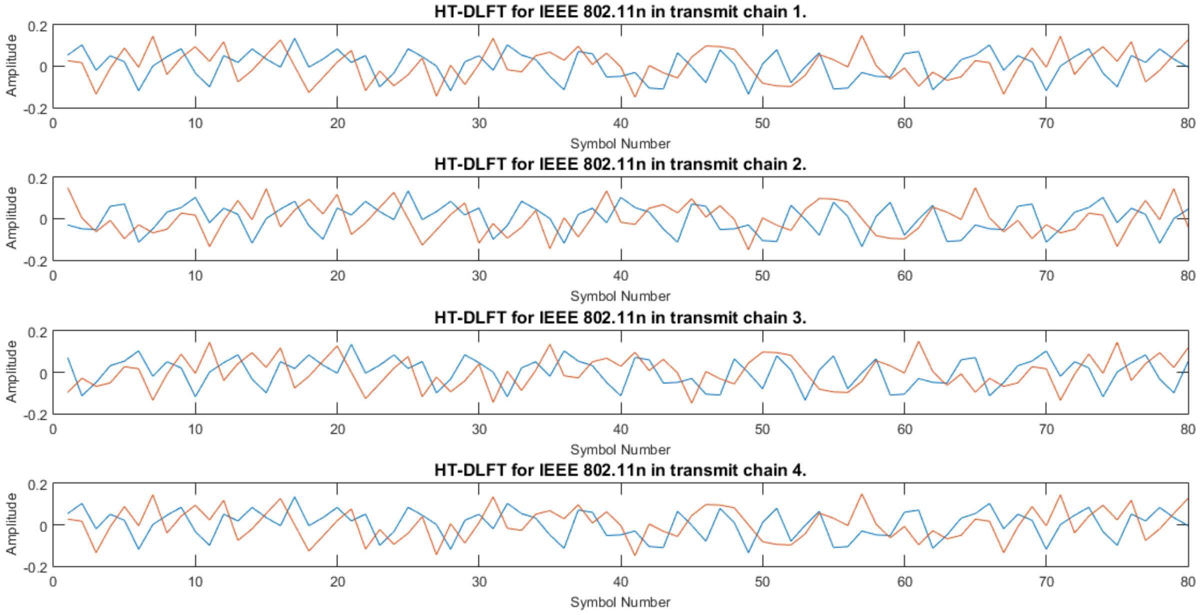


***Fig 2.5:** Real and Imaginary part of HT-DLTF.*

CHAPTER 3

LOW DENSITY PARITY CHECK CODES

Overview:

A communication system transmits data from a source to a receiver through a channel or medium such as wired or wireless. The reliability of received data depends on the channel medium and external noise. The noise can create interference to the signal and introduce errors in transmitted data. Shannon through his coding theory showed that reliable transmission could be achieved only if the data rate is less than that of channel capacity. The theorem states that it is possible to transmit a sequence of message at rate less than the channel capacity with an infinitesimal probability of bit error. However, the theorem doesn't state any particular method to achieve channel's capacity. To reach the channel capacity it is important to have error detection and correction.

Error detection and correction can be achieved by adding redundant symbols to the original data called error correction codes (ECCs). Without ECCs data needs to be retransmitted if an error is detected in the received data. ECCs are also called as forward error correction (FEC) as it is to correct bits without retransmission.

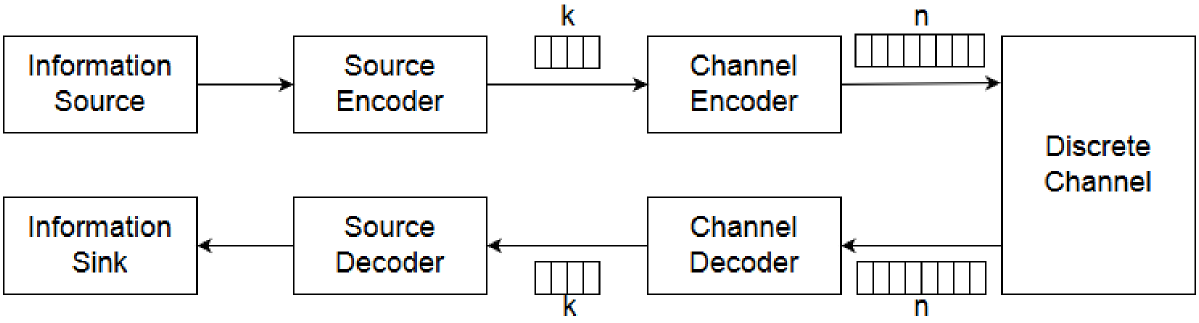


Fig 3.1: *Information Theoretic Block Diagram of a Communication Link*

Figure 3.1 shows a communication link diagram from information theoretic point of view. Data is generated from an information source and then passed to a source encoder that removes the redundancy from the data based upon the statistical properties of it. The output of the source encoder is passed through channel encoder that introduces controlled redundancy to the data. This coded information signal passes through a discrete channel that may introduce noise and other abnormalities to it. The output of the channel is fed to channel decoder that tries to utilise the redundancy introduced by channel encoder to estimate the correct transmitted signal. The decoded data is passed to the final information sink through source decoder.

FEC: Forward Error Correction

Forward Error Correction is a technique for controlling error in data transmission over an unreliable or noisy communication channel. The central is the transmitter encodes the message in a redundant way by using error correction code. The redundancy allows the receiver to detect a limited number of errors that may occur anywhere in the message, and often to correct these errors without retransmission. Unlike FEC, another popular error correction technique is Automatic Repeat Query which requires a reverse channel to request retransmission of data. Forward error correction is also known as Channel Coding.

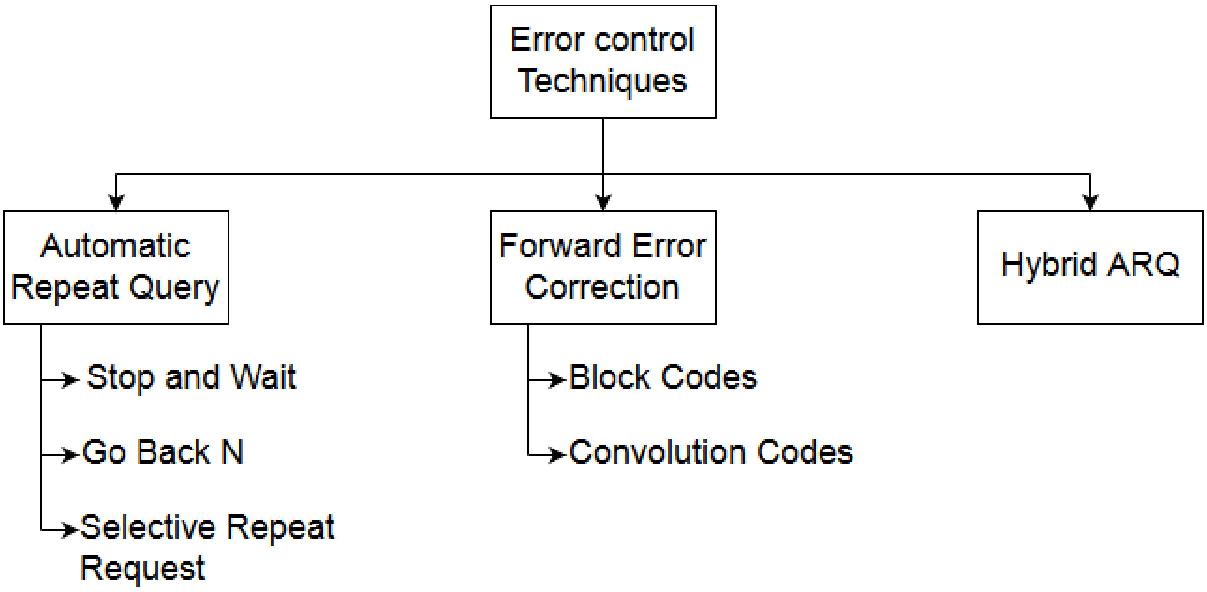


Fig 3.2: *Error Control Techniques and their types*

Figure 3.2 shows the three broad classes of Error Control Techniques. Automatic Repeat Query (Automatic Repeat Request) and Forward Error Correction have been discussed above. Hybrid ARQ uses features from both FEC as well as ARQ.

Classification of Forward Error Correction Techniques

Forward error correction techniques are primarily classified into two subcategories that are Linear and Non-Linear. In a linear code, the sum of two codewords always leads to another valid codeword, while this is not always true in the case of non-linear code. Most of the popular and well-studied Forward Error Correction Codes are linear. The linear FECs can be either Block codes or Convolution Codes.

<u>Block Codes</u>: The term Block code refers to an error correcting code that acts on a block of k bits of input data to produce n bits of output data. Consequently, the block coder is a memoryless device. There is a vast number of examples for block codes, many of which have a wide range of practical applications. Examples of

block codes are Reed-Solomon codes, Hamming codes, Hadamard codes, Golay codes etc.

Convolution Codes: It is a type of error-correcting code that generates parity symbols via the sliding application of a Boolean polynomial function to the data stream. The sliding application represents the convolution of the encoder over the data. Clearly, the convolution encoding requires memory.

The forward error correction techniques can be classified as systematic and non-systematic. In systematic codes, the actual information bits appears unaltered in the encoded data, and the redundant bits are added. However, in non-systematic codes, the actual information bits do not occur, and they need to be extracted completely from the encoded message during the decoding process.

Low Density Parity Check Codes

Low Density parity-check (LDPC) codes are a class of linear block forward error correction codes (FEC) which provide near-capacity performance. They were invented by Robert Gallager in 1962. However, these codes were neglected for more than 30 years, since the hardware requirements for the encoding process could not be fulfilled by the technology of that era. With increased computational capabilities and the development of relevant theories such as the belief propagation algorithm and Turbo codes, LDPC codes were rediscovered by Mackay and Neal in 1996.

LDPC codes are linear block codes that may be denoted by (n, k) or (n, w_c, w_r), where ‘n’ is the length of the codeword, ‘k’ is the length of the message bits, w_c is the column weight (i.e. the number of nonzero elements in a column of the parity-check matrix), and w_r is the row weight (i.e. the number of nonzero elements in a row of the parity-check matrix).

The name Low Density Parity Check Codes reflects two essential features of it, and they are:

Parity Check: Like all other block codes, an LDPC code is always associated with a parity check matrix (usually denoted by ‘H’). H is a binary matrix that must satisfy

$$cH^T = 0$$

$$where\ c\ is\ a\ valid\ codeword\ vector\ and\ 0\ is\ null\ matrix.$$

Low Density: The parity check matrix ‘H’ must be sparse. This essentially means that the number of 1’s in the matrix is much less than the number of 0’s in the matrix. This property of LDPC codes ensures that the encoding and decoding complexity of LDPC codes remains less.

The basic matrices, equations, and denotations related with the LDPC coding are as given below:

$\boldsymbol{Entity}$	**Description**
k	Length of an uncoded message to be processed at a time.
n	Length of a coded message.
r	$r = n - k$
w_c	Number of 1's in each column of a parity check matrix.
w_r	Number of 0's in each row of a parity check matrix.
$rate$	$rate = k/n$
m	The uncoded message bit sequence. $k = len(m)$
c	The coded message bit sequence. $n = len(c)$
H	The parity check matrix associated with the LDPC scheme. $[r,\ n] = size(H)$
G	The generator matrix associated with the parity check matrix. $[n,\ k] = size(G)$
$c = mG^T$	Generator equation used to encode a message.
$cH^T = 0$	Parity check equation to be used in decoding algorithm.

Table 3.1: *Terms related to LDPC coding*

Tanner Graphs

Besides the general expression as an algebraic matrix, LDPC codes can also be represented by a bipartite Tanner graph, which was proposed by Tanner in 1981. The Tanner graph consists of two sets of vertices: n vertices for the codeword bits (also called variable nodes), and k vertices for the parity-check equations (called check nodes). An edge joins a variable node to a check node if that bit is included in the corresponding parity-check equation and so the number of edges in the Tanner graph is equal to the number of ones in the parity-check matrix.

To demonstrate a Tanner graph, we use a parity check matrix as given below.

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

Then the Tanner graph for the above parity check matrix is as shown below.

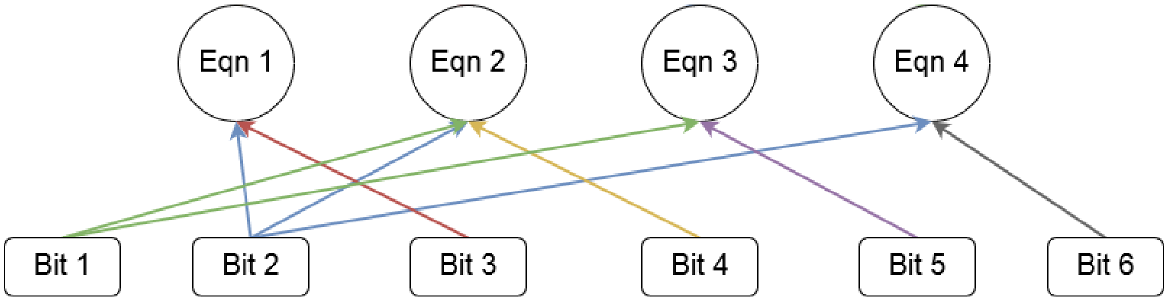


Fig 3.3: *Tanner graph for the H parity check matrix.*

The structure of parity check matrix and hence the Tanner graph is crucial for the performance of LDPC codes. LDPC codes use an iterative decoding algorithm based on the statistical independence of message transitions between the different nodes. When there exists a cycle, a message generated from one node will be passed back to itself, thus negating the assumption of independence, so that the decoding accuracy is impacted.

Cycle: A cycle (or loop) in a Tanner graph is a sequence of connected vertices which starts and ends at the same vertex in the graph, and which contains other vertices no more than once. The length of a cycle is the number of edges it contains. Since Tanner graphs are bipartite, every cycle is of even length. As discussed above, cycle in Tanner graphs is directly related to the performance of the LDPC code. Hence it is necessary to ensure that the length of cycles as high as possible.

Girth: It is the minimum length of all the cycles present in the given tanner graph. The girth of a given parity check matrix is a crucial measure of its performance.

Regular and Irregular LDPC codes

A regular LDPC code is the one that has an equal number of 1's in all the columns (i.e. w_c is constant) and an equal number of 1's in all the rows (i.e. w_r is constant), in its parity check matrix. These kind of LDPC codes are computationally less complex than irregular codes, which do not abide by the rule mentioned above. However, irregular LDPC codes perform better in error correction capabilities than regular LDPC codes.

Generation of LDPC codes

The most obvious method for the construction of LDPC codes is via constructing a parity-check matrix with the properties described in the previous chapter. Based on the design criteria, an LDPC code can be constructed to implement efficient encoding and decoding, in order to obtain near-capacity performance.

Several methods exist for the construction of LDPC codes, and two of them are discussed below:

Random construction: This is suited for LDPC codes with large values of n and k, and it leads to formation of irregular matrices. These codes have been shown to approach the theoretical capacity limits for the Additive White Gaussian Noise. These codes will outperform any algebraically constructed code, but their implementation is complex.

An example of randomly constructed LDPC matrix with n=20; r=15; w_r=4 is given below

$$\begin{matrix}
0&1&0&0&1&0&0&0&0&1&0&0&0&0&0&0&1&0&0&0\\
0&0&0&0&0&0&0&1&1&0&0&0&0&0&1&0&0&1&0&0\\
0&1&0&1&0&0&0&0&0&0&0&0&1&0&0&0&1&0&0&0\\
0&1&0&0&0&0&0&0&0&0&0&1&0&0&1&0&0&0&1&0\\
0&0&1&0&0&1&0&0&0&0&0&0&0&1&0&1&0&0&0&0\\
0&0&0&0&0&0&0&0&1&1&0&0&0&0&1&0&0&0&1&0\\
1&0&0&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&0&0\\
0&0&0&0&1&0&1&0&0&0&0&0&1&0&0&1&0&0&0&0\\
0&0&0&1&0&0&0&1&0&0&1&0&0&0&1&0&0&0&0&0\\
0&0&0&1&1&1&0&0&0&0&0&0&0&0&1&0&0&0&0&0\\
0&1&0&1&0&1&0&0&0&0&0&0&0&0&0&0&0&0&1&0\\
0&0&1&0&0&0&0&0&0&0&1&0&1&0&0&0&1&0&0&0\\
1&0&0&0&1&0&0&0&0&0&0&0&0&0&1&0&0&1&0&0\\
0&0&0&0&0&0&0&1&1&0&0&1&0&0&1&0&0&0&0&0\\
0&0&1&0&0&0&0&0&0&0&0&0&0&1&0&0&0&1&1&0
\end{matrix}$$

Gallager construction: Gallager construction is used to generate H matrices of medium and small size. The method always results to construction of a regular LDPC code. Complete method of Gallager construction is discussed below:

We know that H is an $r \times n$ matrix with each column having w_c number of 1's and each row having w_r number of 1's in it. Now we write H as followed:

$$H = \begin{bmatrix} H_1 \\ H_2 \\ H_3 \\ \vdots \\ H_{w_c} \end{bmatrix}$$

$$where\ size(H_i) = \left[{}^{r}/_{w_c}\ n\right]\ and\ H_i\ contains\ single\ '1'\ in\ each\ column\ and\ w_r\ 1's\ in\ each\ row.$$

H_1 also abides by the rule mentioned above, and it contains all 1’s in row i from column number (i-1).w_r to i.w_r. All other H_i matrices are column permutation of the H_1.

An example of of H matrix generated using the Gallager method for n=20; r=15; w_c=3 and w_r=4 is given below.

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

Generator matrix from Parity Check Matrix:

The generator matrix can be obtained from the given parity check matrix. The method for obtaining generator matrix is derived below.

$$c = mG^T\ and$$

$$cH^T = 0$$

$$implies\ mG^TH^T = 0\ for\ all\ possible\ message\ m.$$

$$which\ implies\ G^TH^T = 0. or\ HG = 0$$

Now H can be written as $[H_1|H_2];\ where\ H_1 is\ an\ r \times r\ invertible\ matrix$

$$and\ H_2\ is\ a\ r * k\ matrix.$$

Similarly, G can be written as $\begin{bmatrix} G_1 \\ G_2 \end{bmatrix}$; $where\ G_1 is\ an\ r \times k\ matrix$

$$and\ G_2\ is\ a\ k * k\ matrix.$$

$$Now, HG = 0\ imlies\ H_1G_1 + H_2G_2 = 0$$

We can arrive at a generator matrix for a given parity check matrix easily if we make the G_2 matrix an identity matrix. In that case, we have

$$G_2 = I, and\ hence$$

$$H_1G_1 + H_2 = 0$$

$$H_1G_1 = H_2$$

$The\ above\ equation\ holds\ true\ because\ H\ and\ G\ belong\ to\ GF(2)\ domain\ where -x = x.$

$$G_1 = H_1^{-1}H_2$$

So that the G matrix is equal to $G = \begin{bmatrix} G_1 \\ G_2 \end{bmatrix} = \begin{bmatrix} H_1^{-1}H_2 \\ I \end{bmatrix}$. The assumption in deriving this equation is that H_1 is invertible which is generally take care of while developing the H matrices.

The G^T matrix produced using this method for the above mentioned random matrix is

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

Generator matrix from Quasi-Cyclic Parity Check Matrix:

A quasi-cyclic block-based parity check matrix contains entries lower than an integer Z (called block size) and greater than or equal to -1. In the quasi-cyclic parity check matrix, each of the entry corresponds to a right rotated identity matrix of dimension Z×Z. 0 corresponds to the identity matrix (i.e. identity matrix right rotated by 0 positions). Similarly, value of n corresponds to identity matrix right rotated by n positions.

For Z=3, the identity matrix is given by

$$I_0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

So, the block corresponding to a zero entry is I_0. And the blocks corresponding to n=1 and 2 are

$$I_1 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \text{ and } I_2 = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

It should also be noted that the -1 entry corresponds to a null matrix of size Z×Z. That is

$$I_{-1} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

For example, the parity check matrix of a quasi-cyclic code with Z=3 is given by

$$H = \begin{bmatrix} -1 & 1 & 0 \\ 0 & 2 & 1 \end{bmatrix}$$

Then it should be effectively translated into

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

Which is a 6*9 matrix.

With the use of a particular type of quasi-cyclic codes (as explained below) the need of explicit calculation of G matrix is diminished. Exploiting the structure of H matrix and the quasi-cyclic properties, it becomes possible to generate the parity bits.

LDPC encoder can be designed using the lower triangular matrix approximation, where the parity check matrix is converted into a lower triangular form by row and column permutations without changing its linearity. To convert into a lower triangular form, the parity check matrix has been divided into sub matrices A, B, C, D, T, E has a certain order as shown below.

$$H = \begin{bmatrix} A & B & T \\ C & D & E \end{bmatrix}$$

Dimensions of the submatrices are as given below

$Entity$	**Dimension**
A	(r-g)×(n-r)
B	g×(n-r)
C	(r-g)×g
D	g×g
T	(r-g)×(r-g)
E	g×(r-g)

Table 3.2: *Terms related to quasi-cyclic LDPC encoding*

The encoding complexity is O (n+g^2), g is the row of matrix E. Figure 3.1 shows a standard lower triangular form of Parity Check Matrix obtained by performing certain column switching operations. For the 802.11n standards the value of g is kept as Z which is a constant. This reduced the complexity of encoder to O (n).

Description of the sub matrices:

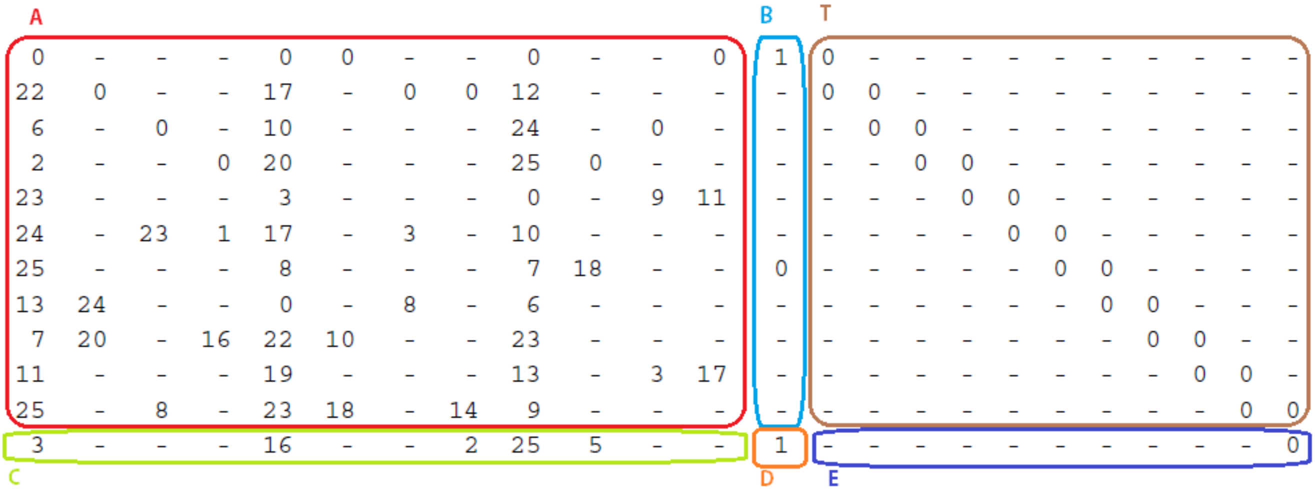
The A and C are sparse matrices with no particular structure. In-fact the lack of pattern in A and C helps in improving the performance of LDPC code.

The B matrix is a column vector (when represented in the condensed form). It should be noted that only two elements on this column vector are not -1. The first element is always non -1 entry and then a randomly chosen row has non -1 entry.

The D matrix is just one entry in condensed form and it is always non -1 entry.

The T matrix has all the elements as -1 except for the diagonal and the entries below the diagonal. All non-zero entries are 0 in the T matrix. This structure of T matrix is used during encoding.

The E matrix is a row vector in condensed form and all its elements are -1 except the last entry which is 0.



```
 0   -   -   -   0   0   -   -   0   -   -   0    1   0  -  -  -  -  -  -  -  -  -  -
22   0   -   -  17   -   0   0  12   -   -   -    -   0  0  -  -  -  -  -  -  -  -  -
 6   -   0   -  10   -   -   -  24   -   0   -    -   -  0  0  -  -  -  -  -  -  -  -
 2   -   -   0  20   -   -   -  25   0   -   -    -   -  -  0  0  -  -  -  -  -  -  -
23   -   -   -   3   -   -   -   0   -   9  11    -   -  -  -  0  0  -  -  -  -  -  -
24   -  23   1  17   -   3   -  10   -   -   -    -   -  -  -  -  0  0  -  -  -  -  -
25   -   -   -   8   -   -   -   7  18   -   -    0   -  -  -  -  -  0  0  -  -  -  -
13  24   -   -   0   -   8   -   6   -   -   -    -   -  -  -  -  -  -  0  0  -  -  -
 7  20   -  16  22  10   -   -  23   -   -   -    -   -  -  -  -  -  -  -  0  0  -  -
11   -   -   -  19   -   -   -  13   -   3  17    -   -  -  -  -  -  -  -  -  0  0  -
25   -   8   -  23  18   -  14   9   -   -   -    -   -  -  -  -  -  -  -  -  -  0  0
 3   -   -   -  16   -   -   2  25   5   -   -    1   -  -  -  -  -  -  -  -  -  -  0
```

Fig 3.4: *Submatrices A, B, C, D, T and E obtained from H matrix.*

Encoding Process:

The codeword generated after encoding is represented by *x*, then *x* can be written as

$$x^T = \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ \cdot \\ \cdot \\ m_k \\ p_1 \\ \cdot \\ p_r \end{bmatrix}$$

Here, m_1 to m_k are the message bits arranged in a Z×1 column vector fashion. p_1 to p_k are similarly the parity checks to be calculated.

Also, since *x* is the encoded output we have

$$Hx^T = 0$$

This results in the following set of equations. Assuming b^{th} row has the other non -1 element in the B matrix.

$$\sum_{j=1}^{k} h_{1,j} m_j + \ h_{1,k+1} p_1 + p_2 = 0$$

$$\sum_{j=1}^{k} h_{2,j}m_j + p_2 + p_3 = 0$$

$$\sum_{j=1}^{k} h_{3,j}m_j + p_3 + p_4 = 0$$

$$\cdots$$

$$\cdots$$

$$\sum_{j=1}^{k} h_{b,j}m_j + h_{b,k+1}p_1 + p_b + p_{b+1} = 0$$

$$\sum_{j=1}^{k} h_{b+1,j}m_j + p_{b+1} + p_{b+2} = 0$$

$$\sum_{j=1}^{k} h_{r,j}m_j + h_{r,k+1}p_1 + p_r = 0$$

The parity check matrix is so formed that $h_{r,k+1} = h_{1,k+1}$. Adding all these equations we get

$$\sum_{i=1}^{r}\sum_{j=1}^{k} h_{i,j}m_j + h_{b,k+1}p_1 = 0$$

Furthermore for $h_{b,k+1} = 0$; the equation simply becomes

$$\sum_{i=1}^{r}\sum_{j=1}^{k} h_{i,j}m_j = p_1$$

Substituting this result in first and last equation, the values of all unknown *p's* are calculated.

CHAPTER 4

LDPC DECDING USING SUM PRODUCT ALGORITHM

The most crucial part of LDPC code is its decoding algorithm. Apart from developing the LDPC itself, Gallager also proposed a method near optimal decoding algorithm for it. This algorithm has been modified for various purposes and is popular in the form of Sum Product Algorithm, Belief Propagation Algorithm and Message Passing Algorithm.

The Sum Product Algorithm works in the domain of Probability by using the maximum a posteriori or MAP directly. It is also possible to use the Sum Product Algorithm in log likelihood domain. However, we will keep our discussion to Probability domain.

Much like optimal (maximum *a posteriori*, MAP) symbol by symbol decoding of trellis code we are interested in computing the a posteriori probability (APP) of a given bit in the transmitted codeword c= [c_0 c_1 c_2 c_3….c_{n-1}] equals ‘1’, given the received word y= [y_0 y_1 y_2 …. y_{n-1}]. Without loss of generality, let us focus on the decoding of bit c_i so that we are interested in computing the APP. Then we have

$$b_i = \Pr(c_i = 1|y)$$

$\boldsymbol{Entity}$	**Description**
V_i	The probability of ith bit node being 1.
$\overline{V_i}$	$\overline{V_i} = 1 - V_i$
C_i	The probability of ith check node’s parity being satisfied.
q_{ij}	Message from the ith bit node to the jth check node. $-1 \leq q_{ij} \leq 1$
r_{ij}	Message from the ith check node to the jth bit node. $0 \leq r_{ij} \leq 1$
$\overline{r_{ij}}$	$\overline{r_{ij}} = 1 - r_{ij}$
b_i	*A posteriori* probability of the signal received from the channel.
o_i	The output bit corresponding to decoded message. $o_i \in \{0,1\}$

Table 4.1: *Terms related to SPA decoding*

The sum product algorithm can be best understood with the help of Tanner graph which is extended to show iteration. Each vertex of the Tanner graph (bit and check) correspond to a function that acts on all the input edges and produced a value assigned to that node. Another function then acts on the node value and returns the output to one of the output edges. It must be noted that some complex functions have be broken into simpler functions for the ease of understanding.

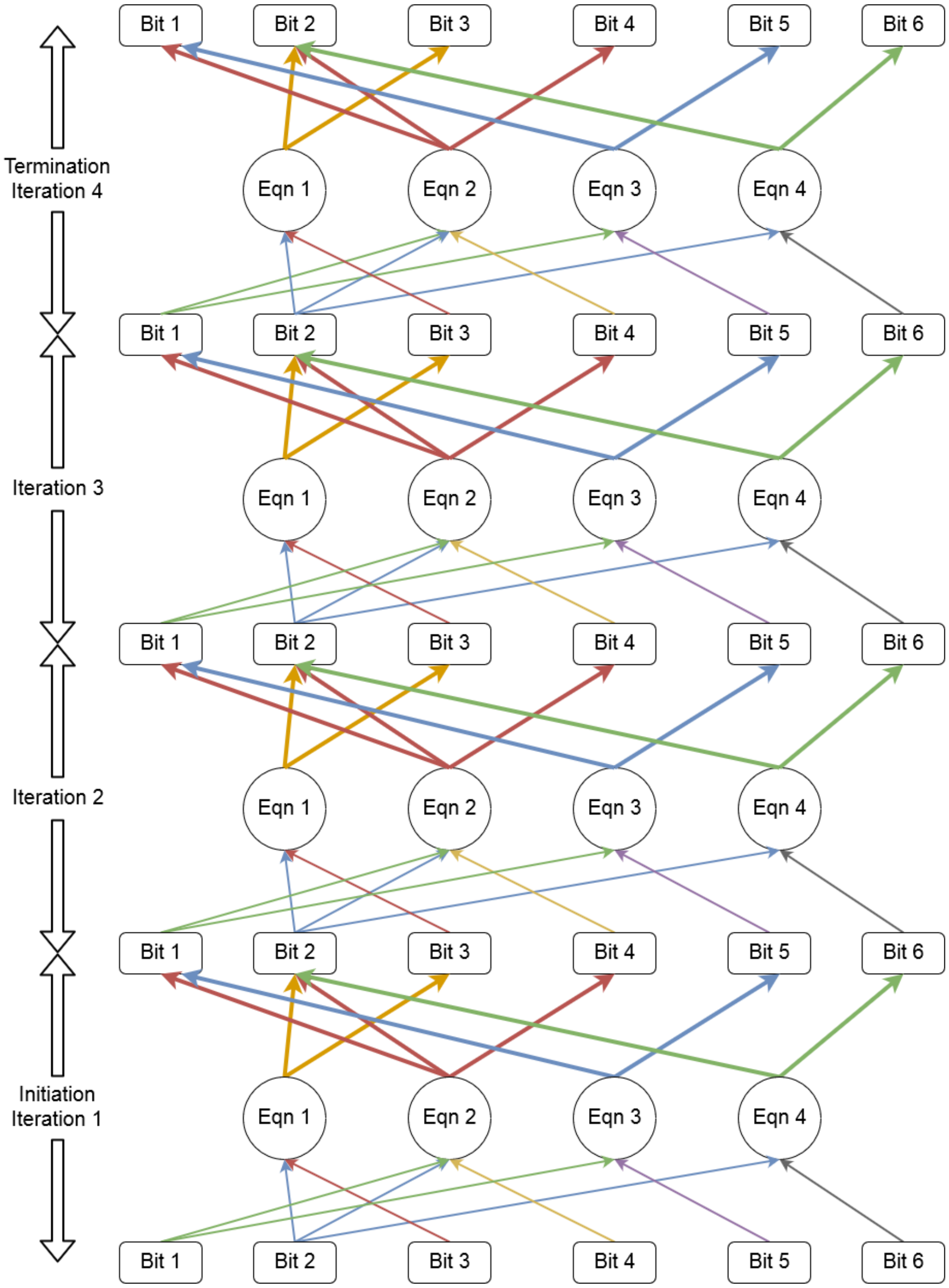


***Fig 4.1:** Extended Tanner graph showing the flow of SPA.*

In the above figure, the rectangular nodes represent the bit node; circular nodes represent check node; thin edges represent a message from bit node to check node or $q_{ij,}$ and thick edges represent a message from a check node to bit node or r_{ij}.

Initiation:

In the first iteration of the algorithm, all the bit nodes are given the APP values received from the channel. That is

$$V_i = b_i$$

We introduce a temporary variable t_{ij} corresponding to variable q_{ij} that keeps the unnormalized value of q_{ij}. For the first iteration only, we have

$$t_{ij} = V_i$$

Iteration:

All the steps given below are repeated iteratively. The APP probability received in the last step is passed on from the bit to check edge applying the function

$$q_{ij} = 1 - 2 \times t_{ij}$$

After this, we calculate check node as given below

$$C_j = \prod_i q_{ij}$$

Using the values of check node and input edge of the check node we calculate the value of edge from ith check node to jth bit node.

$$r_{ij} = \frac{1}{2}\left\{1 - \frac{C_j}{q_{ji}}\right\} \, and$$
$$\overline{r_{ij}} = 1 - r_{ij}$$

The ‘r_{ij}’ values are used to calculate the value of bit nodes V_j using the equation given below.

$$V_j = \, b_j \times \prod_i r_{ij} \; and$$
$$\overline{V_j} = \, (1 - b_j) \times \prod_i \overline{r_{ij}}$$

At this point, we introduce another temporary variable s_{ij} and $\overline{s_{ij}}$ which also corresponds to t_{ij} and hence q_{ij}.

$$s_{ij} = \frac{V_i}{r_{ji}} and$$

$$\overline{s_{ij}} = \frac{\overline{V_i}}{r_{ji}}$$

And finally, we have the normalisation equation

$$t_{ij} = \frac{s_{ij}}{s_{ij} + \overline{s_{ij}}}$$

The above set of equations are repeated in an iterative function until termination.

Termination:

The algorithm uses hard decision on the value of V_i to stop the algorithm as soon as the equation

$$cH^T = 0$$

is satisfied. The algorithm is also fed with a maximum iteration value that it should run and should stop even if the parity check equation is not satisfied. This is done to ensure that the algorithm doesn't enter an infinite loop. The equation for hard decision as given below

$$o_i = \begin{matrix} 0 \; if \; V_i < 0.5 \\ 1 \; if \; V_i > 0.5 \end{matrix}$$

In figure 4.2 the SPA algorithm is shown graphically. It must be noted that the algorithm flows bottom-up.

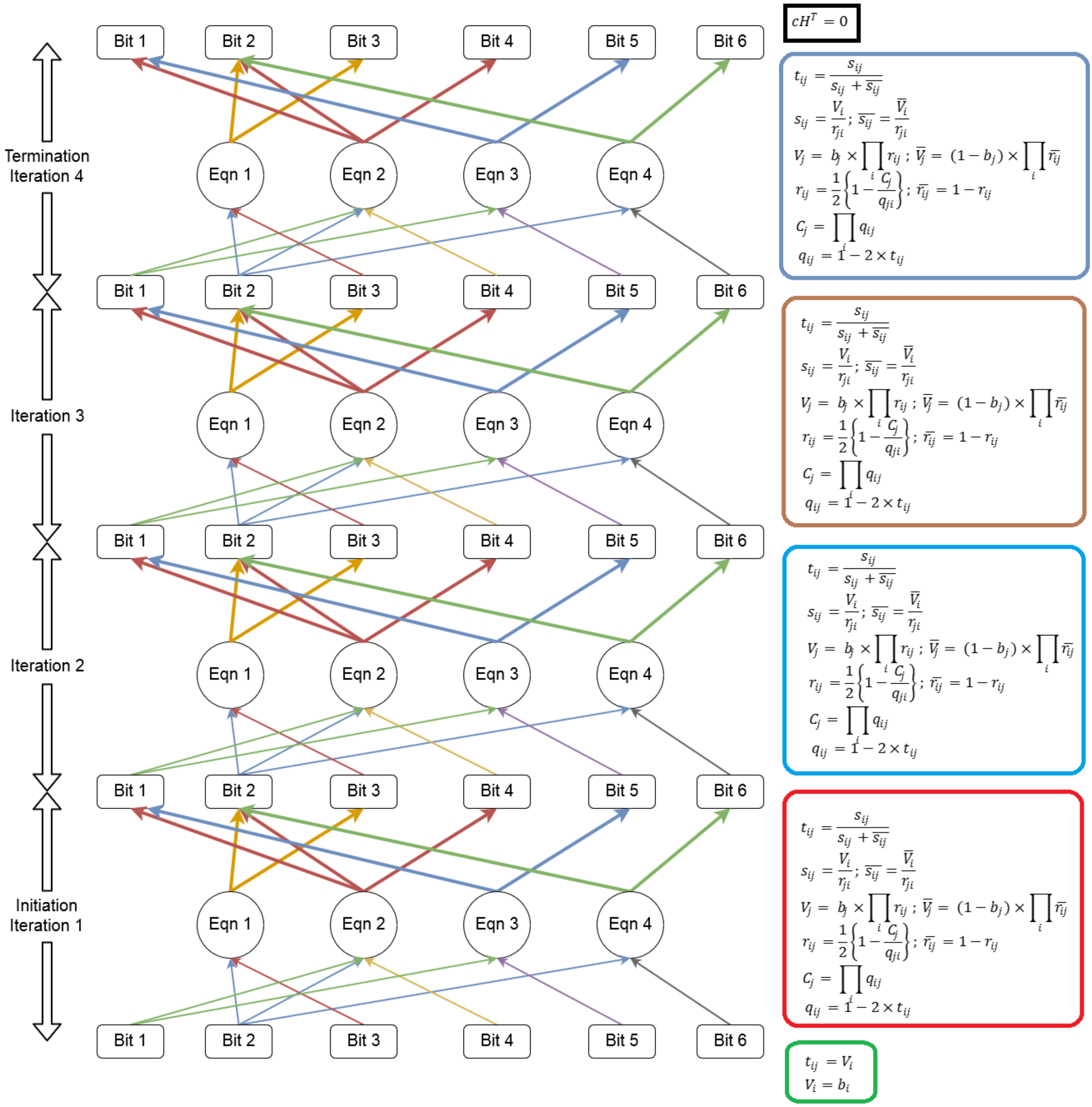

Bit 1
Bit 2
Bit 3
Bit 4
Bit 5
Bit 6
Eqn 1
Eqn 2
Eqn 3
Eqn 4
Termination
Iteration 4
Iteration 3
Iteration 2
Initiation
Iteration 1
$cH^T = 0$
$t_{ij} = \frac{s_{ij}}{s_{ij} + \overline{s_{ij}}}$
$s_{ij} = \frac{V_i}{r_{ji}}; \overline{s_{ij}} = \frac{\overline{V_i}}{r_{ji}}$
$V_j = b_j \times \prod_i r_{ij}; \overline{V_j} = (1 - b_j) \times \prod_i \overline{r_{ij}}$
$r_{ij} = \frac{1}{2}\left\{1 - \frac{C_j}{q_{ji}}\right\}; \overline{r_{ij}} = 1 - r_{ij}$
$C_j = \prod_i q_{ij}$
$q_{ij} = 1 - 2 \times t_{ij}$
$t_{ij} = V_i$
$V_i = b_i$

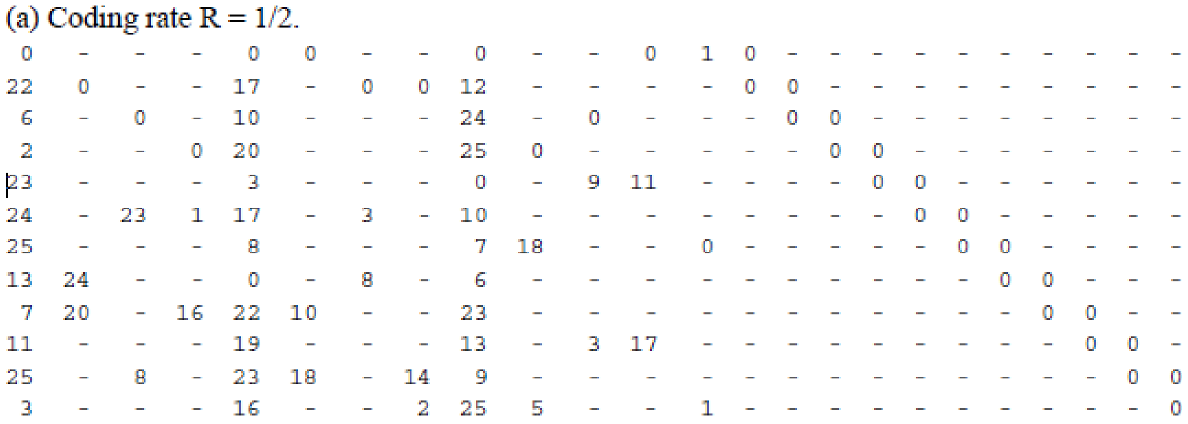
Fig 4.2: *Extended Tanner graph showing the flow of SPA.*

Performance of 802.11 specified matrices:

LDPC was introduced to IEEE 802.11 standards for the first time in its Amendment 5 that is 802.11n-2009 standards. As it has been discussed earlier, LDPC is a near-capacity coding scheme. Hence, with the requirement of increased data rate, LDPC turned out to be the perfect solution.

By 802.11n standards, it supports four rates of encoding – $^{1}/_{2}, ^{2}/_{3}, ^{3}/_{4},$ $and\ ^{5}/_{6}$. The values of ‘n’ or codeword length are – 648, 1296, and 1944. This satisfies a wide range of requirement desired by any product or service. Despite all its advantages, LDPC is not mandatory for 802.11 product and a legacy support for devices working on older version has been established.

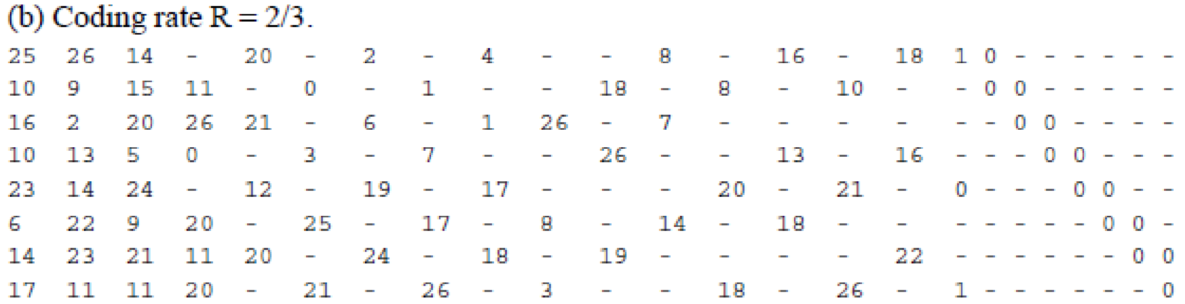
The LDPC parity check matrices given in IEEE 802.11 are as given below



(a) Coding rate R = 1/2.

0	-	-	-	0	0	-	-	0	-	-	0	1	0	-	-	-	-	-	-	-	-	-	-
22	0	-	-	17	-	0	0	12	-	-	-	-	0	0	-	-	-	-	-	-	-	-	-
6	-	0	-	10	-	-	-	24	-	0	-	-	-	0	0	-	-	-	-	-	-	-	-
2	-	-	0	20	-	-	-	25	0	-	-	-	-	-	0	0	-	-	-	-	-	-	-
23	-	-	-	3	-	-	-	0	-	9	11	-	-	-	-	0	0	-	-	-	-	-	-
24	-	23	1	17	-	3	-	10	-	-	-	-	-	-	-	-	0	0	-	-	-	-	-
25	-	-	-	8	-	-	-	7	18	-	-	0	-	-	-	-	-	0	0	-	-	-	-
13	24	-	-	0	-	8	-	6	-	-	-	-	-	-	-	-	-	-	0	0	-	-	-
7	20	-	16	22	10	-	-	23	-	-	-	-	-	-	-	-	-	-	-	0	0	-	-
11	-	-	-	19	-	-	-	13	-	3	17	-	-	-	-	-	-	-	-	-	0	0	-
25	-	8	-	23	18	-	14	9	-	-	-	-	-	-	-	-	-	-	-	-	-	0	0
3	-	-	-	16	-	-	2	25	5	-	-	1	-	-	-	-	-	-	-	-	-	-	0

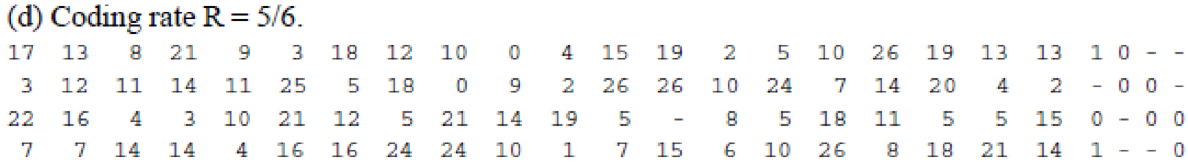
Fig 4.3: *rate = ½; n=648*



(b) Coding rate R = 2/3.

25	26	14	-	20	-	2	-	4	-	-	8	-	16	-	18	1	0	-	-	-	-	-	-
10	9	15	11	-	0	-	1	-	-	18	-	8	-	10	-	-	0	0	-	-	-	-	-
16	2	20	26	21	-	6	-	1	26	-	7	-	-	-	-	-	-	0	0	-	-	-	-
10	13	5	0	-	3	-	7	-	-	26	-	-	13	-	16	-	-	-	0	0	-	-	-
23	14	24	-	12	-	19	-	17	-	-	-	20	-	21	-	0	-	-	-	0	0	-	-
6	22	9	20	-	25	-	17	-	8	-	14	-	18	-	-	-	-	-	-	-	0	0	-
14	23	21	11	20	-	24	-	18	-	19	-	-	-	-	22	-	-	-	-	-	-	0	0
17	11	11	20	-	21	-	26	-	3	-	-	18	-	26	-	1	-	-	-	-	-	-	0

Fig 4.4: *rate = 2/3; n=648*



(d) Coding rate R = 5/6.

17	13	8	21	9	3	18	12	10	0	4	15	19	2	5	10	26	19	13	13	1	0	-	-
3	12	11	14	11	25	5	18	0	9	2	26	26	10	24	7	14	20	4	2	-	0	0	-
22	16	4	3	10	21	12	5	21	14	19	5	-	8	5	18	11	5	5	15	0	-	0	0
7	7	14	14	4	16	16	24	24	10	1	7	15	6	10	26	8	18	21	14	1	-	-	0

Fig 4.5: *rate = 5/6; n=648*

All the hyphens in the matrix given above are to be replaced with zero matrices of size 27 × 27. A numerical entry of any number I in the matrix above refers to a 27 × 27 identity matrix with its columns being right-circular shifted by i. So, 0 represents an identity matrix while 1 represents a matrix as given below

$$\begin{matrix} 0 & 1 & 0 & . & . & 0 \\ 0 & 0 & 1 & . & . & 0 \\ \vdots & \vdots & \vdots & 1 & \vdots & \vdots \\ 0 & 0 & . & . & . & 1 \\ 1 & 0 & . & . & 0 & 0 \end{matrix}$$

Bit Error rate of LDPC codes

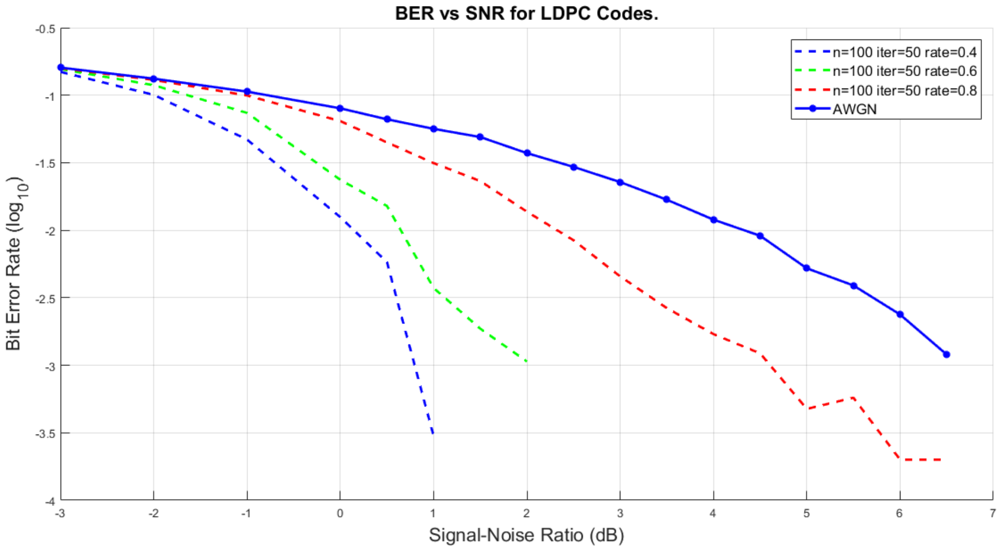


Fig 4.6: *BER vs SNR for BPSK in for randomly generated parity check matrix.*

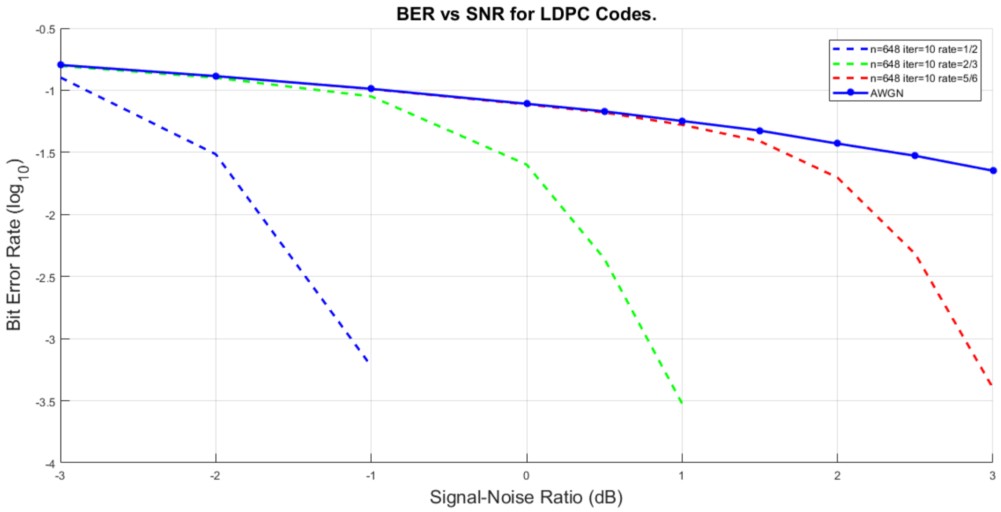


Fig 4.7: *BER vs SNR for BPSK in for 802.11n specified matrix.*

However, in LDPC it makes more sense to talk about Packet Error Rate than Bit Error Rate. Hence, the PER vs BER is found and plotted as shown below.

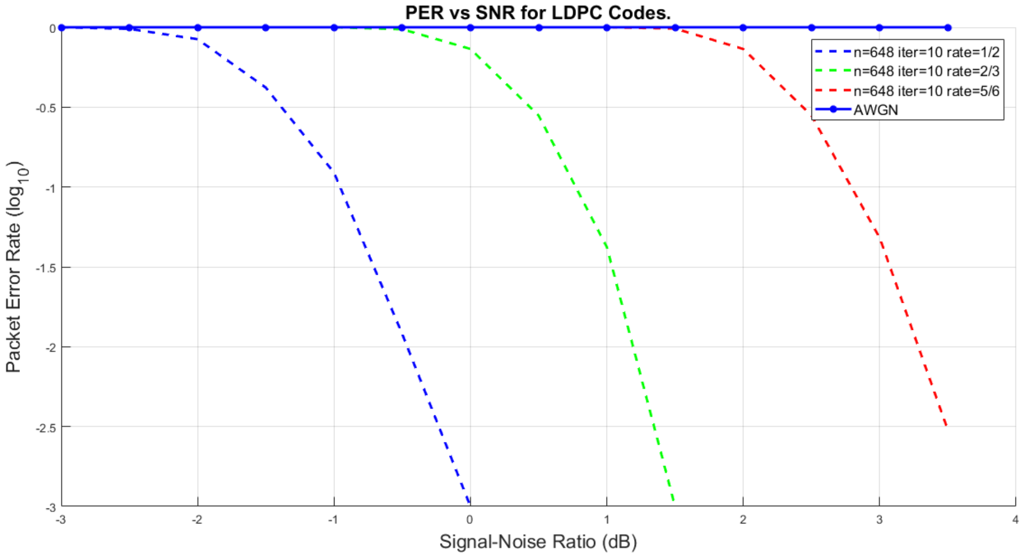


Fig 4.8: *PER vs SNR for BPSK in for 802.11n specified matrix.*

CHAPTER 5

MIMO CHANNEL ESTIMATION AND EQUALISATION

The HT-SIG field is followed by HT-STF that is used for Automatic Gain Control. After the gain control of the MIMO channel, it is essential to estimate the channel for each subcarrier of the OFDM. The channel estimation is achieved through the HT-DLFT (and for Sounding packets HT-ELTF too).

Channel Estimation

It is important to understand that the HT-DLFT undergoes transmitter 2 for its generation and the subcarriers of the HT-DLFT are multiplied with a P matrix (not to be confused with Precoder matrix), that is orthogonal in nature and hence helpful in channel estimation. It should be noted that the channel is measured between the output of STBC and the input of spatial demapper.

N_{STS}	$N_{HT\text{-}LTF}$
1	1
2	2
3	4
4	4

Table 5.1: *Number of LTF required for given number of space time streams.*

A $\hat{P}$ matrix is generated using the P matrix given in standards. To generate the $\hat{P}$ matrix, first N_{STS} rows and first $N_{LT\text{-}DLTF}$ columns of P matrix are chosen and others are discarded. The P matrix is given below. $\hat{P}$ can only take size $1 \times 1, 2 \times 2, 3 \times 4, and\ 4 \times 4$, according to the number of DLTF for given number of STS. The P matrix is given below:

$$p = \begin{bmatrix} 1 & -1 & 1 & 1 \\ 1 & 1 & -1 & 1 \\ 1 & 1 & 1 & -1 \\ -1 & 1 & 1 & 1 \end{bmatrix}$$

Now, we can write the input received signal as followed in a MIMO system:

$$Y = HX + n$$

And this will hold true for any subcarrier at any time. To estimate the channel, we need the signals from different time instances (i.e. from different DLTFs). We denote these time instances by t_1, t_2, t_3, … t_{NDLFT}. Accounting N_{DLFT} signals in the equation above, it becomes:

$$[Y_{t_1}|Y_{t_2}|Y_{t_3}|\ldots|Y_{t_{NDLTF}}] =$$

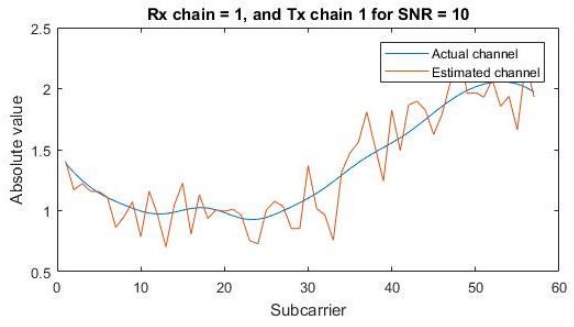
$$\begin{bmatrix} h_{11} & h_{12} & h_{13} & . & . & h_{1N_{STS}} \\ h_{21} & h_{22} & h_{23} & . & . & h_{2N_{STS}} \\ h_{31} & h_{32} & \vdots & \vdots & \vdots & \vdots \\ . & . & . & . & . & . \\ h_{N_{RX}1} & h_{N_{RX}2} & h_{N_{RX}3} & . & . & h_{N_{RX}N_{STS}} \end{bmatrix} .\hat{P}.HTLFT + [N_{t_1}|N_{t_2}|N_{t_3}|\ldots|N_{t_{NDLTF}}]$$

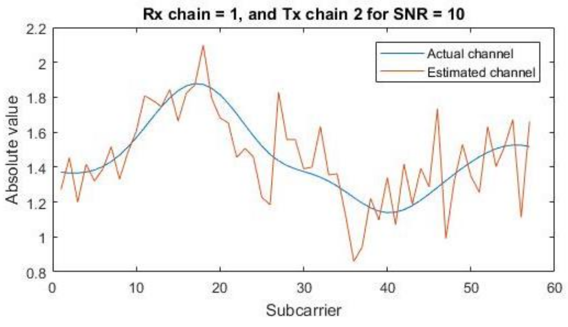
Using this equation, we can easily get an estimate of the channel as followed:

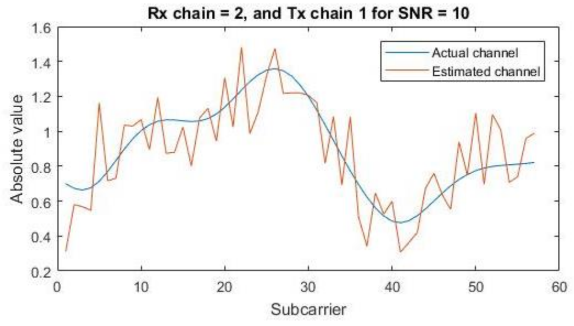
$$\begin{bmatrix} h_{11} & h_{12} & h_{13} & . & . & h_{1N_{STS}} \\ h_{21} & h_{22} & h_{23} & . & . & h_{2N_{STS}} \\ h_{31} & h_{32} & \vdots & \vdots & \vdots & \vdots \\ . & . & . & . & . & . \\ h_{N_{RX}1} & h_{N_{RX}2} & h_{N_{RX}3} & . & . & h_{N_{RX}N_{STS}} \end{bmatrix} = [Y_{t_1}|Y_{t_2}|Y_{t_3}|\ldots|Y_{t_{NDLTF}}] \,.\widehat{P^T}.\frac{1}{N_{LTF}.HTLFT}$$

This is a zero forcing estimation of the MIMO channel. Other algorithms using MMSE and Belief propogation can be used in place of zero forcing to get better performance.

Using the above set of equation, channel estimation was implemented on MATLAB and LABVIEW based platform. The channel estimate for TGn channel model D for each subcarrier is plotted below.







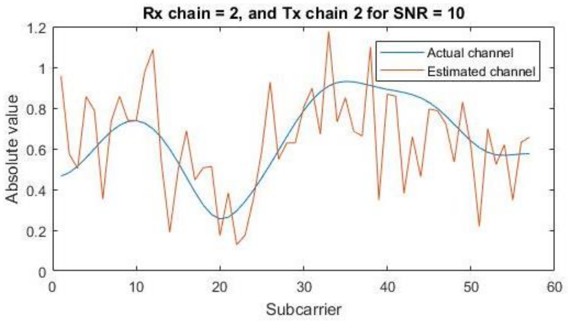


Fig 5.1: *Plot of MIMO channel estimation for SNR = 10.*

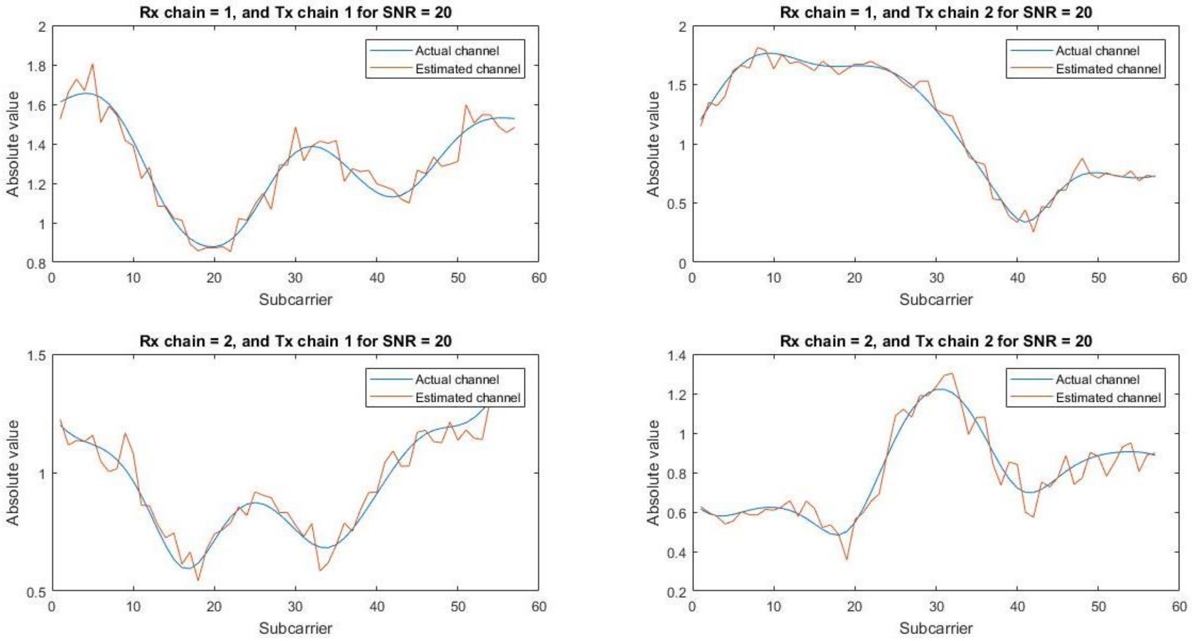


Fig 5.2: *Plot of MIMO channel estimation for SNR = 20.*

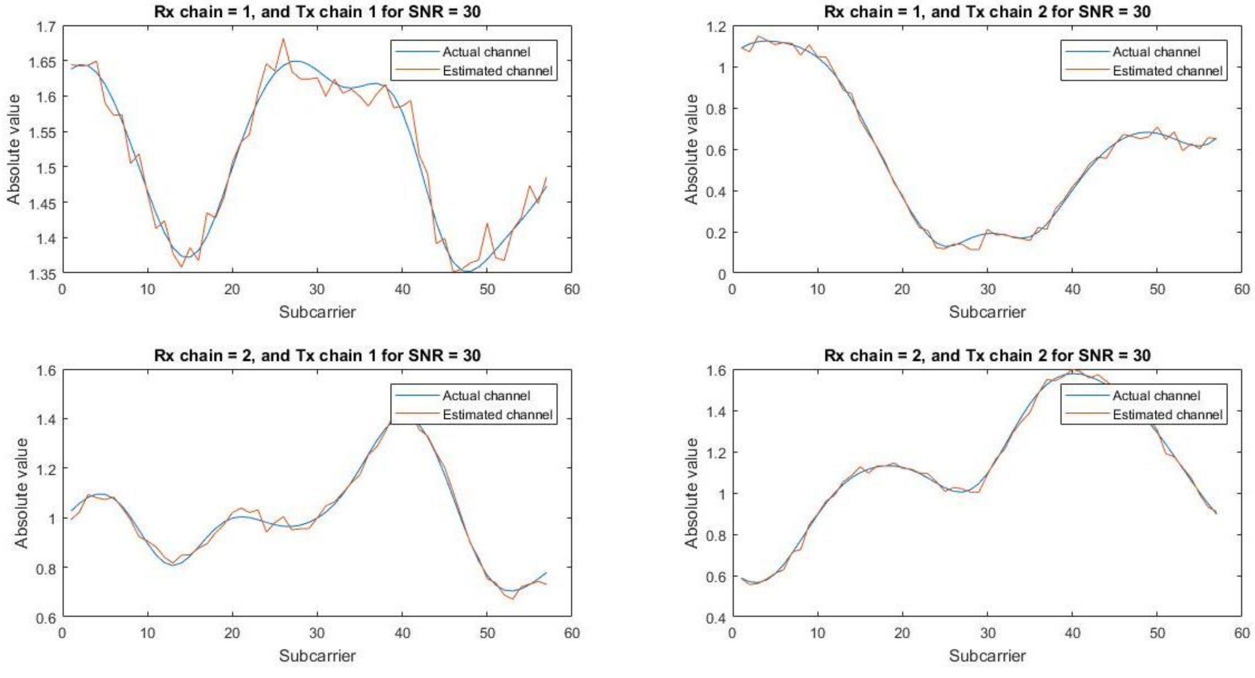


Fig 5.3: *Plot of MIMO channel estimation for SNR = 30.*

The plots mentioned above are generated using zero forcing algorithm for channel estimation. As it can be seen the channel estimation provides better performance at higher SNR. The channel model used for generation of these plots is TGn channel model D that has maximum delay of 350nS.

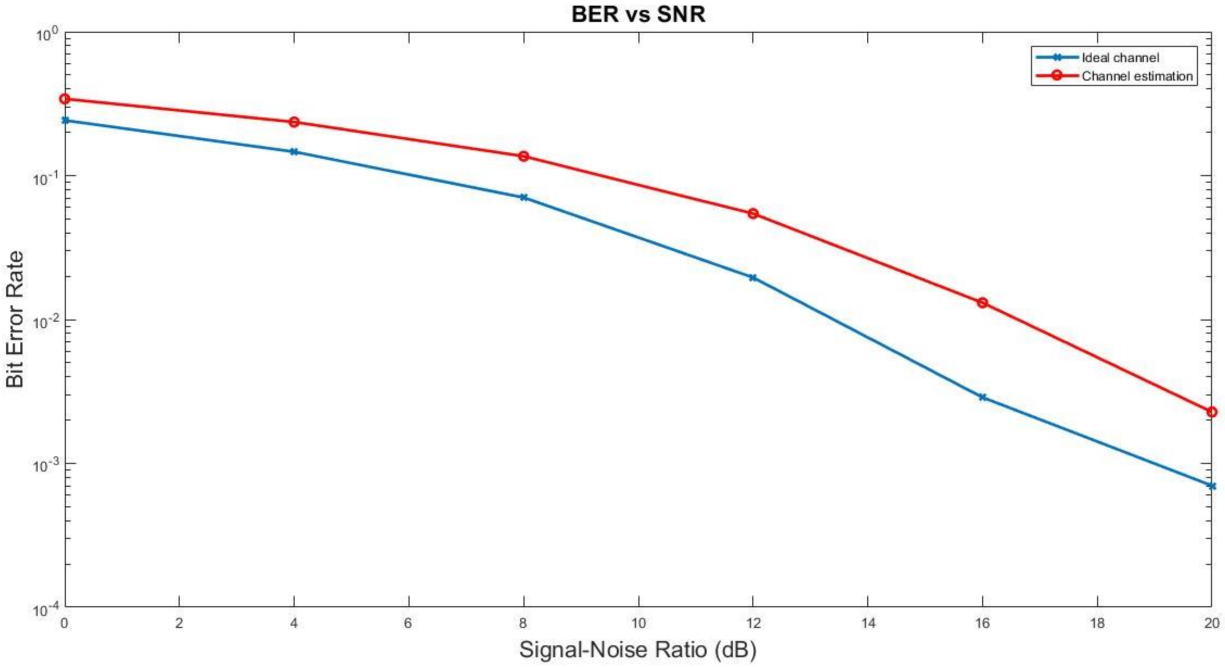


Fig 5.4: *Ideal channel estimates vs Channel estimation for two streams V-BLAST MIMO technique using QPSK.*

Channel Equalisation

Once the channel has been estimated, the received data needs to be equalised in order to get the constellation symbols. The 802.11n standard doesn't suggest any particular algorithm for channel estimation or equilisation and it is left to the developers to come up with suitable algorithms. In the simulator, we have used zero forcing algorithm for channel equalisation. However, better performing algorithms like MMSE or belief propogation based non-linear eqalisers can be used to enhance the performance.

The received signal is given by

$$Y = HX + n$$

And let the estimated channel be $\widehat{H}$.

Then the estimation of transmitted symbol vector X is given by

$$\hat{X} = (\widehat{H^H}\widehat{H})^{-1}\widehat{H}Y$$

Which implies

$$\hat{X} = (\widehat{H^H}\widehat{H})^{-1}\widehat{H}X + (\widehat{H^H}\widehat{H})^{-1}\widehat{H}n$$

When the channel power is low, the value of $(\widehat{H^H}\widehat{H})^{-1}$ is high and that leads to noise enhancement and thus worse performance. To avoid this, MMSE can be used. The zero forcing equalisers are however easy to implement and study.

Constellation diagrams of 2×2 V-BLAST is shown in the plots below for various signal to noise ratios. QPSK signals were transmitted for this simulation.

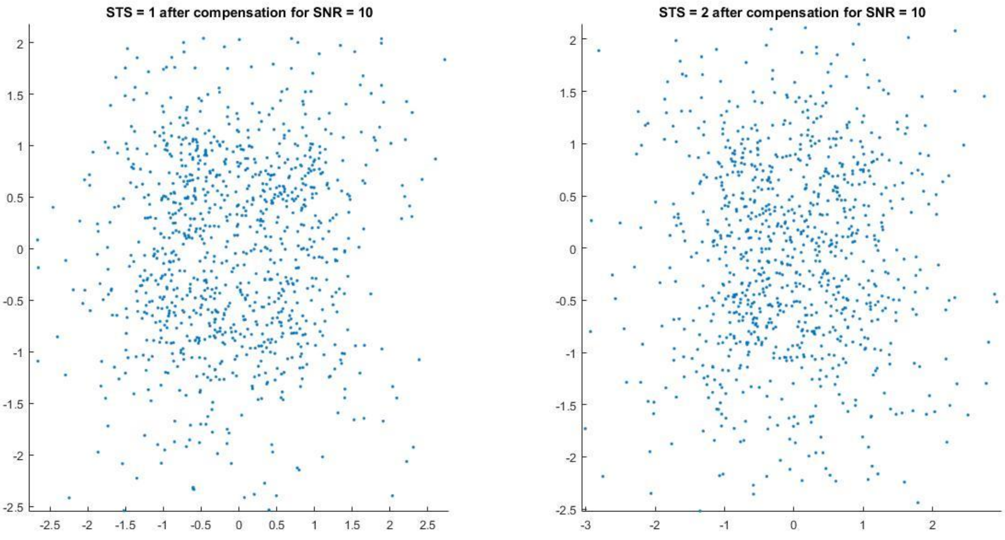


Fig 5.5: *Constellation diagram after channel equalisation for SNR = 10.*

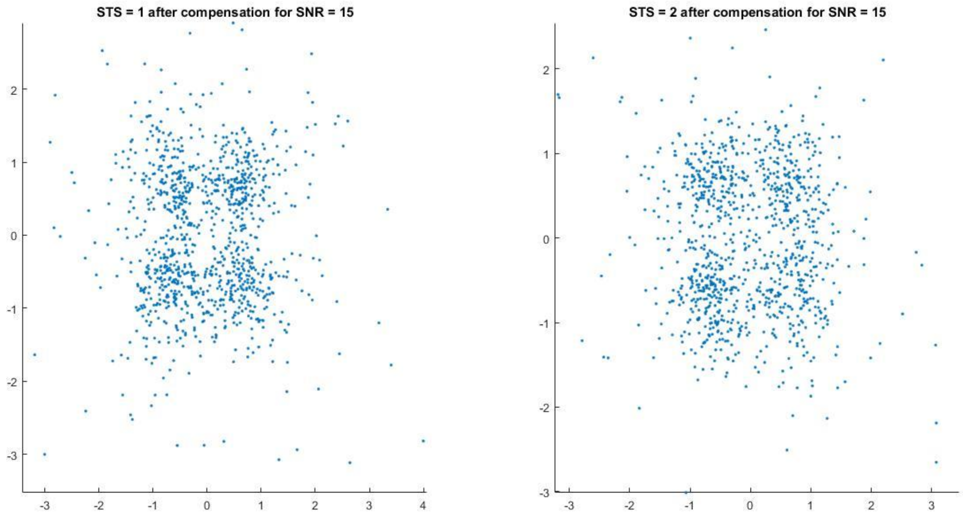


Fig 5.6: *Constellation diagram after channel equalisation for SNR = 15.*

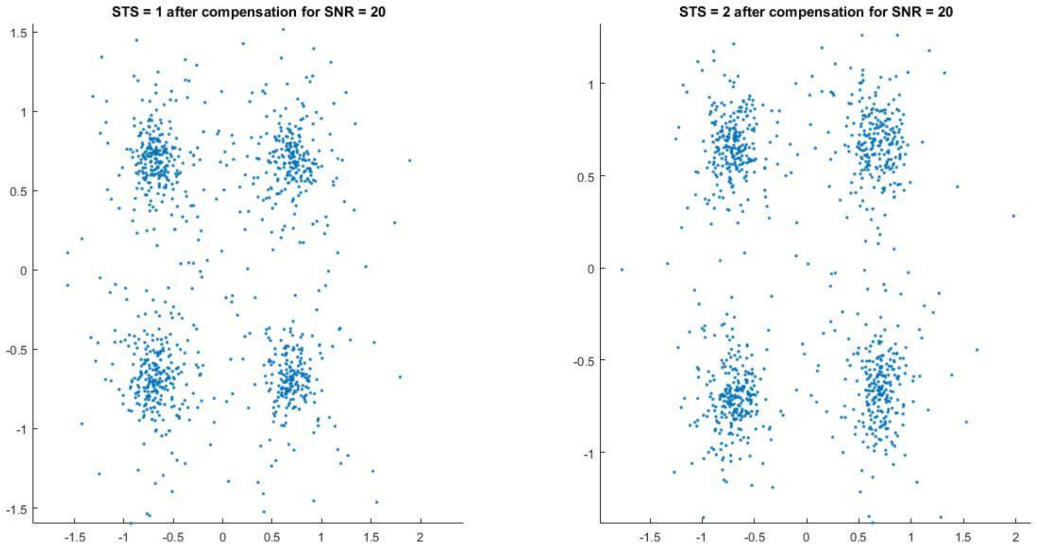


Fig 5.7: *Constellation diagram after channel equalisation for SNR = 20.*

CHAPTER 6

MIMO CHANNEL FEEDBACK

In a MIMO communication link, the transmitter often needs to know the channel state in order to make a wise decision of precoder matrix. However, the channel state may not always be available at the transmitter. In the absence of channel state feedback, either of the two methods is used:

Precoder matrix index

Instead of complete channel state feedback, a codebook may be agreed upon in advance, and just the precoder matrix index is sent back to the receiver. This method is actively used in LTE systems. This method always leads to the suboptimal solution.

Implicit feedback

The channel formed between devices are not reciprocal due to transmitter and receiver RF distortions/impairments. However, it is possible to make the MIMO channel reciprocal by calibration. Going by 802.11n standards, it is possible to use implicit feedback. Once the channel has been made reciprocal, the transmitter can use the downlink channel estimation to get an estimate of the uplink channel. Implicit feedback requires hardware to support calibration which is not yet supported by many Wi-Fi devices.

The channel State can be fed back to the user in three different ways. The art of sending back channel information is called Explicit feedback.

Explicit feedback

Non-Compressed Feedback: Instead of sending back the channel state information directly, the CSI is used at the receiver to get the best beamforming weights (usually using the SVD technique) and the weights and corresponding unitary 'V' matrix is fed back in its raw form. This method of feedback leads to decrease of efficiency as the feedback for 4×4 MIMO channel with 48 subcarriers can reach thousands of byte.

Compressed Feedback: The 'V' matrices can also be compressed to save bits during feedback. This is usually achieved with the channel state being transposed to polar coordinates and by the use of Givens matrices.

CSI: The method of complete Channel State Information feedback is seldom used due to heavy feedback associated with it, but it allows a huge amount of signal processing algorithms and proprietary algorithms to run on the transmitter. Due to these benefits of CSI, it has been thoroughly studied, and a method of CSI feedback proposed in 802.11n standards is presented here.

The CSI method is beneficial as it can be used by an Access Point to get CSI of all mobile station in an area and create a profile of propagation environment. Many cheap devices do not support computation of beamforming weights, and in that case, the CSI is the only feedback possible. Knowledge of CSI also helps the transmitter in link adaptation and Implicit Feedback.

One of the methods of CSI feedback is as shown below:

We start with the channel estimated by the receiver

$$h(k) = \begin{bmatrix} h^I_{11}(k) + h^Q_{11}(k) & h^I_{12}(k) + h^Q_{12}(k) & \cdot & h^I_{1N_{STS}}(k) + h^Q_{1N_{STS}}(k) \\ h^I_{21}(k) + h^Q_{21}(k) & h^I_{22}(k) + h^Q_{22}(k) & \cdot & h^I_{2N_{STS}}(k) + h^Q_{2N_{STS}}(k) \\ \cdot & \cdot & \cdot & \cdot \\ h^I_{N_{RX}1}(k) + h^Q_{N_{RX}1}(k) & h^I_{N_{RX}2}(k) + h^Q_{N_{RX}2}(k) & \cdot & h^I_{N_{RX}N_{STS}}(k) + h^Q_{N_{RX}N_{STS}}(k) \end{bmatrix}$$

Then we define m$_H$(k) as given below

$$m_H(k) = \max\left[max_{i,j}\left(h^I_{ij}\right), max_{i,j}\left(h^Q_{ij}\right)\right]$$

We define M and M$_H$(k) as

$$M = \max_k[m_H(k)]$$

$$M_H(k) = \min\{7, 20\log_{10}\frac{M}{m_H(k)}\}$$

Then, we have $m_H^{lin}(k)$ as

$$m_H^{lin}(k) = \left\{\begin{matrix} m_H(k); if\,\dfrac{M}{2.23} \leq m_H(k) \leq M \\ \dfrac{m_H(k)}{2.23}; otherwise \end{matrix}\right\}$$

Then in the feedback for each of the subcarrier ‘k’, $m_H(k)$ which can be encoded in 3 bits is sent. The actual value of reathe l and imaginary part is encoded using N$_b$ bits in 2’s complementary form and then transmitted. The expression for encoding real and imaginary part is given below:

$$\hat{h}_{ij}^{I,Q}(k) = round\left[\frac{h_{ij}^{I,Q}(k)}{m_H^{lin}(k)}(2^{N_b-1}-1)\right]$$

As mentioned before, this kind of compressed complete channel feedback information is necessary for much digital processing algorithm implementation and hence a crucial part of this study.

CHAPTER 7

CAPACITY ANALYSIS OF MIMO CODEBOOK

Capacity of a Channel:

Channel capacity is the tight upper bound on the rate at which information can be reliably transmitted over a communications channel. It is the highest information transmission that can be achieved with arbitrarily small error probability.

Information theory developed by Claude E. Shannon, defines the notion of channel capacity and provides a mathematical model by which one can compute it. The key result states that the capacity of the channel is given by the maximum of the mutual information between the input and output of the channel, where the maximization is with respect to the input distribution.

Channel Capacity of an AWGN Channel:

Let $h(X)$ denote the differential entropy of a random variable X, and $h(Y|X)$ denote the conditional differential entropy of random variable Y given the random variable X. Then the information between random variable $X\ and\ Y$ is given by

$$I(X,Y) = \sum_{x\in X}\sum_{y\in Y} p(x,y)\log\left(\frac{p(x,y)}{p(x)p(y)}\right)$$

$$= h(Y) - h(X|Y)$$
$$= h(X) - h(X|Y)$$

Also, the received signal from an AWGN channel is

$$y = x + n$$

Therefore, the capacity of an AWGN channel is:

$$\begin{aligned} C &= \max\{I(X,Y)\} \\ &= \max\{h(Y) - h(Y|X)\} \\ &= \max\{h(Y) - h(X+N|X) \\ &= \max\{h(Y) - h(N|X)\} \end{aligned}$$

Where, $h(Y)$ is the differential entropy of the received signal and it is maximized if and only if Y is a gaussian random variable. It should also be noted that the noise random variable N is independent of input signal X.

Now, the differential entropy of a gaussian random variable is given by:

$$h(Y) = \frac{1}{2}log_2(2\pi e\sigma^2)$$

Here σ^2 is the variance of the gaussian random variable Y. So, the channel capacity is:

$$C = \frac{1}{2}log_2\big(2\pi e(P + N_0)\big) - \frac{1}{2}log_2(2\pi eN_0)$$

$$= \tfrac{1}{2}log_2\left(1 + \tfrac{P}{N_0}\right)\ per\ channel\ use$$

Channel Capacity of a MIMO Link:

For a MIMO link the differential entropy of the received signal $Y = HX + N$ is given by

$$h(Y) \leq log_2[\det\{\pi eR_{YY}\}]$$

$$where\ R_{YY} = E[YY^H]$$
$$and\ Y^H\ is\ hermitian\ of\ Y.$$

And the differential entropy of N is given by

$$h(n) = log_2[\det\{\pi eR_{NN}\}]$$

$$where\ R_{NN} = E[NN^H] = N_0I$$

Then the capacity of a MIMO channel is given by

$$C = h(Y) - h(N|X) = h(Y) - h(N)$$

Now the received signal Y is re-written as $Y = \sqrt{\frac{E_s}{N_T}}HPX + N$; where H is the channel co-efficient matrix of a Rayleigh fading channel, P is a precoder matrix (discussed in Chapter 3), X is the energy normalised input signal and $\sqrt{\frac{E_s}{N_T}}$ is the energy transmitted per antenna per channel use. It should be noted that $R_{PXPX} = E[PX(PX)^H] = R_{PP} = E[PP^H]\ as\ R_{XX} = E[XX^H] = I.$

So, the value of R_{YY} turns out to be

$$R_{YY} = E[YY^H]$$

$$R_{YY} = E\left[\left(\sqrt{\frac{E_s}{N_T}}HPX + N\right)\left(\sqrt{\frac{E_s}{N_T}}HPX + N\right)^H\right]$$

$$= E[HPX(PX)^H H^H + \ NN^H]$$

$$= \ \frac{E_s}{N_T} HR_{PP}H^H + \ R_{NN}$$

So, the equation for channel capacity turn out to be

$$C = h(Y) - h(n)$$

$$= \ log_2\left[\det\left\{\pi e N_0\left(\frac{E_s}{N_T}HR_{PP}H^H + \ R_{NN}\right)\right\}\right] - \ log_2[\det\{\pi e N_0 I\}]$$

$$= \ \boldsymbol{log_2}\left[\mathbf{det}\left\{\frac{\boldsymbol{E_s}}{\boldsymbol{N_T N_0}}\boldsymbol{HR_{PP}H^H} + \ \boldsymbol{I}\right\}\right]$$

The channel capacity for MIMO link in Rayleigh fading channel shall be used in the next chapter to determine the channel capacity for various predefined Precoder matrices as defined in LTE standards.

Precoder Matrix:

The knowledge of complete CSI at the transmitter can be inefficient for the system. Instead of sending the complete CSI, it is in favour of efficiency to calculate the best weightage that transmitter needs to assign to transmit antenna, and feedback the weightage to transmitter.

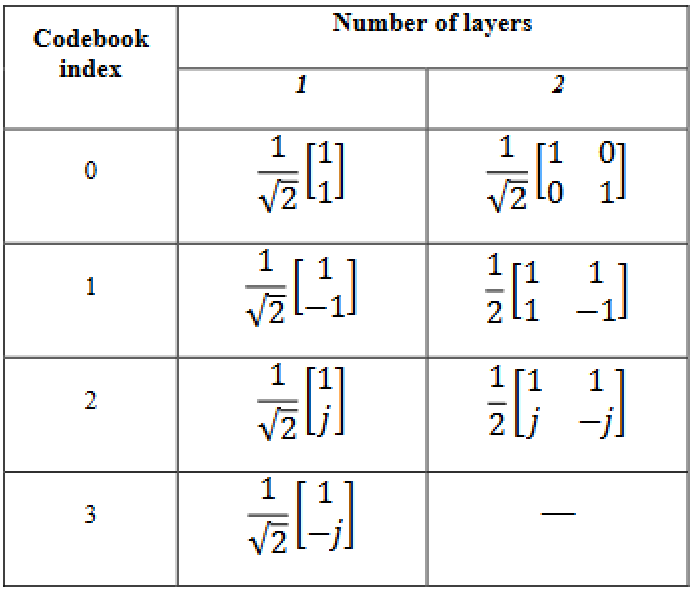
A Predefined set of matrices corresponding to the number of antennae deployed in the system is stored beforehand on both transmitter and the receiver. This set of matrices is called **Codebook** and the individual matrices are called **Precoder Matrix.** The number of precoder matrices in a codebook is dependent on the number of transmit antenna used. As soon as the receiver receives the CSI, it computes the best precoder matrix for the received CSI. The receiver then needs to feedback only the precoder matrix index to the transmitter and not the complete CSI.

This method saves a lot of bandwidth in the control channel and is an efficient solution to the CSI feedback mechanism. The precoder matrix method for MIMO has been standardised in LTE.

LTE Codebook:

LTE used Precoder based MIMO configuration. The LTE Release 8 defines seven different transmission modes that uses the precoder matrix. These matrices are given below in a tabular form:

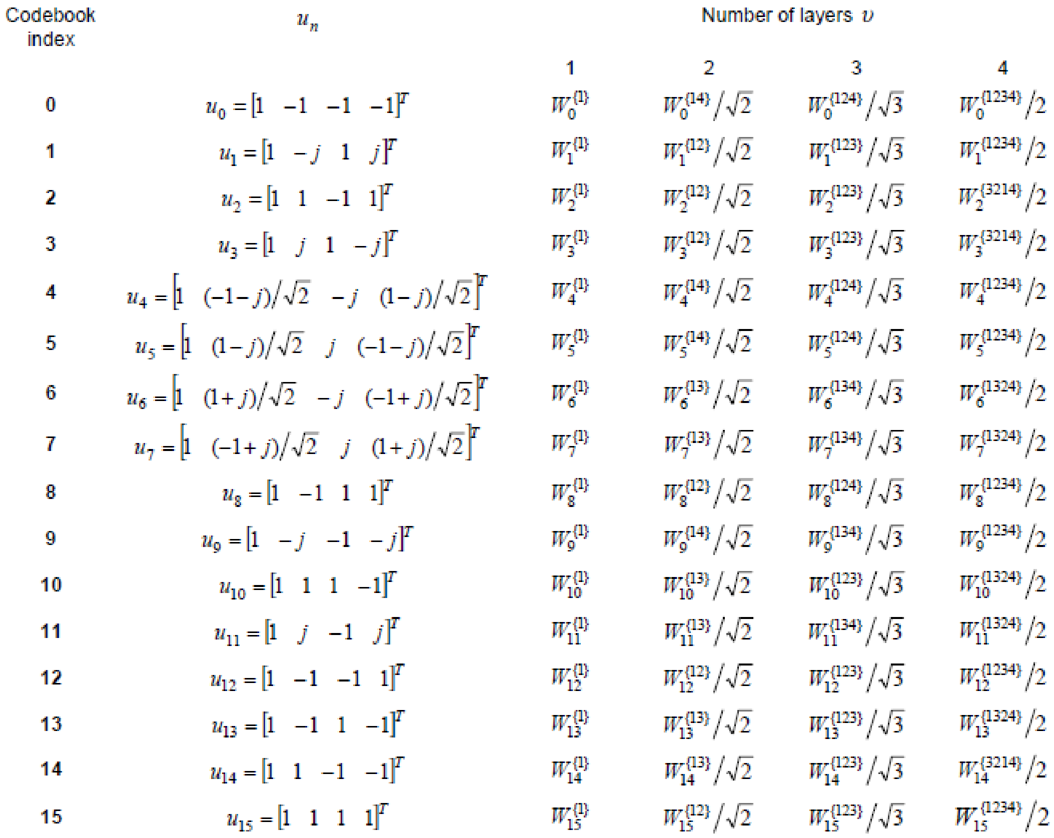
Source: https://www.sharetechnote.com/html/BasicProcedure_LTE_PHY_Precoding.html



Codebook index	Number of layers	
	1	2
0	$\frac{1}{\sqrt{2}}\begin{bmatrix}1\\1\end{bmatrix}$	$\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}$
1	$\frac{1}{\sqrt{2}}\begin{bmatrix}1\\-1\end{bmatrix}$	$\frac{1}{2}\begin{bmatrix}1 & 1\\1 & -1\end{bmatrix}$
2	$\frac{1}{\sqrt{2}}\begin{bmatrix}1\\j\end{bmatrix}$	$\frac{1}{2}\begin{bmatrix}1 & 1\\j & -j\end{bmatrix}$
3	$\frac{1}{\sqrt{2}}\begin{bmatrix}1\\-j\end{bmatrix}$	—

***Fig 7.1**: Codebook for two transmit antenna*

Source: https://www.sharetechnote.com/html/BasicProcedure_LTE_PHY_Precoding.html



Codebook index	u_n	Number of layers υ			
		1	2	3	4
0	$u_0 = [1 \quad -1 \quad -1 \quad -1]^T$	$W_0^{\{1\}}$	$W_0^{\{14\}}/\sqrt{2}$	$W_0^{\{124\}}/\sqrt{3}$	$W_0^{\{1234\}}/2$
1	$u_1 = [1 \quad -j \quad 1 \quad j]^T$	$W_1^{\{1\}}$	$W_1^{\{12\}}/\sqrt{2}$	$W_1^{\{123\}}/\sqrt{3}$	$W_1^{\{1234\}}/2$
2	$u_2 = [1 \quad 1 \quad -1 \quad 1]^T$	$W_2^{\{1\}}$	$W_2^{\{12\}}/\sqrt{2}$	$W_2^{\{123\}}/\sqrt{3}$	$W_2^{\{3214\}}/2$
3	$u_3 = [1 \quad j \quad 1 \quad -j]^T$	$W_3^{\{1\}}$	$W_3^{\{12\}}/\sqrt{2}$	$W_3^{\{123\}}/\sqrt{3}$	$W_3^{\{3214\}}/2$
4	$u_4 = \left[1 \quad (-1-j)/\sqrt{2} \quad -j \quad (1-j)/\sqrt{2}\right]^T$	$W_4^{\{1\}}$	$W_4^{\{14\}}/\sqrt{2}$	$W_4^{\{124\}}/\sqrt{3}$	$W_4^{\{1234\}}/2$
5	$u_5 = \left[1 \quad (1-j)/\sqrt{2} \quad j \quad (-1-j)/\sqrt{2}\right]^T$	$W_5^{\{1\}}$	$W_5^{\{14\}}/\sqrt{2}$	$W_5^{\{124\}}/\sqrt{3}$	$W_5^{\{1234\}}/2$
6	$u_6 = \left[1 \quad (1+j)/\sqrt{2} \quad -j \quad (-1+j)/\sqrt{2}\right]^T$	$W_6^{\{1\}}$	$W_6^{\{13\}}/\sqrt{2}$	$W_6^{\{134\}}/\sqrt{3}$	$W_6^{\{1324\}}/2$
7	$u_7 = \left[1 \quad (-1+j)/\sqrt{2} \quad j \quad (1+j)/\sqrt{2}\right]^T$	$W_7^{\{1\}}$	$W_7^{\{13\}}/\sqrt{2}$	$W_7^{\{134\}}/\sqrt{3}$	$W_7^{\{1324\}}/2$
8	$u_8 = [1 \quad -1 \quad 1 \quad 1]^T$	$W_8^{\{1\}}$	$W_8^{\{12\}}/\sqrt{2}$	$W_8^{\{124\}}/\sqrt{3}$	$W_8^{\{1234\}}/2$
9	$u_9 = [1 \quad -j \quad -1 \quad -j]^T$	$W_9^{\{1\}}$	$W_9^{\{14\}}/\sqrt{2}$	$W_9^{\{134\}}/\sqrt{3}$	$W_9^{\{1234\}}/2$
10	$u_{10} = [1 \quad 1 \quad 1 \quad -1]^T$	$W_{10}^{\{1\}}$	$W_{10}^{\{13\}}/\sqrt{2}$	$W_{10}^{\{123\}}/\sqrt{3}$	$W_{10}^{\{1324\}}/2$
11	$u_{11} = [1 \quad j \quad -1 \quad j]^T$	$W_{11}^{\{1\}}$	$W_{11}^{\{13\}}/\sqrt{2}$	$W_{11}^{\{134\}}/\sqrt{3}$	$W_{11}^{\{1324\}}/2$
12	$u_{12} = [1 \quad -1 \quad -1 \quad 1]^T$	$W_{12}^{\{1\}}$	$W_{12}^{\{12\}}/\sqrt{2}$	$W_{12}^{\{123\}}/\sqrt{3}$	$W_{12}^{\{1234\}}/2$
13	$u_{13} = [1 \quad -1 \quad 1 \quad -1]^T$	$W_{13}^{\{1\}}$	$W_{13}^{\{13\}}/\sqrt{2}$	$W_{13}^{\{123\}}/\sqrt{3}$	$W_{13}^{\{1324\}}/2$
14	$u_{14} = [1 \quad 1 \quad -1 \quad -1]^T$	$W_{14}^{\{1\}}$	$W_{14}^{\{13\}}/\sqrt{2}$	$W_{14}^{\{123\}}/\sqrt{3}$	$W_{14}^{\{3214\}}/2$
15	$u_{15} = [1 \quad 1 \quad 1 \quad 1]^T$	$W_{15}^{\{1\}}$	$W_{15}^{\{12\}}/\sqrt{2}$	$W_{15}^{\{123\}}/\sqrt{3}$	$W_{15}^{\{1234\}}/2$

$$W_n = I - 2u_n u_n^H / u_n^H u_n$$

__Fig 7.2__: Codebook for four transmit antenna

Capacity Analysis of the Codebook:

In order to maximise the capacity of the channel, the receiver has to search for the best capacity for all possible precoder matrices. The capacity for all precoder matrices is to be calculated using the equation.

$$C = log_2 \left[\det \left\{ \frac{E_s}{N_T N_0} H R_{PP} H^H + I \right\} \right]$$

A MATLAB code was written to search for the best precoder matrix in order to maximise the capacity. The code was run for ten different Rayleigh fading channels each time and observations were made.

Observation and Result:

- A direct relation between SNR and best Layer for Precoder was observed. At the lower SNRs (0 dB or below) the Layer 1 Precoder matrices performed better than the higher layer matrices. As the SNR of the signal increases, it was found that the higher SNRs perform better and for 12 dB or more, layer 4 matrices consistently performed better than other matrices.
- It was observed that the capacity remains constant for all precoder matrices of the Layer 4. This observation can be arrived at, theoretically too by taking into account that R_{PP} value remains constant and fixed at 1 for all the Precoder matrices of layer 4.
- No direct relation between the channel matrix and the optimum precoder matrix could be established.

SNR (in dB)	Channel Capacity (b/s/Hz)	Optimum Layer	Optimum Precoder
0	3.7282	2	6
3	5.3683	3	6
6	7.5356	3	6
9	10.0193	3	6
12	12.7427	4	1
15	16.0796	4	1

__Table 7.1__: Optimum precoder matrix for various SNRs for the channel matrix

$$H = \begin{bmatrix} 0.38 + 0.60i & 0.90 - 0.31i & 1.42 - 0.73i & 1.18 - 0.82i \\ -0.06 + 0.27i & -0.57 + 0.07i & 0.01 - 0.22i & 0.08 + 1.68i \\ -0.53 - 0.81i & -0.87 - 0.17i & 0.21 + 0.54i & 0.37 + 1.07i \\ -0.49 + 0.02i & 0.15 - 0.13i & -0.66 + 1.23i & -0.67 + 0.11i \end{bmatrix}$$

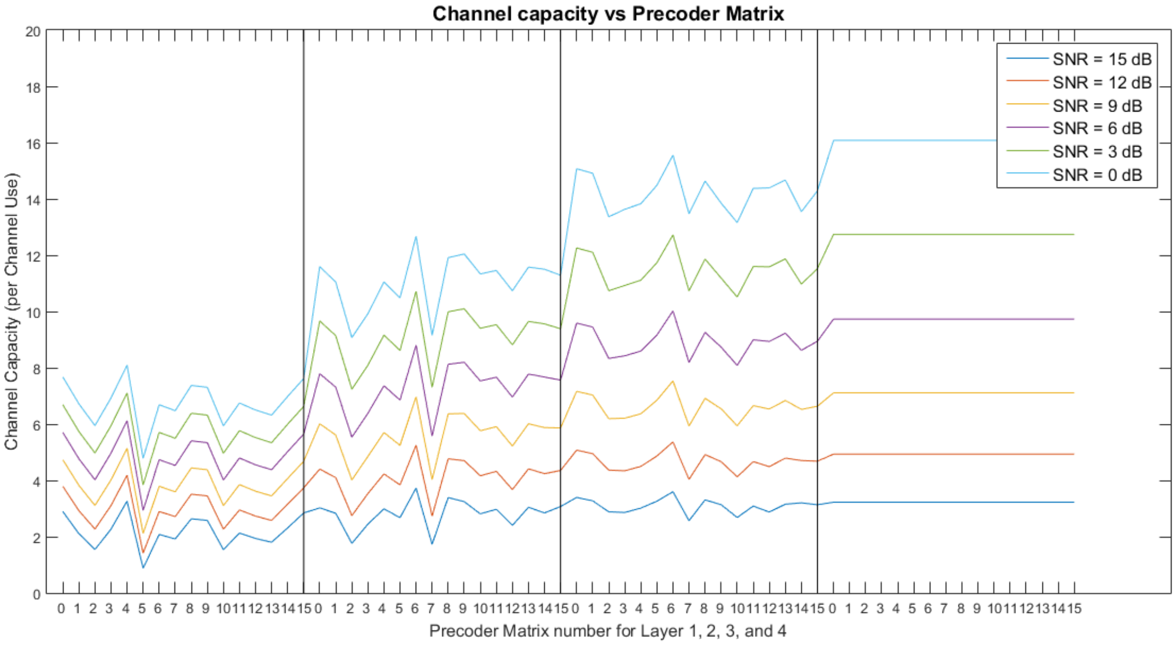


***Fig 7.3**: Channel capacity vs Precoder Matrix for channel matrix mentioned above.*

CHAPTER 8

PACKET DETECTOR

The double window energy ratio method for frame detection is briefly explained in by Juha Heiskala and John Terry. For this work, we derive the expressions for threshold that optimizes detection for a given probability of false alarm (Neyman Pearson detector).

The assumptions for the signal model are that the noise is AWGN with a standard deviation σ_n and the received packet is also white Gaussian with a standard deviation σ_s which is generally higher than σ_n.

In this method, we take the ratio of energies in two sliding windows. Each window is of length N samples and there is a separation of L samples between the two windows. We consider the two hypothesis H_0 and H_1. H_0 is the case where both the windows have all noise samples. H_1 is the case where first window has all data samples and the second window has all noise samples. The two hypotheses are shown in Figure 4.1 and Figure 4.2.

In Figure 4.1 the *F* window and the *B* window, both contain samples from noise. In Figure 4.2, the *F* window contain samples of signals and the *B* window contains samples from frame. Although the figures show 1D data, practically, the samples are complex baseband symbols. The energy in each window is calculated as E_F and E_B, and their ratio ρ_i is calculated. The windows are sliding over the receiver buffer so ρ_i is calculated for each position *i*.

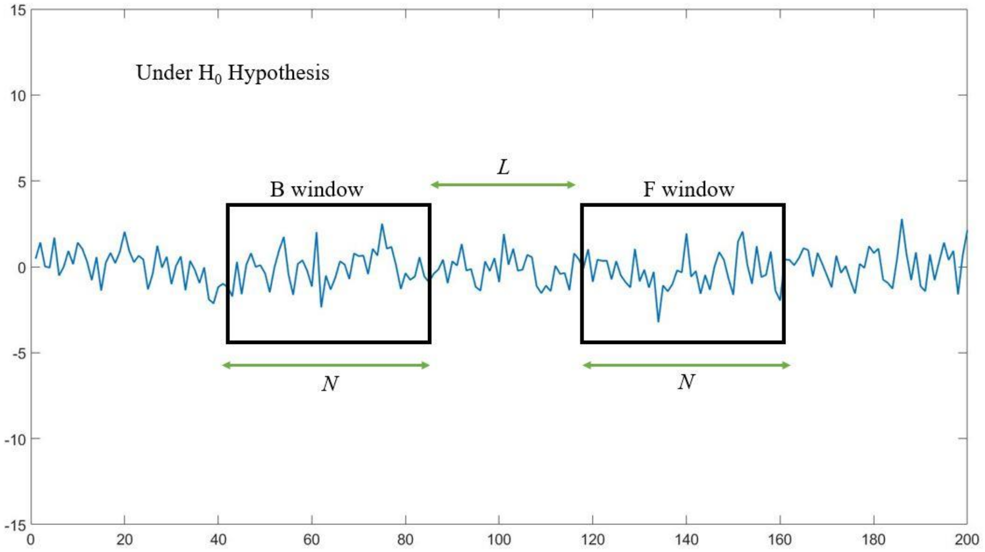


Fig 8.1: *Hypothesis H_0*

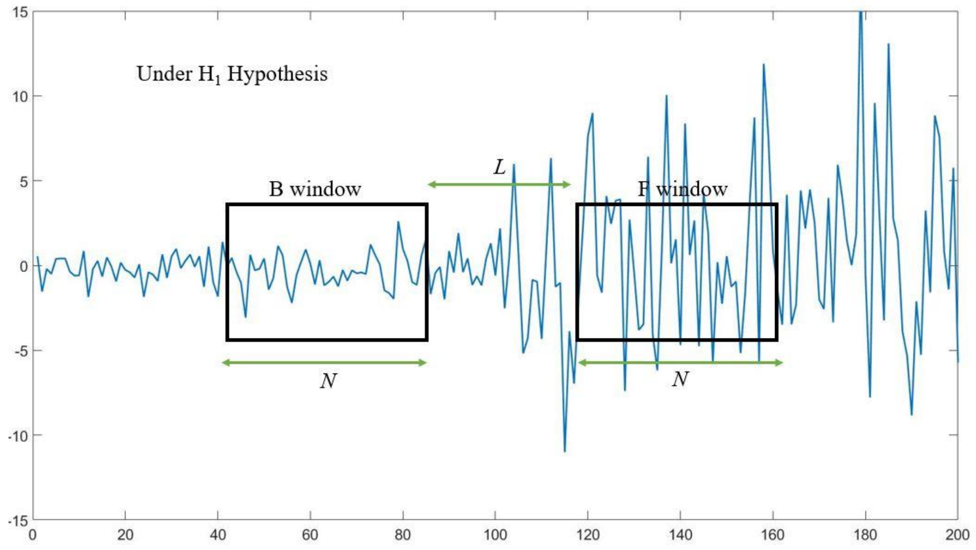


Fig 8.2: *Hypothesis H_1*

The energy in the forward and the back window can be given by:

$$E_F[i] = \sum_l r[i+N+L+l]\overline{r[i+N+L+l}$$

$$E_B[i] = \sum_l r[i+l]\overline{r[i+l}$$

$for\ l\ ranging\ from\ 0\ \ to\ N-1.$

Also, the ratio of instant energy is given by

$$\rho_i = \frac{E_F[i]}{E_B[i]}$$

Derivation for Hypothesis H_0:

For the null Hypothesis, we have

$$r[k] = w[k] \quad where \quad Re\{w\}, Im\{w\} \sim N(0, \sigma_n^2)$$

$$\therefore r[k]\overline{r[k]} \sim exp(2\sigma_n^2) \quad where \quad exp(2\sigma_n^2) = \frac{1}{2\sigma_n^2} e^{\frac{-r\overline{r}}{2\sigma_n^2}}$$

$$\therefore \frac{r\bar{r}}{2\sigma_n^2} \sim \chi^2(2)$$

where $\chi^2(2)$ refers to chi square distribution of 2 degrees of freedom.

Sum of N such random variables has the following distribution:

$$\therefore \frac{\sum_k r[k]\overline{r[k]}}{2\sigma_n^2} = \frac{E}{2\sigma^2} \sim \chi^2(2N) = \frac{(\frac{\sum r\bar{r}}{2\sigma_n^2})^{N-1} e^{(\frac{-\sum r\bar{r}}{4\sigma_n^2})}}{2^N\Gamma(N)} = \frac{(\frac{E}{2\sigma_n^2})^{N-1} e^{(\frac{-E}{4\sigma_n^2})}}{2^N\Gamma(N)}$$

Under null hypothesis, the energies in the *F* window and the *B* window are statistically similar. Thus, the distribution of E_F and E_B are same under null hypothesis.

Now consider their ratio ρ. We shall obtain it's distribution under null hypothesis.

$$\rho = \frac{\sum_k r[k_f]\overline{r[k_f]}}{\sum_k r[k_b]\overline{r[k_b]}} = \frac{\frac{\sum_k r[k_f]\overline{r[k_f]}}{2\sigma_n^2}}{\frac{\sum_k r[k_b]\overline{r[k_b]}}{2\sigma_n^2}}$$

Thus, ρ is a ratio of two standard χ^2 distributed random variables each with $2N$ degrees of freedom. The distribution of such a ratio is defined as *Beta prime distribution*. Beta prime distribution with parameters α and β is given as:

$$\beta'_\rho(\alpha,\beta) = \frac{\rho^{\alpha-1}(1+\rho)^{-\alpha-\beta}}{B(\alpha,\beta)}$$

Here $B(\alpha,\beta)$ is the beta function which is given in terms of Gamma function as

$$\frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}$$

The β^0 distribution is related to the Gamma distribution as:

$$X_k \sim \Gamma(\alpha_k,\theta_k) => \frac{X_1}{X_2} \sim \beta'(\alpha_1,\alpha_2,1,\frac{\theta_1}{\theta_2}$$

The Gamma distribution is related with the χ^2 distribution of $2N$ degrees of freedom as: $\chi^2(2N) = \Gamma(N,2)$. Using this, the distribution of ρ can be written as:

$$\rho \sim \beta'_p(N,N,1,1)$$

which boils down to $\beta'_p(N,N)$. Therefore,

$$\rho|_{H0} \sim \beta'_\rho(N,N) = \frac{\rho^{N-1}(1+\rho)^{-2N}}{B(N,N)}$$

Derivation for Hypothesis H_1:

For H_1 hypothesis, the received signal model differs for *F* window from the model seen in H_0 case. Whereas for the *B* window the model remains the same as the H_0 case. The *F* window now contains the OFDM signal packet. So, for *B* window

$$r_B[k] = w[k] \quad where \quad Re\{w\}, Im\{w\} \sim N(0,\sigma_n^2)$$

and for *F* window:

$$r_F\ [k] = \ s[k] + w[k]\ where\ Re\{w\}, Im\{w\} \sim\ \ N(0, \sigma_n^2)\ and$$
$$Re\{s\}, Im\{s\} \sim\ \ N(0, \sigma_s^2)$$
$$Re\{r_F\}, Im\{r_F\} \sim N(0, \sigma_s^2 + \sigma_N^2)$$

The distribution of the energy of the *B* window remains same as obtained above. Following, similar arguments, we can represent the distribution of energy in *F* window under H_1 scenario in terms of standard χ^2 distribution as follows:

$$\frac{\sum_k r_F[k]\overline{r_F[k]}}{2(\sigma_n^2+\sigma_s^2)} = \frac{E_F}{2(\sigma_n^2+\sigma_s^2)} \sim \chi^2(2N) = \frac{(\frac{E_F}{2(\sigma_n^2+\sigma_s^2)})^{N-1} e^{(\frac{-E_F}{4(\sigma_n^2+\sigma_s^2)})}}{2^N\Gamma(N)}$$

So, for the ratio of the energies of the two window we may calculate the distribution as follows:

$$\rho = \frac{E_F}{E_B} = \frac{\frac{E_F}{2(\sigma_s^2+\sigma_n^2)}}{\frac{E_B}{2\sigma_n^2}} \frac{(\sigma_n^2+\sigma_s^2)}{\sigma_n^2}$$

$$Let \quad \rho' = \rho\frac{\sigma_n^2}{\sigma_n^2+\sigma_s^2} = \frac{\frac{E_F}{2(\sigma_s^2+\sigma_n^2)}}{\frac{E_B}{2\sigma_n^2}}$$

The numerator and the denominator on the RHS are already shown to be standard χ^2 distributed with $2N$ degrees of freedom. Their ratio, as discussed in equations 7 to 10 takes the form of standard β' distribution.

$$\rho' \sim \beta'(N,N) = \frac{\rho'^{N-1}(1+\rho')^{-2N}}{B(N,N)}$$

$$\therefore \rho\frac{\sigma_n^2}{\sigma_n^2+\sigma_s^2} \sim \frac{\rho\frac{\sigma_n^2}{\sigma_n^2+\sigma_s^2}^{N-1}(1+\rho\frac{\sigma_n^2}{\sigma_n^2+\sigma_s^2})^{-2N}}{B(N,N)}$$

$$\therefore \rho|_{H1} \sim \frac{\rho\frac{\sigma_n^2}{\sigma_n^2+\sigma_s^2}^{N-1}(1+\rho\frac{\sigma_n^2}{\sigma_n^2+\sigma_s^2})^{-2N}}{B(N,N)}\frac{\sigma_n^2}{\sigma_n^2+\sigma_s^2} = (\frac{\sigma_n^2}{\sigma_n^2+\sigma_s^2})^N\frac{\rho^{N-1}(1+\rho\frac{\sigma_n^2}{\sigma_n^2+\sigma_s^2})^{-2N}}{B(N,N)}$$

Thus, we now have the distribution of the energy ratio under both the hypothesis. We may now derive the detectors that optimally distinguishes between the two hypothesis. We shall first try to obtain the LRT based detector.

LRT based Detector:

First, we solve the likelihood ratio test.

$$LRT: \quad \frac{P(\rho|_{H1})}{P(\rho|_{H0})} > \tau$$

Which is true for H_1.

Here, τ is the optimum threshold in NP sense. We wish to obtain expression of the threshold τ or maybe a simplified threshold which could be more relevant for our measured quantity ρ. It must be noted that though the process is similar to Neyman Pearson Theorem, it is not that. As the Neyman Pearson criterion must be applied only on raw data while our start is not the raw data but ratio of two energies from sliding window.

Replacing the expressions of the distributions from previous equations we get

$$LRT: \quad (\frac{\sigma_n^2}{\sigma_n^2+\sigma_s^2})^N(\frac{1+\rho\frac{\sigma_n^2}{\sigma_n^2+\sigma_s^2}}{1+\rho})^{-2N} > \tau$$

If we let

$$\frac{\sigma_n^2}{\sigma_n^2+\sigma_s^2} = q$$

We get the following equation for H_1 being true.

$$\therefore (\frac{1+q\rho}{1+\rho})^{-2N} > \tau q^{-N}$$

$$\therefore \frac{1+\rho}{1+q\rho} > \tau^{\frac{1}{2N}} q^{-0.5}$$

$$\therefore \frac{1}{q} - \frac{1+\rho}{1+q\rho} < \frac{1}{q} - \tau^{\frac{1}{2N}} q^{-0.5}$$ when H_1 is true

$$\Rightarrow \frac{1-q}{1+q\rho} < 1 - \tau^{1/2N} q^{-0.5}$$

$$\Rightarrow 1 + q\rho > \frac{1-q}{1-\tau^{1/2N} q^{-0.5}}$$

$$\Rightarrow \rho > \frac{1-q}{q(1-\tau^{1/2N} q^{-0.5})} - \frac{1}{q}$$

We make the $RHS = \tau'$ and hence obtain H_1 is true whenever $\rho > \tau'$.

Thus, τ*’* partitions the space of values of ρ into two. All the values of ρ measured larger than τ*’* represent the case of H_1 hypothesis. We now find the expression for τ*’* using the conditions of NP detector.

We define probability of detection P_d as the probability that H_1 is true and it is interpreted as H_1. This will be obtained as the probability that $\rho|_{H1} > \tau'$. Or we may write:

$$P_d = \int_{\tau'}^{\infty} P(\rho|_{H1})d\rho$$

Next, we define the probability of false alarm (P_f) as the probability that H_0 is the true hypothesis but H_1 is detected. This we may calculate as the probability that $\rho|_{H0} > \tau'$. The expression for P_f can we obtained from:

$$P_f = \int_{\tau'}^{\infty} P(\rho|_{H0})d\rho$$

The expressions or values of P_d and P_f are important to quantify the performance of the detector. Under NP detection, we calculate the threshold(s) such that the probability of false alarm is always less than a user specified value α.

So, we obtain the optimum threshold τ' by limiting the probability of false alarm as follows:

$$\alpha = \int_{\tau'}^{\infty} P(\rho|_{H0})d\rho = \int_{\tau'}^{\infty} \beta'_{\rho}(N, N)d\rho$$

$$\therefore \alpha = \int_{\tau'}^{\infty} \frac{\rho^{N-1}(1+\rho)^{-2N}}{B(N, N)}d\rho$$

From the above equation we see that the RHS is a function of τ' and N. So, for a given α we can solve the equation numerically to calculate τ'. Thus, the threshold is dependent on SNR, the chosen probability of false alarm, and window size.

CHAPTER 9

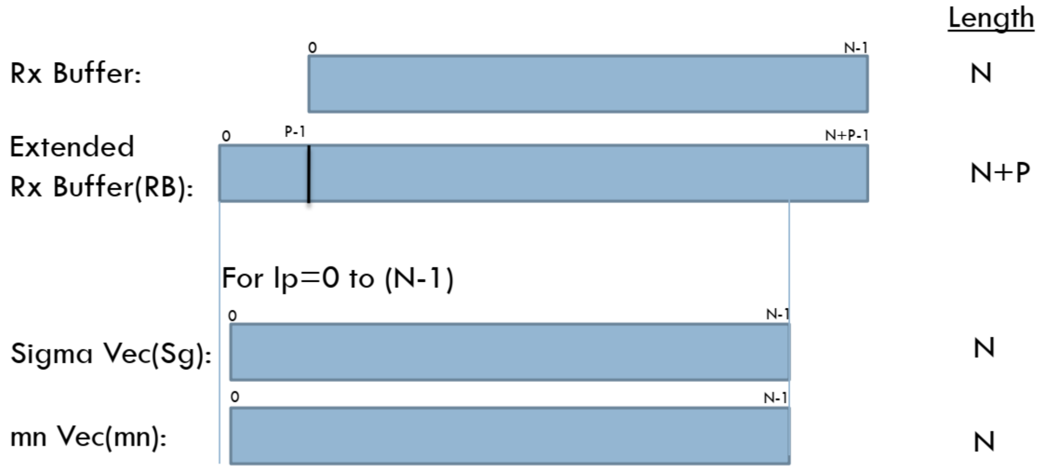
AN ALGORITHM FOR PACKET DETECTION

An algorithm for the packet detection using the method discussed in the last chapter was developed during the project. The algorithm can run on any sequential computer. The algorithm was implemented on NI-USRP using the LabVIEW platform.

System Model Description:

The input to algorithm is assumed to be a buffer (equivalent of pointer) of fixed size that is filled with baseband data. Every time the buffer is filled, the algorithm is called, the algorithm computes the ρ value described in previous chapter that is the ratio of two sliding window energies and returns it in output buffer (equivalent of pointer) of fixed size.

The memory elements of the pseudocode can be summarised with the following diagram.



$$Sg(lp) = Sg(lp-1)modN - |RB(lp)|^2 + |RB(lp+P)|^2$$

$$mn(lp) = \frac{Sg(lp)}{Sg((lp-64)modN)}$$

Fig 9.1: *Memory elements of the algorithm for packet detection.*

The pseudocode for the proposed algorithm is given below

```
/* Variable description and initialisation
p <- Pointer to input buffer
gap_size <- Gap between two windows
```

```
win_size <- Window size
N <- length of input buffer
Sigma[0:N-1] <- An array of size N that stores energy of a
window
past[0:win_size-1] <- Array that stores data from last buffer.
Output[0:N-1] <- Pointer to output buffer.
*/

If (first_loop)
     Sigma(win_size-1) = sum(p[0:win_size-1]);
Else
     For i=0 to win_size
          Sigma(i) = Sigma[(i-1)%N] + p[i] – past[i];
          output = sigma[i]/sigma[(i-gap_size-win_size)%N];

For i= win_size:N-1
     Sigma(i) = Sigma(i-1) + p(i) – p(i-win_size);
     output = sigma[i]/sigma[(i-gap_size-win_size)%N];

copy p[N-win_size:N-1] into past[0:win_size];
```

Although the algorithm presented here is exclusively for sequential computers, the low-level interface nature of it makes it very easy to be converted into algorithm for parallel computers or even hardware. That is with minimal changes the algorithm can be developed into hardware using any hardware description language and can be implemented on FPGA or ASIC.

Simulation of the packet detection:

The scenario of packet transmission and reception under the scenario discussed was simulated on MATLAB understand the deviation form theoretical results derived in Chapter 8.

The scenario was simulated for various window lengths and signal to noise ratios. The simulation result was found to coincide with theoretical derivation as summarised by the graph below.

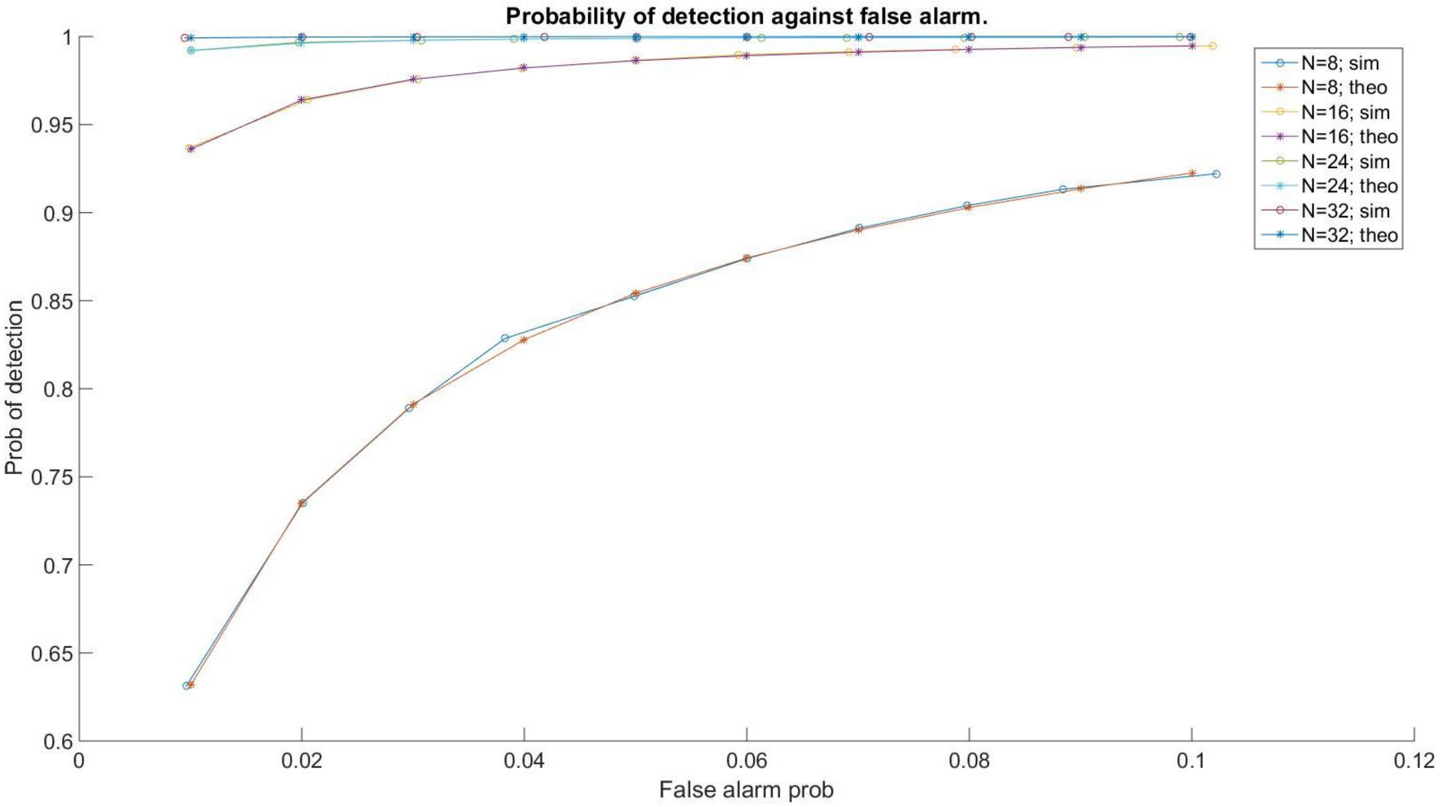


Fig 9.2: *Theoretical and Simulation P_{fa} against P_d.*

The Receiver Operating Characteristics for various lengths of N was observed at different SNRs as shown below.

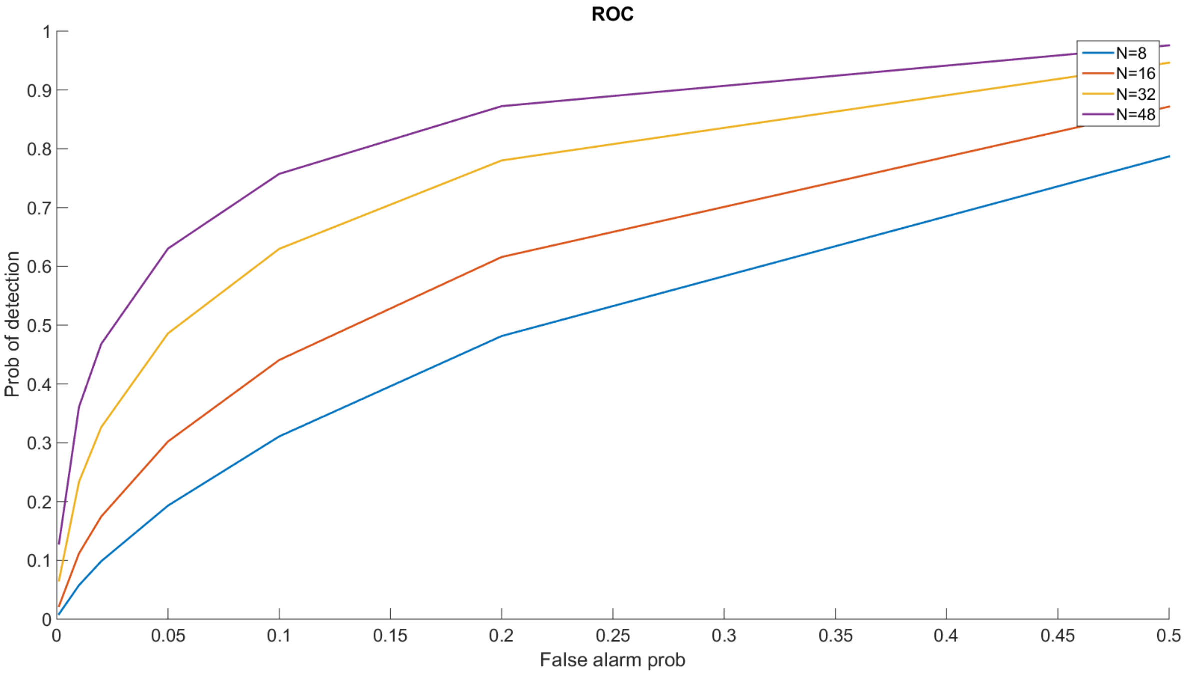


Fig 6.2: *ROC for SNR = -3dB.*

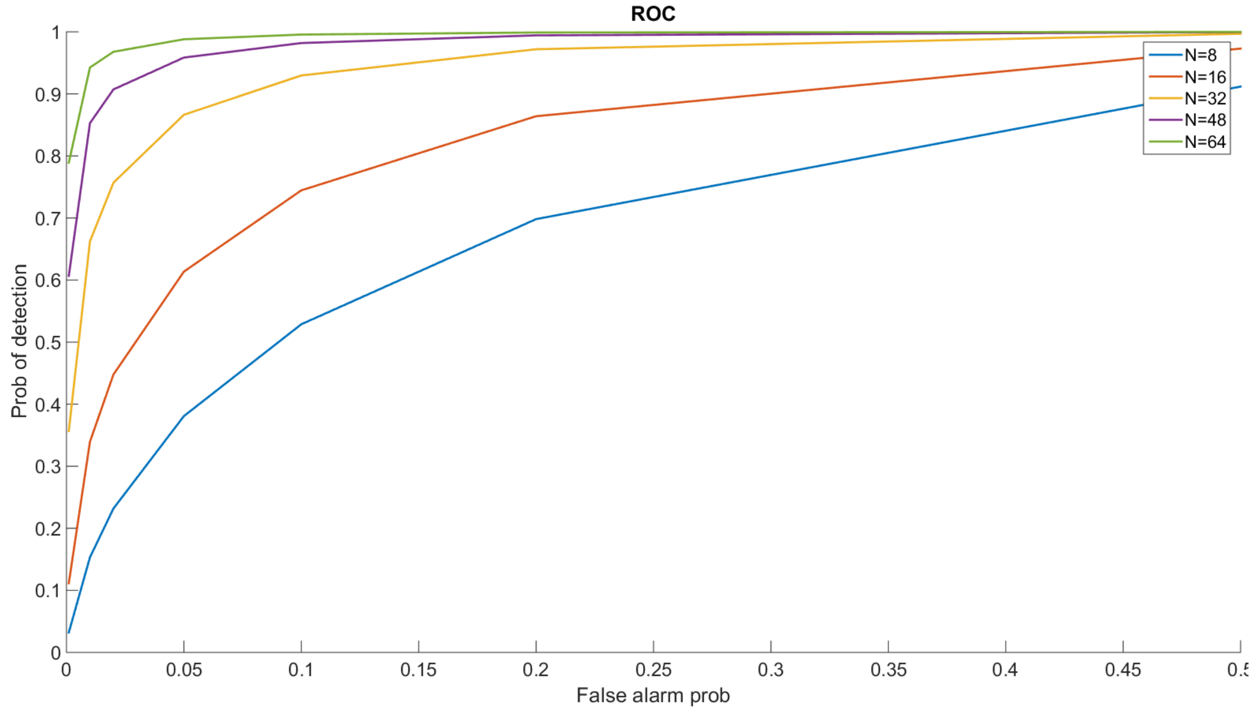


Fig 9.3: *ROC for SNR = 0dB.*

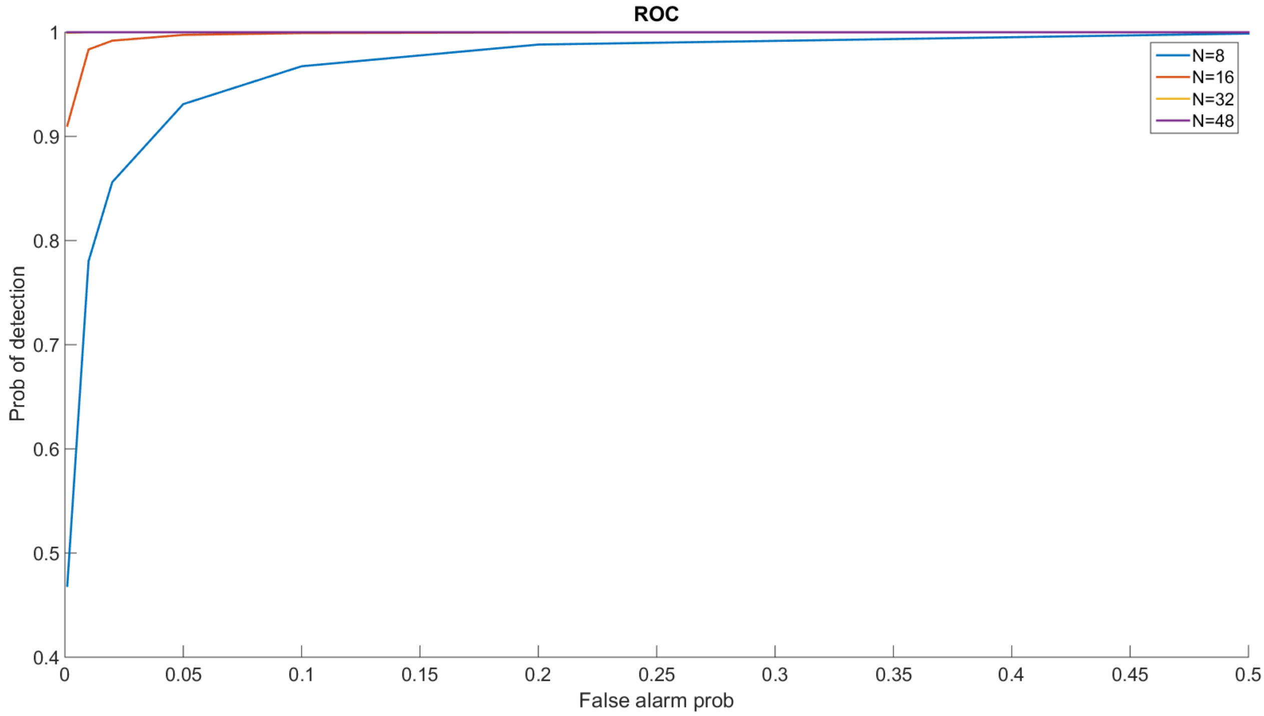


Fig 9.4: *ROC for SNR = 6dB.*

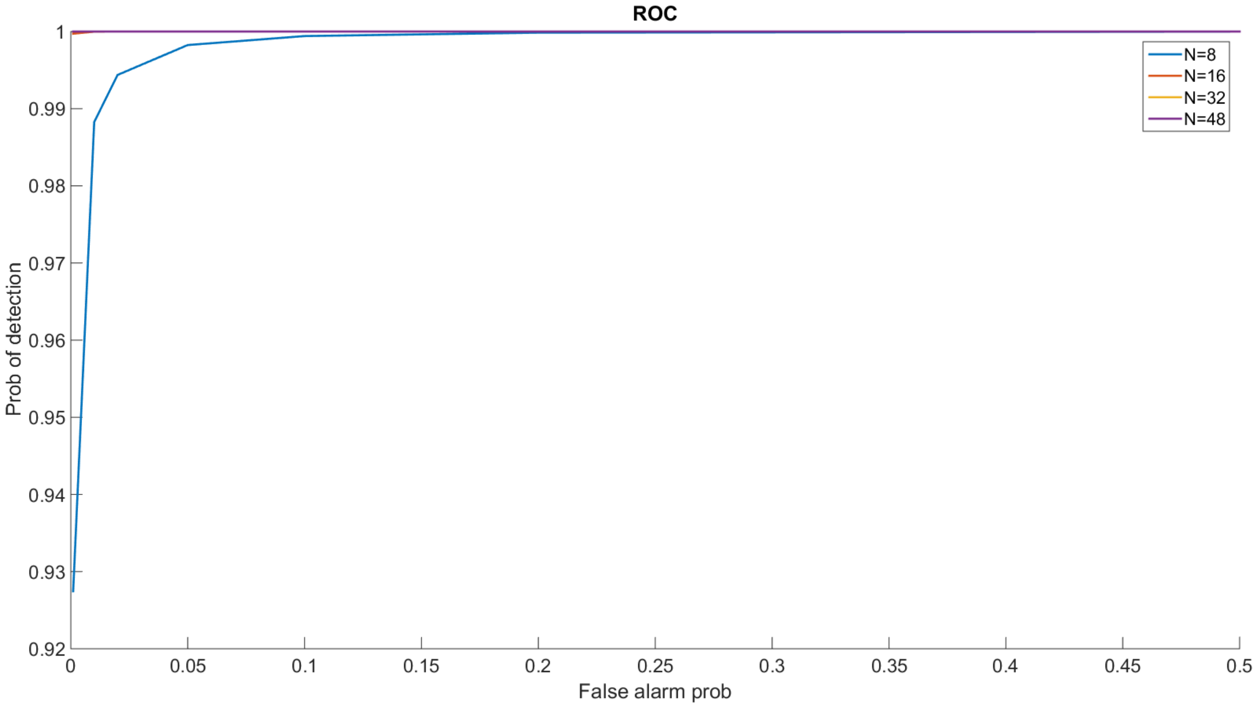


Fig 9.5: *ROC for SNR = 10dB.*

The behaviour of the ROC with increase in SNR is just as expected. The probability of false alarm decreases as the effect of noise decreases. Also, as expected, the receiver performs better when the size of the window in double sliding window algorithm is increases. This is intuitive too. As the number of observation samples increases the probability of making a better guess should increase.

It was observed during the simulation that the computation of the beta prime function is a computationally heavy process and it was impossible to compute the beta prime function for large window sizes at high SNR. This appears to be a disadvantage of the algorithm. However, it must be noted that the curves shown above need to be computed only once and then the threshold values according to our P_{fa} and P_d requirement can be hard coded into the algorithm.

CHAPTER 10

HIGH THROUGHPUT MIMO CHANNEL MODEL

Overview:

A document presented by High Throughput Task Group of IEEE presents the channel models that are directly relevant to the IEEE 802.11a. The channel models with some modification (introduction of MIMO) can be used to simulate channels for 802.11n scenario. A brief description of the MIMO channel models is presented in this chapter.

MIMO:

As discussed in prior chapters, MIMO was introduced in IEEE 802.11n. The use of multiple antennas offers extended range, improved reliability and higher throughputs than conventional single antenna communication systems. Multiple antenna systems can be generally separated into two main groups: smart antenna-based systems and spatial multiplexing based multiple-input multiple-output (MIMO) systems.

In MIMO systems, each transmit antenna can broadcast at the same time and in the same bandwidth an independent signal sub-stream. This corresponds to the second category of multi-antennas systems, referred to as spatial multiplexing-based multiple-input multiple-output (MIMO) systems. Using this technology with *n* transmit and *n* receive antennas, for example, an *n*-fold increase in data rate can be achieved over a single antenna system.

The channel matrix *H* fully describes the propagation channel between all transmit and receive antennas. If the number of receive antennas is *n* and transmit antennas is *m*, the channel matrix *H* has a dimension of *n* x *m*. To arrive at channel matrix *H*, we use a method that employs correlation matrix and i.i.d. matrix (zero-mean unit variance independent complex Gaussian random variables).

SISO Model:

The TGn document states five channel models for the indoor Non-Line of Sight and Line of Sight scenarios. The first channel, denoted by Model A is single tap channel with unity variance. Other channel models (Channel B to E) consist of clusters as described by Saleh and Valenzuela model. Each of the cluster contains multiple taps and their power distribution. The TGn documents provides a much-detailed description of the channel models that is skipped in this chapter for the sake of redundancy.

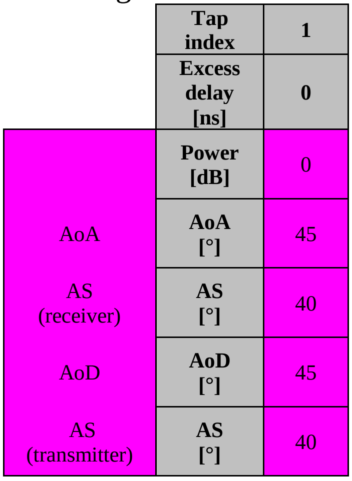
Description of environment for channel model A to E

- Model A for a typical office environment, non-line-of-sight (NLOS) conditions, and 50 ns rms delay spread.
- Model B for a typical large open space and office environments, NLOS conditions, and 100 ns rms delay spread.
- Model C for a large open space (indoor and outdoor), NLOS conditions, and 150 ns rms delay spread.
- Model D, same as model C, line-of-sight (LOS) conditions, and 140 ns rms delay spread (10 dB Ricean *K*-factor at the first delay).
- Model E for a typical large open space (indoor and outdoor), NLOS conditions, and 250 ns rms delay spread.

In our simulation of the channel we have considered the channel to be purely Rayleigh in nature and Ricean factor is forced to null.

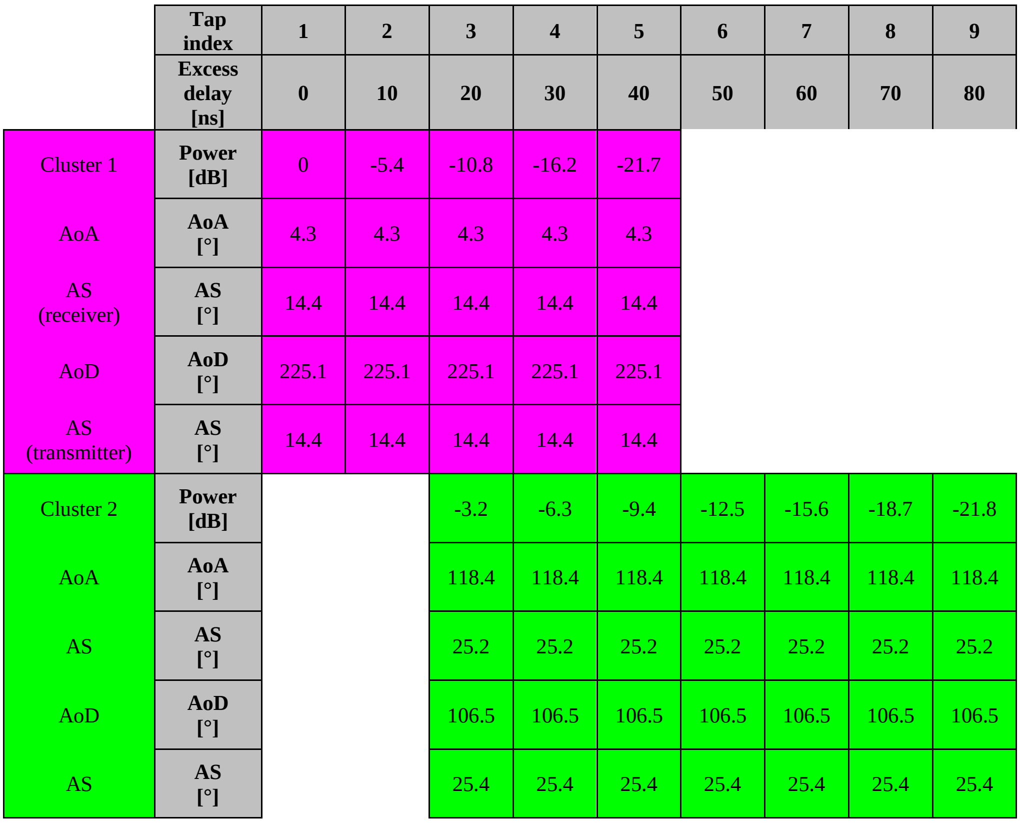
Model A and Model B are shown below:

Source: tgn-channel-models, Appendix C, model A, Page 34



	Tap index	**1**
	Excess delay [ns]	**0**
	Power [dB]	0
AoA	**AoA [°]**	45
AS (receiver)	**AS [°]**	40
AoD	**AoD [°]**	45
AS (transmitter)	**AS [°]**	40

Fig 10.1: *Channel model A*



	Tap index	1	2	3	4	5	6	7	8	9
	Excess delay [ns]	0	10	20	30	40	50	60	70	80
Cluster 1	Power [dB]	0	-5.4	-10.8	-16.2	-21.7				
AoA	AoA [°]	4.3	4.3	4.3	4.3	4.3				
AS (receiver)	AS [°]	14.4	14.4	14.4	14.4	14.4				
AoD	AoD [°]	225.1	225.1	225.1	225.1	225.1				
AS (transmitter)	AS [°]	14.4	14.4	14.4	14.4	14.4				
Cluster 2	Power [dB]			-3.2	-6.3	-9.4	-12.5	-15.6	-18.7	-21.8
AoA	AoA [°]			118.4	118.4	118.4	118.4	118.4	118.4	118.4
AS	AS [°]			25.2	25.2	25.2	25.2	25.2	25.2	25.2
AoD	AoD [°]			106.5	106.5	106.5	106.5	106.5	106.5	106.5
AS	AS [°]			25.4	25.4	25.4	25.4	25.4	25.4	25.4

***Fig 10.2:** Channel model B*

Source: tgn-channel-models, Appendix C, model B, Page 35

Extension to MIMO:

The channel models provide the power delay profile for various taps in the channel model that can be exploited to form the MIMO channel. For a $N_t \times N_r$ channel we follow the steps given below to generate a MIMO channel.

1. Choose a channel model and generate $N_t \times N_r$ channel profiles.
2. Form a matrix of size $N_t \times N_r$ using the channel profiles obtained in last step and call it H_{iid}.
3. Find the transmitter antenna and receiver antenna correlation matrices given by R_t and R_r.
4. Using the formula given below find the channel profile for the MIMO channel.

$$H = R_t^{1/2} H_{iid} R_r^{1/2}$$

Note: The process mentioned above should be followed for all channel taps.

Cluster Modeling:

The cluster model was introduced first by Saleh and Valenzuela and later verified, extended, and elaborated upon by many other researchers. The received signal amplitude β_{kl} is a Rayleigh-distributed random variable with a mean-square value that obeys a double exponential decay law

$$\overline{\beta_{kl}^{\,2}} = \overline{\beta^2(0,0)}e^{-T_l/\Gamma}e^{-\tau_{kl}/\gamma}$$

where $\overline{\beta^2(0,0)}$ represents the average power of the first arrival of the first cluster, T_l represents the arrival time of the lth cluster, and τ_{kl} is the arrival time of the kth arrival within the lth cluster, relative to T_l.

The number of clusters found in different indoor environments varies between 1 and 7. During the experiment for finding channel models by the TGn, the average number of clusters was found to be 3 for one building, and 7 for another building. The number of clusters reported was found to be 2 for line-of-sight (LOS) and 5 for non-LOS (NLOS) conditions.

Figure 8.3 shows Model D delay profile with clusters outlined by exponential decay (straight line on a log-scale).

Source: tgn-channel-models, Figure 1, Page 11

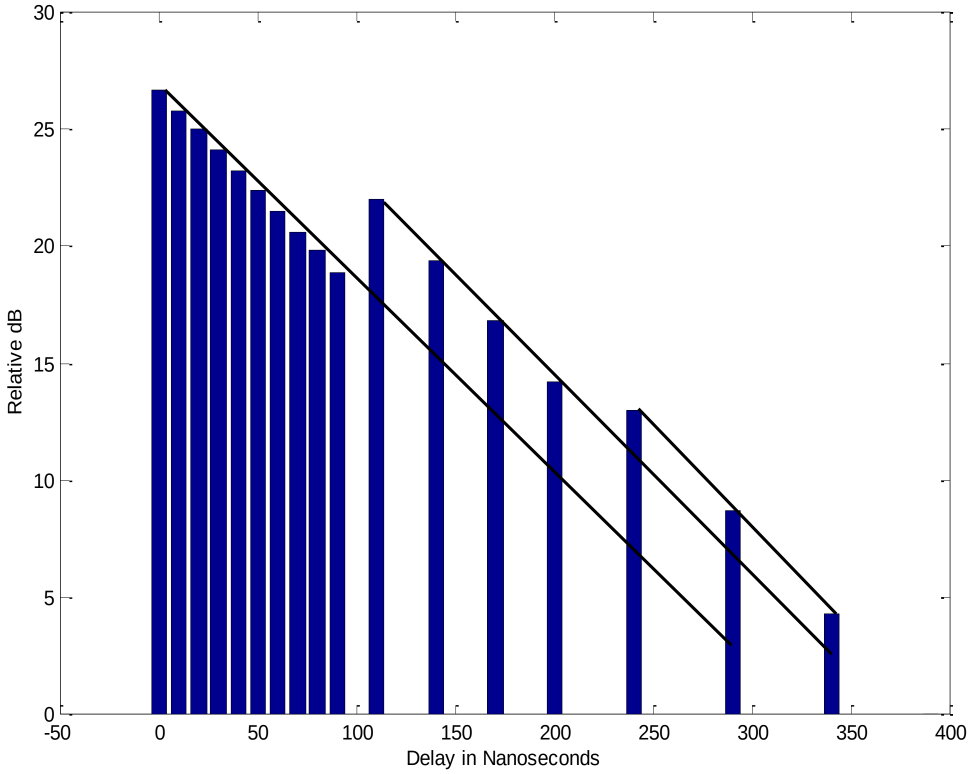


***Fig 10.3:** Model D delay profile with cluster extension (overlapping clusters).*

Clearly, three clusters can be identified. For Models B, C, D, E, and F, 2, 2, 3, 4 and 6 clusters can be identified, respectively. The number of clusters in each of the models B-F agrees well with the results reported in the literature. We recall that the model A consists of only one tap and hence single cluster.

CHAPTER 11

MIMO MODES AND THE SIMULATOR

The IEEE 802.11n standard supports various modes of MIMO based on single architecture. These are SISO, MRC, V-BLAST, DEM, STBC, and SVD. The IEEE 802.11 standards do not mention codebook-based MIMO schemes. However, they are quite popular and useful and can save feedback bits.

Single Input Single Output

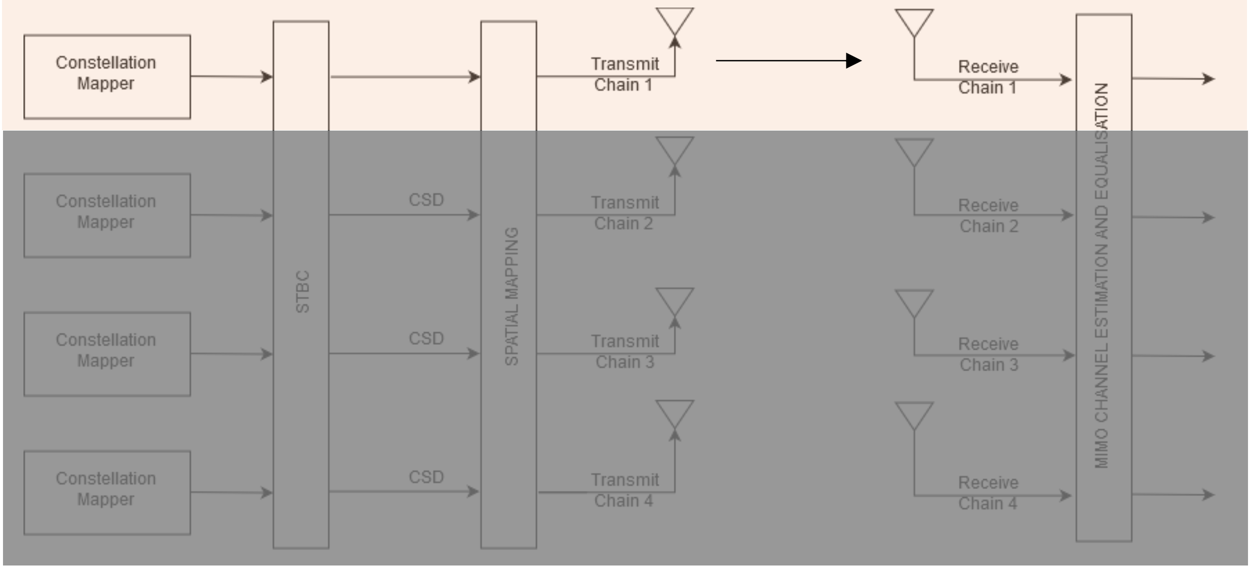


***Fig 11.1:** SISO mode in 802.11n HT-transmitter.*

In the SISO mode, only one spatial stream, space-time stream and a single transmit and receive chain is activated. This is the most fundamental kind of transmission. There is no diversity gain or multiplexing in this scheme.

Maximal Ratio Combining

In the maximal ratio combining mode, only one spatial and space-time stream is used. The MRC can be done at transmitter or the receiver. If the MRC is done at the transmitter, it is a case of transmit MRC otherwise it is receiver MRC.

Transmit MRC: Only one receiver is engaged in transmit MRC, while multiple transmit chains with single data stream is used. This leads to diversity gain and SNR enhancement.

Receive MRC: Single data stream and transmit chain is engaged in this kind of MIMO along with multiple receive chains. This scheme also benefits from diversity gain and undergoes SNR enhancement.

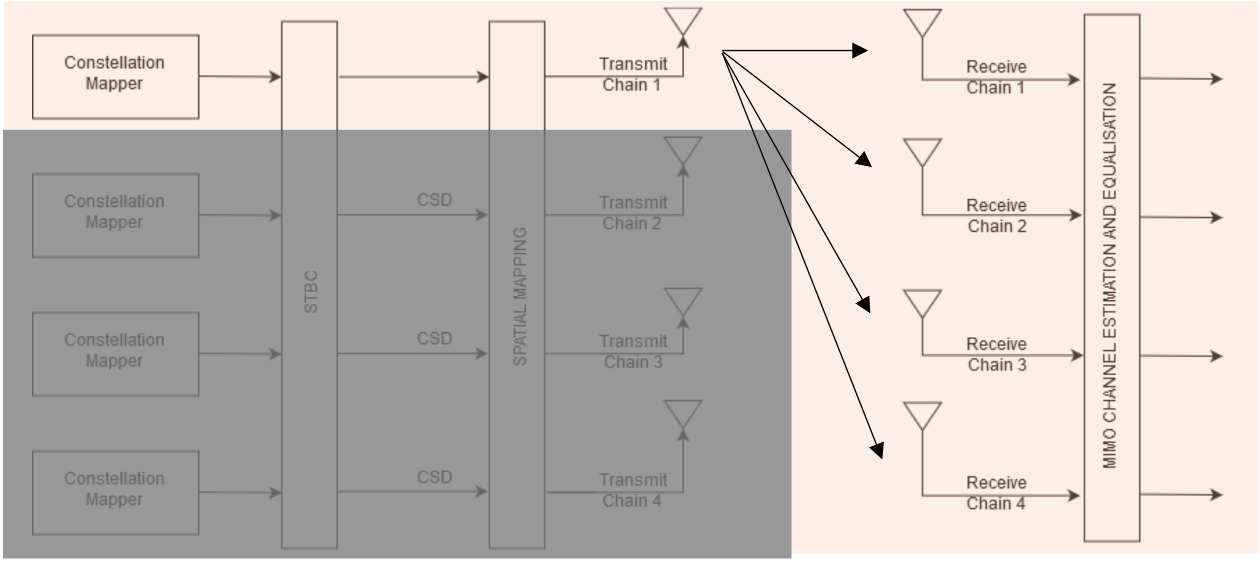


Fig 11.2: *Receive MRC mode in 802.11n HT-transmitter*.

V-BLAST

This scheme of MIMO benefits from spatial multiplexing and uses full multiplexing. Equal number of spatial streams, space time streams, transmit and receive chains are involved in this kind of transmission. An identity matrix is used for the spatial mapping.

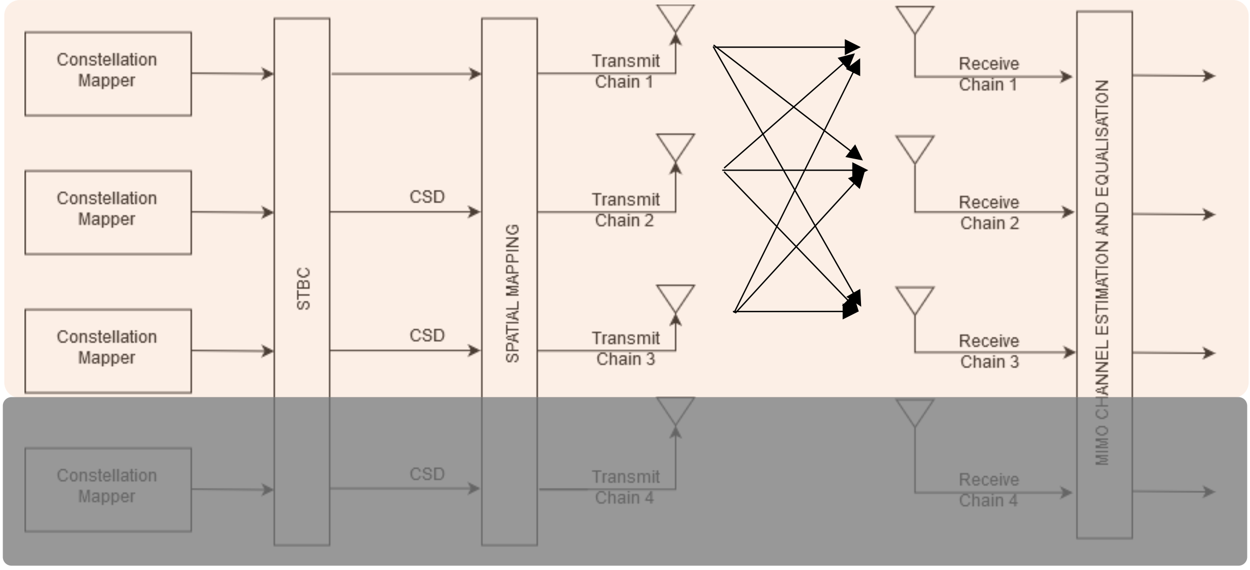


Fig 11.3: *V-BLAST mode in 802.11n HT-transmitter for three streams.*

Dominant Eigen Mode

The dominant eigen mode scheme uses only one spatial stream but more than one transmit and receive chains. It uses diversity gain to get SNR enhancement. The data is transmitted along the most prominent eigenvalue of the channel matrix.

The eigenvector corresponding to the dominant eigenvalue is used as the precoder for the spatial mapping. The DEM can be thought as an special case of SVD with one spatial stream.

Codebook:

Codebook based precoding is a very flexible mode of MIMO where there is no restriction on the number of streams or tranmit and receive chain. The only constraints are $N_{STS} = N_{SS}$ and $N_{RX} \geq N_{SS}$. Though the scheme in itself is flexible, once decided the codebook must not be changed. The codebook based precoder is helpful in reducing feedback overhead as the receiver can just send back the index of the precoder matrix from codebook that will perform best. An example with $N_{SS} = N_{STS} = 1$, $N_{TX} = 2$, and $N_{RX} = 4$ is shown below.

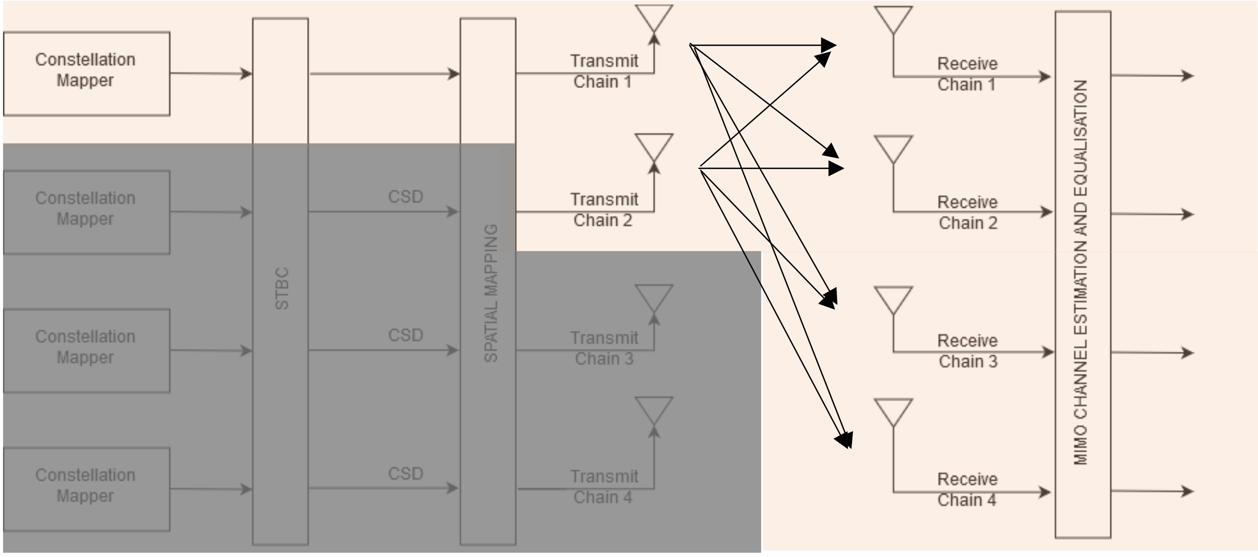


***Fig 11.4:** Precoder matrix based transmission in 802.11n architecture.*

Space Time Block Coding

Space time block coding is different from other schemes discussed here as the number of space time stream is more than the number of spatial streams in this case. The number of transmit chains is equal to space time streams and the number of receive chains is equal to the number of spatial streams. Instead of utilising just the spatial diversity, STBC utilises time along with space to deliver MIMO scheme to receiver with single antenna, without the requirement of feedback. The most popular scheme for STBC is Alamouti that is demonstrated in the figure below.

For systems with multiple carriers, the STBC is not straightforward, as instead of using consecutive samples for STBC, consecutive symbols of waveform are used. So, for each subcarrier, the consecutive data elements are used to get the Alamouti scheme. This is demonstrated in figure 11.6, where x_t^k refers to data at tth time instance and kth subcarrier.

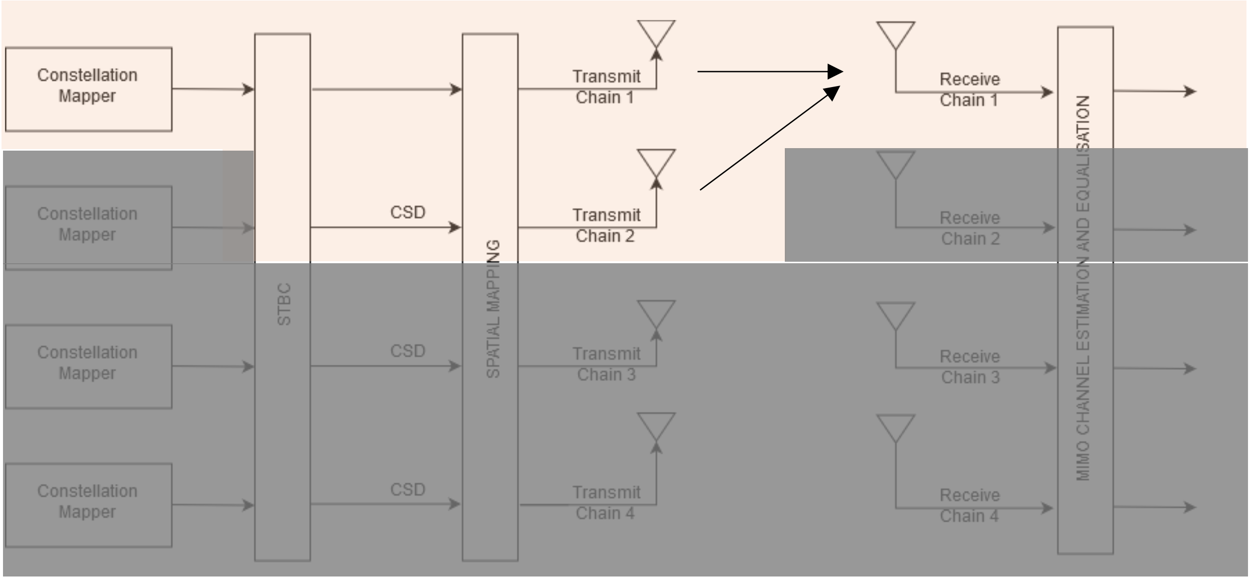
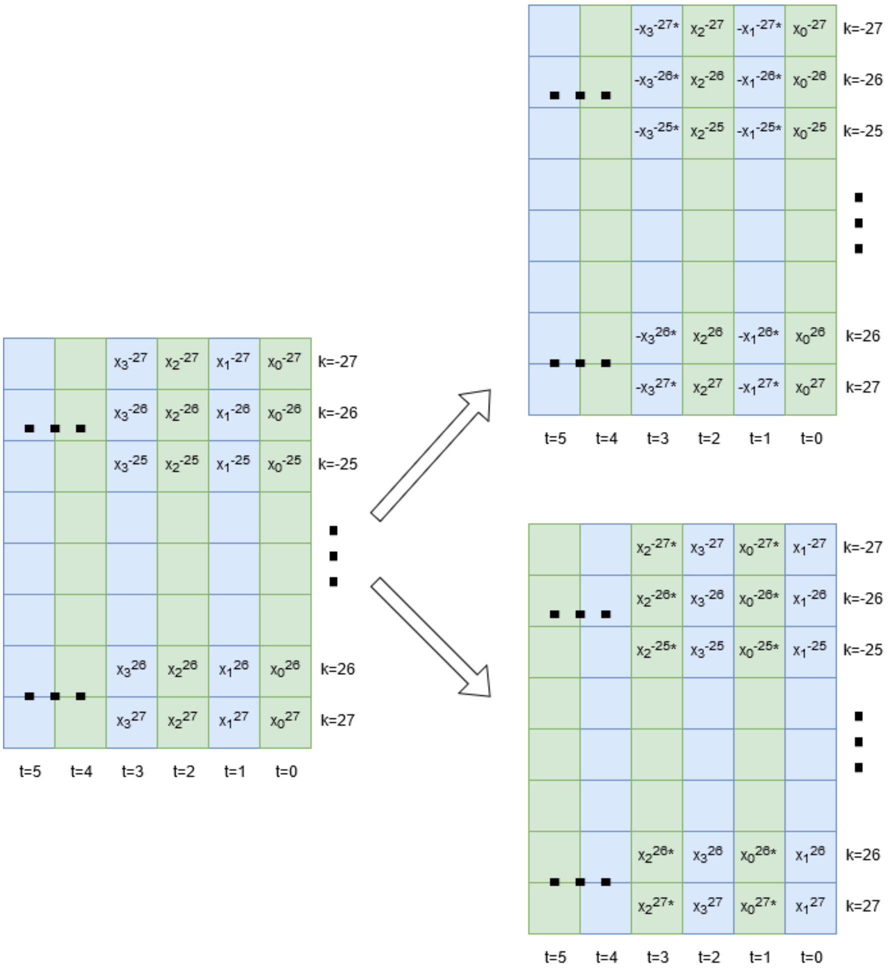


Fig 11.5: *STBC scheme using 802.11n HT-transmitter.*



t=5	t=4	t=3	t=2	t=1	t=0	
		x_3^{-27}	x_2^{-27}	x_1^{-27}	x_0^{-27}	k=-27
		x_3^{-26}	x_2^{-26}	x_1^{-26}	x_0^{-26}	k=-26
		x_3^{-25}	x_2^{-25}	x_1^{-25}	x_0^{-25}	k=-25
		x_3^{26}	x_2^{26}	x_1^{26}	x_0^{26}	k=26
		x_3^{27}	x_2^{27}	x_1^{27}	x_0^{27}	k=27

t=5	t=4	t=3	t=2	t=1	t=0	
		$-x_3^{-27*}$	x_2^{-27}	$-x_1^{-27*}$	x_0^{-27}	k=-27
		$-x_3^{-26*}$	x_2^{-26}	$-x_1^{-26*}$	x_0^{-26}	k=-26
		$-x_3^{-25*}$	x_2^{-25}	$-x_1^{-25*}$	x_0^{-25}	k=-25
		$-x_3^{26*}$	x_2^{26}	$-x_1^{26*}$	x_0^{26}	k=26
		$-x_3^{27*}$	x_2^{27}	$-x_1^{27*}$	x_0^{27}	k=27

t=5	t=4	t=3	t=2	t=1	t=0	
		x_2^{-27*}	x_3^{-27}	x_0^{-27*}	x_1^{-27}	k=-27
		x_2^{-26*}	x_3^{-26}	x_0^{-26*}	x_1^{-26}	k=-26
		x_2^{-25*}	x_3^{-25}	x_0^{-25*}	x_1^{-25}	k=-25
		x_2^{26*}	x_3^{26}	x_0^{26*}	x_1^{26}	k=26
		x_2^{27*}	x_3^{27}	x_0^{27*}	x_1^{27}	k=27

Fig 11.6: *STBC encoding scheme for multiple carrier system.*

In the Alamouti scheme of STBC for multi-carrier we have the following governing equations

$For\ stream\ 1,$

$$y_{2n}^{k} = x_{2n}^{k}$$

$$y_{2n+1}^{k} = -x_{2n+1}^{k\,*}$$

$For\ stream\ 2,$

$$y_{2n}^{k} = x_{2n+1}^{k}$$

$$y_{2n+1}^{k} = -x_{2n}^{k}$$

Where x_t^k refers to data at t^th^ time instance and k^th^ subcarrier.

Singular Value Decomposition

The channel matrix has complex path gains for each transmit-receive pair and hence it is difficult to analyse the channel properties in its raw form. The singular value decomposition of the channel matrix provides an insight into the condition of various possible streams. It provides r streams and their relative strength where r is the rank of the channel matrix.

The singular value decomposition also provides an efficient CSI feedback mechanism. If the V (unitary) is fed back to the transmitter, the transmitter may use it as the precoder matrix. The receiver then uses the S and U matrices known to it for equalisation. The process is demonstrated using the following equations.

The received signal without noise is

$$Y = HX$$

Assuming perfect channel estimation we have $\widehat{H} = H$. Then we perform SVD on H, to get

$$H = USV^H$$

The V matrix is then transmitted back to the transmitter and the transmitter uses it as precoder. So, the transmitter transmits VX instead of X. The received signal in the next iteration becomes

$$Y = HVX$$

The receiver pre-multiples Y with $S^{-1}U^H$ for equalisation to get

$$\hat{X} = S^{-1}U^H HVX$$

$$\Rightarrow \hat{X} = S^{-1}U^H HVX$$

$$\Rightarrow \hat{X} = S^{-1}U^H USV^H VX$$

$$\Rightarrow \hat{X} = S^{-1}ISIX$$

$$\Rightarrow \hat{X} = X$$

The SVD method though resource intensive, gives better performance than other general schemes. The time complexity of SVD is $O(n^3)$, generally and is difficult to implement on hardware, which makes it less a viable option for practical purposes.

The diagram below shows SVD for $N_{SS} = N_{STS} = 2$, $N_{TX} = 3$, and $N_{RX} = 4$.

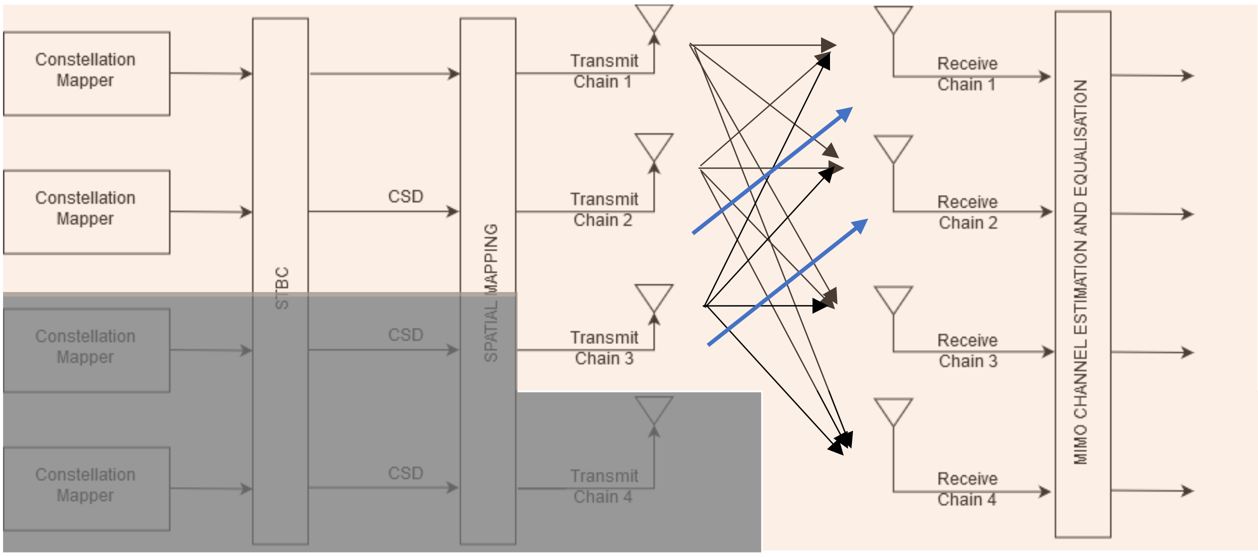


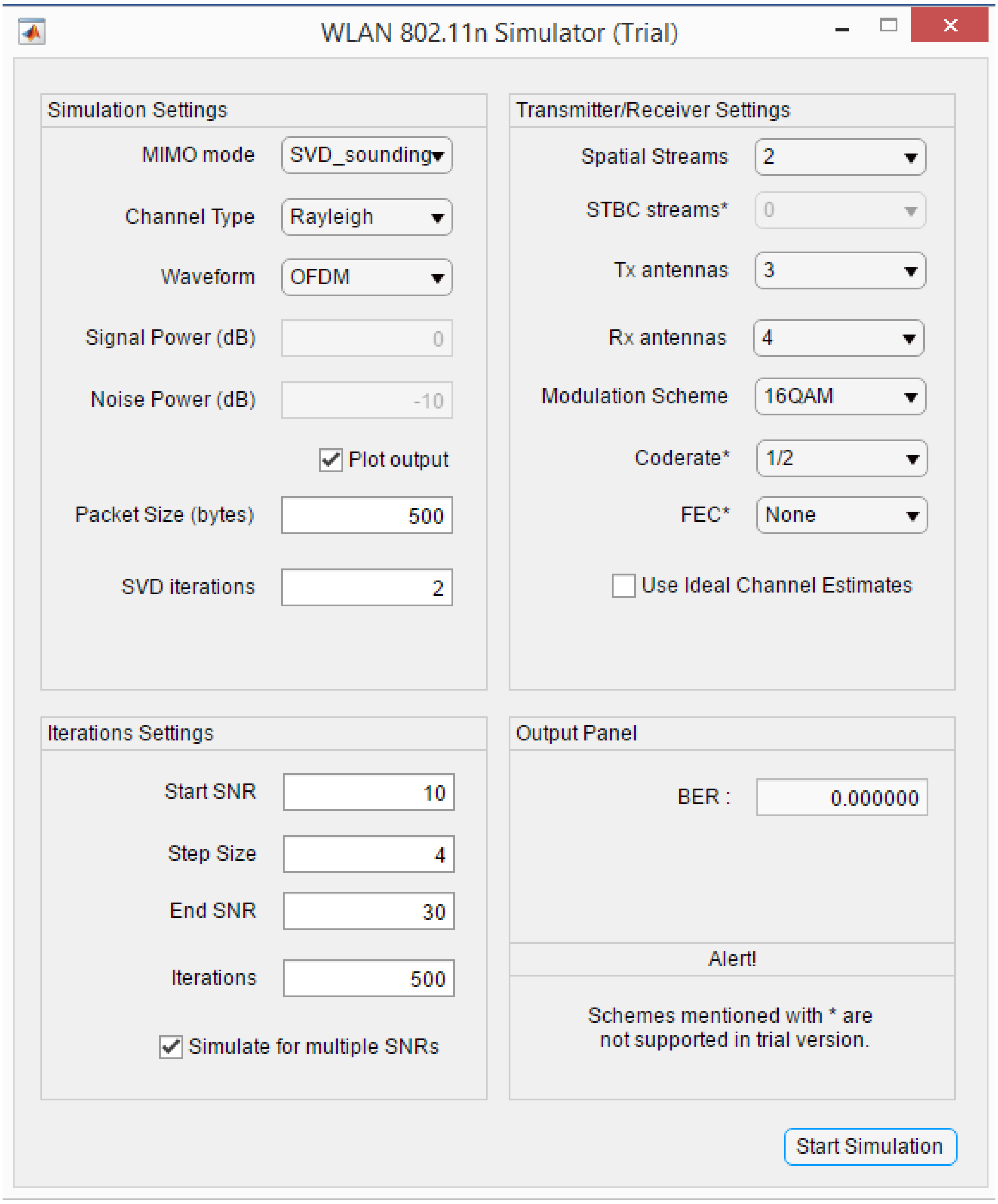
Fig 11.7: Precoder matrix-based transmission in 802.11n architecture.

IEEE 802.11n simulator

A GUI based interface for 802.11n modem was built with end-user as target, instead of telecommunication researchers. The simulator packages all the features mentioned in the report so far, including

- MIMO Modes
- Channel Models
- Waveforms
- Multiple streams
- Multiple transmit chains
- Multiple receive chains
- Constellation Mapping up to 1024QAM
- FEC and BCC with rates $\frac{1}{2}, \frac{2}{3}, \frac{3}{4}, \frac{5}{6}$.

Apart from the above mentioned features the simulator allows user to run multiple iterations at varying SNR to observe BER vs SNR performance and distortion in constellation map with SNR.



WLAN 802.11n Simulator (Trial)

Simulation Settings

MIMO mode: SVD_sounding

Channel Type: Rayleigh

Waveform: OFDM

Signal Power (dB): 0

Noise Power (dB): -10

☑ Plot output

Packet Size (bytes): 500

SVD iterations: 2

Transmitter/Receiver Settings

Spatial Streams: 2

STBC streams*: 0

Tx antennas: 3

Rx antennas: 4

Modulation Scheme: 16QAM

Coderate*: 1/2

FEC*: None

□ Use Ideal Channel Estimates

Iterations Settings

Start SNR: 10

Step Size: 4

End SNR: 30

Iterations: 500

☑ Simulate for multiple SNRs

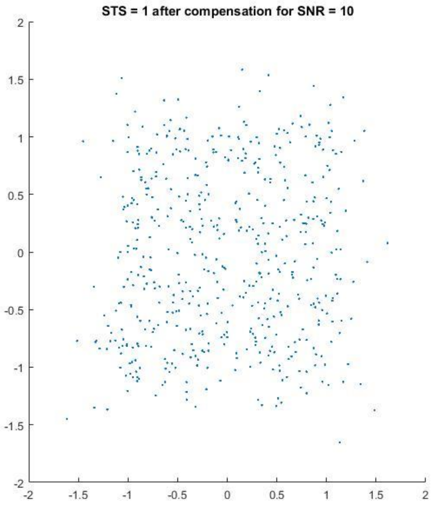
Output Panel

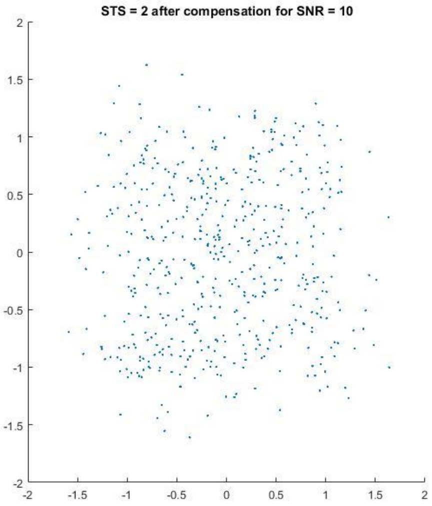
BER : 0.000000

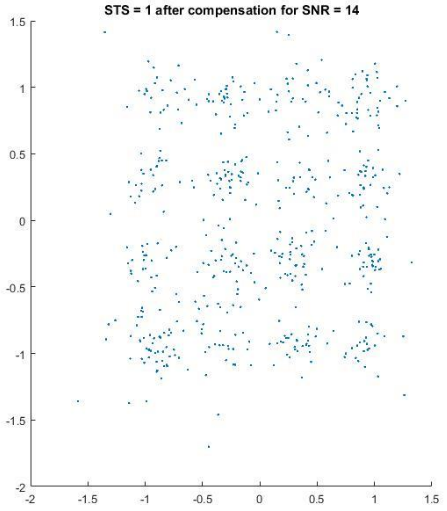
Alert!

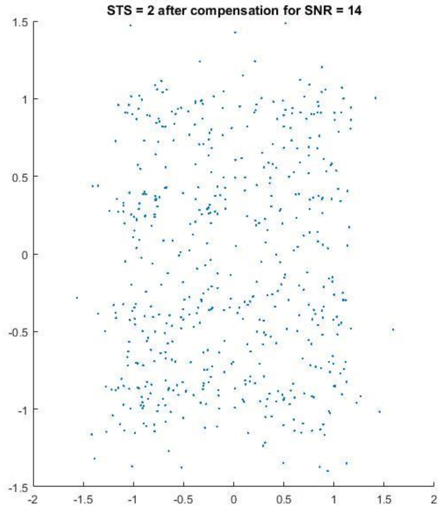
Schemes mentioned with * are not supported in trial version.

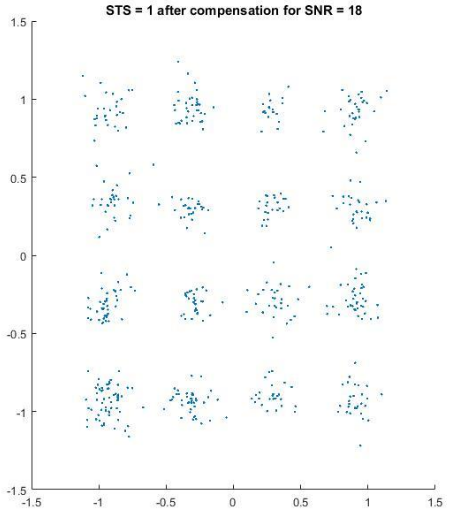
Fig 11.8: *WLAN 802.11n simulator GUI.*

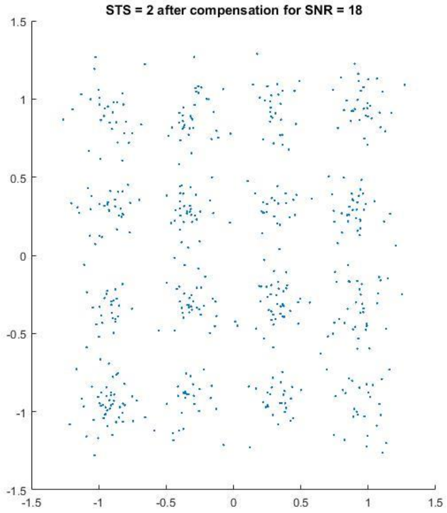

STS = 1 after compensation for SNR = 10


STS = 2 after compensation for SNR = 10


STS = 1 after compensation for SNR = 14


STS = 2 after compensation for SNR = 14


STS = 1 after compensation for SNR = 18


STS = 2 after compensation for SNR = 18

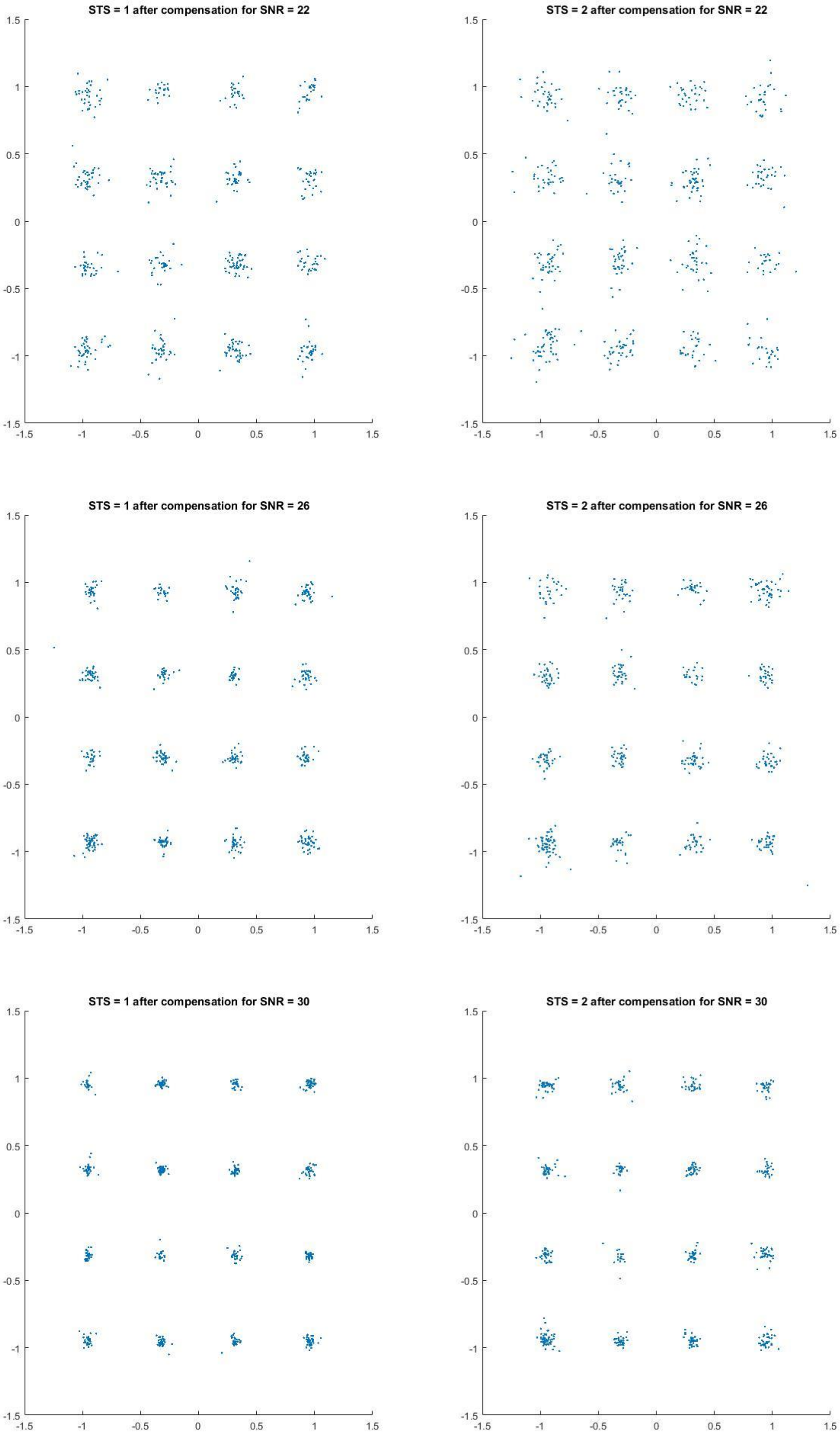


Fig 11.9: *Evolution of constellation diagram with increasing SNR.*

The simulator clearly shows the evolution of constellation diagram as error vector magnitude decreases and the constellation points become less distorted. Apart from the constellation diagram, the simulator also presents bit error rate vs signal to noise ratio curve. The BER vs SNR curve for SVD with 2 streams, 3 transmit chains and 4 receive chains is shown below.

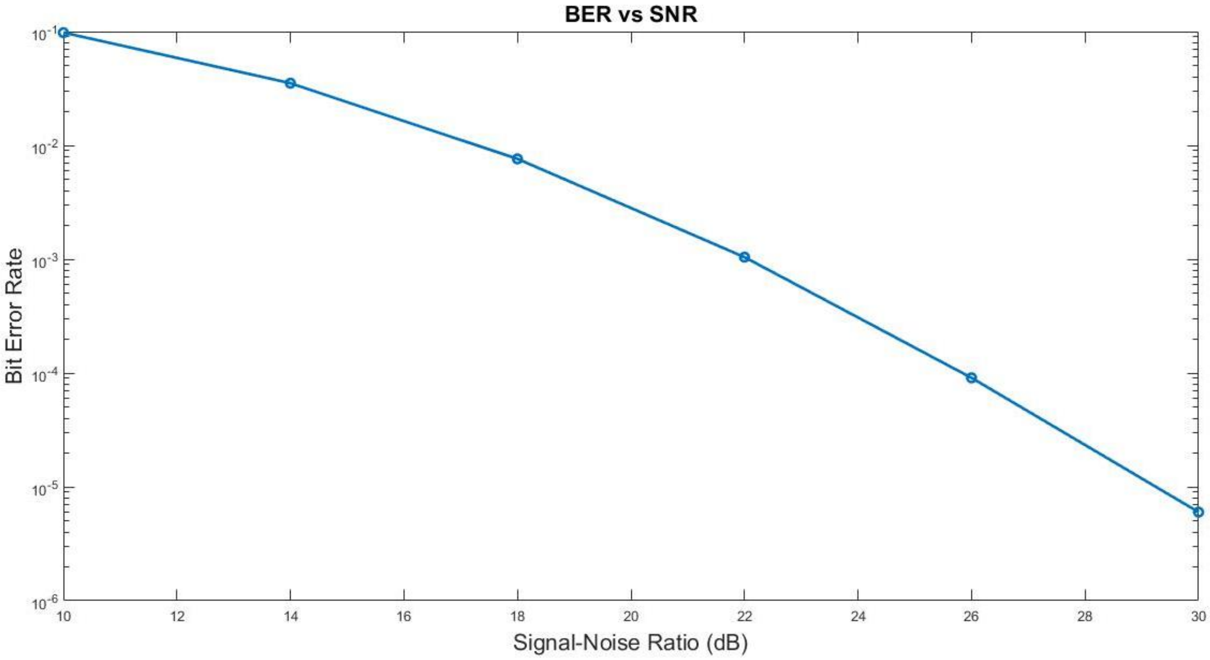


Fig 11.10: *BER vs SNR curve for SVD with N_{SS}=2, N_{TX}=3, and N_{RX}=4 for 16-QAM modulation.*

CHAPTER 12

ARCHITECTURE OF 802.11ac PHYSICAL LAYER

The success of 802.11n, mainly due to the introduction of MIMO was huge and it led to general acceptance of WLAN. One of the biggest advantage with WLAN is the legacy support, which means the WLAN devices continue to support their older versions. Just as 802.11n builds upon the previous versions of 802.11, similarly the 802.11ac builds itself on 802.11n. The 802.11ac furthermore adds MU-MIMO in downlink which adds to the throughput and efficiency of the protocol.

Along with 802.11n transceiver, a transceiver for 802.11ac was studied, implemented and tested for the project. The crucial features of 802.11ac transceiver remain the same as that of 802.11n. The protocol has been highly sanitised by removing redundancies and unpopular features. Apart from MU-MIMO which provides the ability to engage with more than one user at a time to Access Point, the ac series increases the total number of spatial streams and space-time streams to eight from the existing four.

The features added to the 802.11ac can be summarised using the following table.

Feature	802.11n	802.11ac
Bandwidth	20 and 40 MHz only	20, 40, 80, and 160 MHz
Carrier Frequency	2.4 and 5 GHz	5 GHz only
Modulation Schemes	BPSK, QPSK, 16-QAM, and 64-QAM	BPSK, QPSK, 16-QAM, 64-QAM, and 256 QAM
Beamforming Schemes	Explicit and Implicit	NDP explicit beamforming
Spatial Streams	Up to 4	Up to 8, 4 per user
User engagement	Single user	Multi user in downlink

Table 12.1: *Comparison of 802.11ac against 802.11n.*

The 802.11ac adds two new bandwidths 80MHz and 160MHz. The 160MHz bandwidth is supported in either a single bulk processing form or in form of 80MHz + 80MHz chunks processing.

802.11ac supports only 5GHz centre frequency. It adds 256-QAM to the modulation scheme and at the same time, simplifies the MCS table. The 802.11n had 74 MCS entries including the non-uniform distributions. The 802.11ac has only 10 MCS entries with uniform data rate across all spatial streams. The MCS table for 802.11ac is shown below.

MCS	Modulation	Code rate
0	BPSK	1/2
1	QPSK	1/2
2	QPSK	3/4
3	16-QAM	1/2
4	16-QAM	3/4
5	64-QAM	2/3
6	64-QAM	3/4
7	64-QAM	5/6
8	256-QAM	3/4
9	256-QAM	5/6

Table 12.2: *MCS table for 802.11ac.*

Physical Layer Architecture of 802.11ac:

The 802.11ac standards supports its previous version, 802.11a and 802.11n in legacy mode, and hence the architecture of the physical layer for 802.11ac is influenced by mostly 802.11n's (Clause 20) architecture.

The 802.11ac version is called the Very High Throughput version of WLAN and it has its preamble derived from that of 802.11n.

Source: IEEE 802.11ac Part11, Amendment 4, Fig. 22.28, Page 295

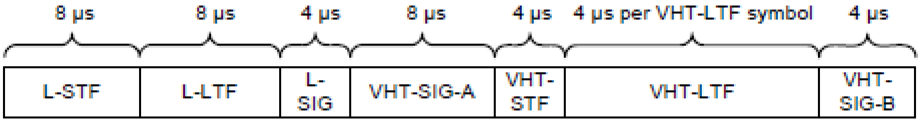


Fig 12.1: *Physical layer frame format for 802.11ac*

The Non-High Throughput and High Throughput PPDU is same as the PPDU of 802.11a and 802.11ac. This is ensured to have legacy support so that devices working on older 802.11a and 802.11n can still communicate with newer devices working on 802.11n protocol.

To generate the L-STF (Legacy Short Training Field), L-LTF (Legacy Long Training Field) and L-SIG (Legacy Signal) and VHT-SIG-A (High Throughput Signal), the transmitter design is as shown below. This transmitter is same as the Transmitter 1 from Chapter 1, Fig. 1.2 with minute changes. Hence, we will call it Transmitter 1 itself. It must be noted that the very high throughput portion of the PPDU is generated using another transmitter than the one given below, hence it is important to distinguish the two transmitters. We shall call the transmitter for the non-HT part as Transmitter 1 and the transmitter for VHT portion of the PPDU, transmitter 3.

Source: IEEE 802.11ac Part11, Amendment 4, Fig. 22.5, Page 231

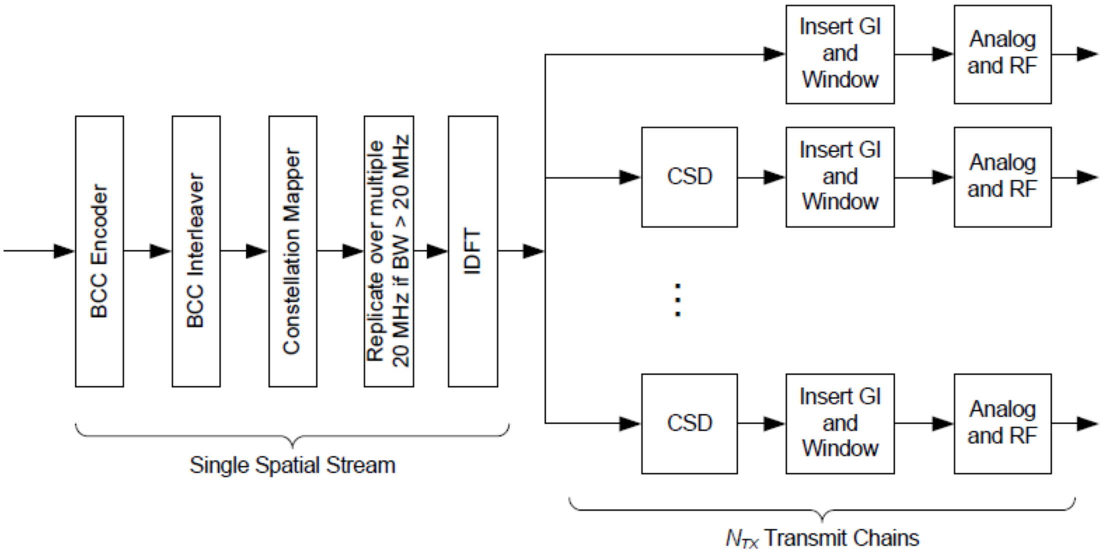


Fig 12.2: *Transmitter 1 (for the non-VHT part)*

For the very high throughput part of the 802.11ac there are multiple transmitter designs depending on the configuration. However, the transmitter 2 shown in Fig. 1.3 essentially covers the features of 802.11ac transmitter for Single User case and it should be used for bandwidth less than or equal to 80MHz. For the multi-user case the transmitter 3 shown below should be used. The building blocks of the transmitter remain the same as that of transmitter 2, however, the packet generation principle changes the transmitter design to support multiple users simultaneously. All the components of transmitter before the spatial mapper, is separated into N_{user} parts that function independent of each other. This ensures that the data intended for a user is not mixed with that for another. At the spatial mapper a precoder is used to beamform the space-time streams towards the respective user antenna.

Source: IEEE 802.11ac Part11, Amendment 4, Fig. 22.12, Page 236

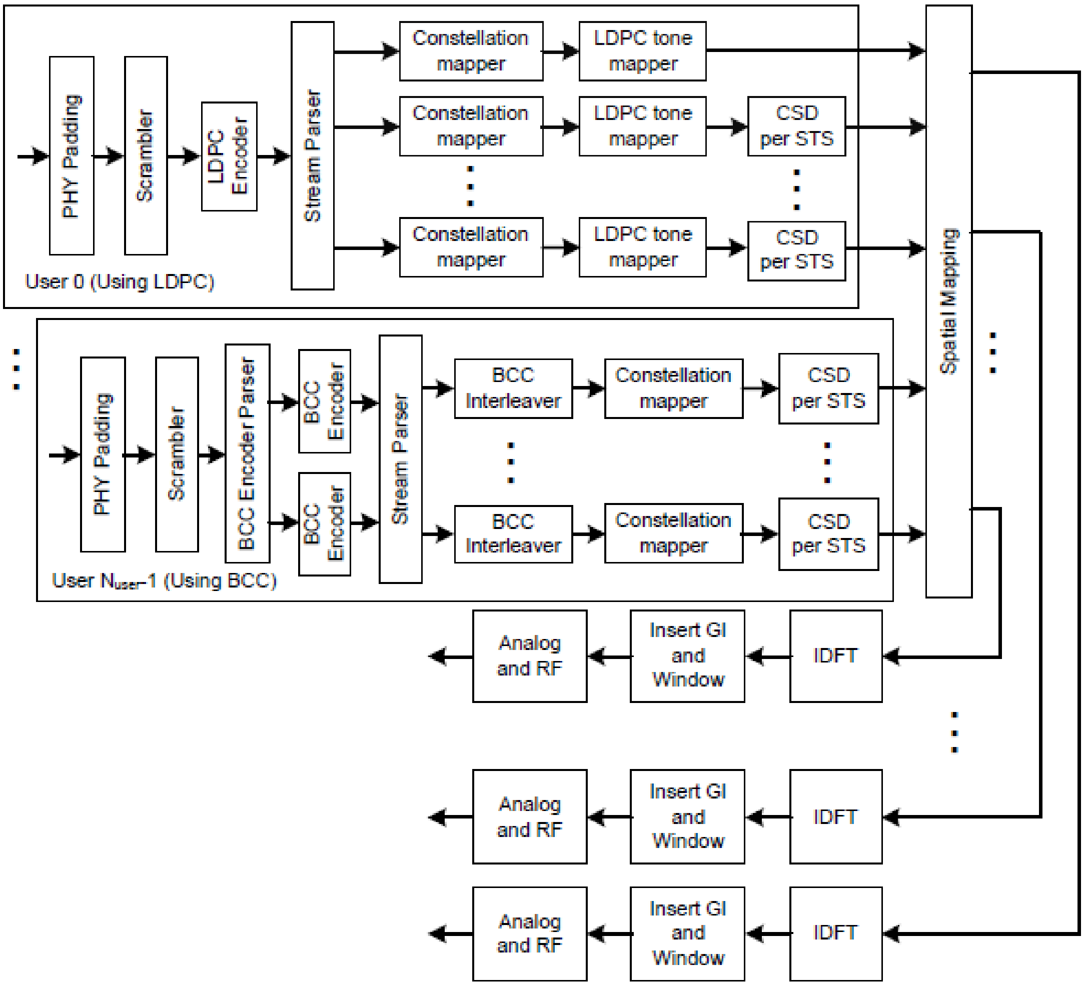


***Fig 12.3:** Transmitter 3 (for multi-user VHT PPDU)*

The values of CSD for cyclic shift delay is different in case of 802.11ac from that in 802.11ac. This is done in order to accommodate more spatial streams. The CSD values can result in fractional sample delay and hence the CSD should now be implemented in the frequency domain.

The table of VHT-LTF required for MIMO channel estimation has been updated and 6 VHT-LTFs are required for 5 or 6 space-time streams, while 8 VHT-LTFs are required for 7 or 8 space-time streams.

The process of scrambling, FEC encoding, constellation mapper, CSD, IDFT, GI and Windowing remain exactly the same as in IEEE 802.11n. However, there is change in the PHY padding and post FEC padding in MU-MIMO transmitter. The length of PPDU is constant for each user, since a single PPDU is transmitted for each user. The data size and modulation scheme differ from user to user and hence the length of usable part from PPDU also differs. To bridge the gap between these differences, PHY padding and post FEC padding is done in a way that the length of PPDU is maintained.

Receiver for 802.11ac:

The receiver for 802.11ac remains the same as the receiver for the 802.11n, as each user is unaffected by the presence of other user as uplink MU-MIMO is not present in 802.11ac. During the downlink, a user can either decode just the space-time streams intended to it or it can go for complete decoding and then discard the irrelevant data.

Decoding the intended space-time stream is secure and suitable way of communication but it requires complicated algorithms to find suitable precoder matrices such that the interference on a user due to presence of data intended for other users is minimal. This is an active field of study and multiple algorithms for minimising the interference have been suggested in literature.

Equalising all the space-time streams is generally not preferred as it leads to security lapses. However, it has the advantage of better performance than former case for the same SNR and interference. The complexity of channel estimators and the equalisers increase in this case though as all the space-time streams (and not just intended STS) need to be equalised.

In the Fig. 12.4 shown below, the AP uses three STS in total. It dedicated two STS to first user that has four receive antenna and one STS to second user that has only two receive chains. In the scheme given below it is not possible to equalise all the space-time streams at user 2, since it can support only two STS due to two transmit chain. However, the user 1 can equalise all the receive chains and benefit from it.

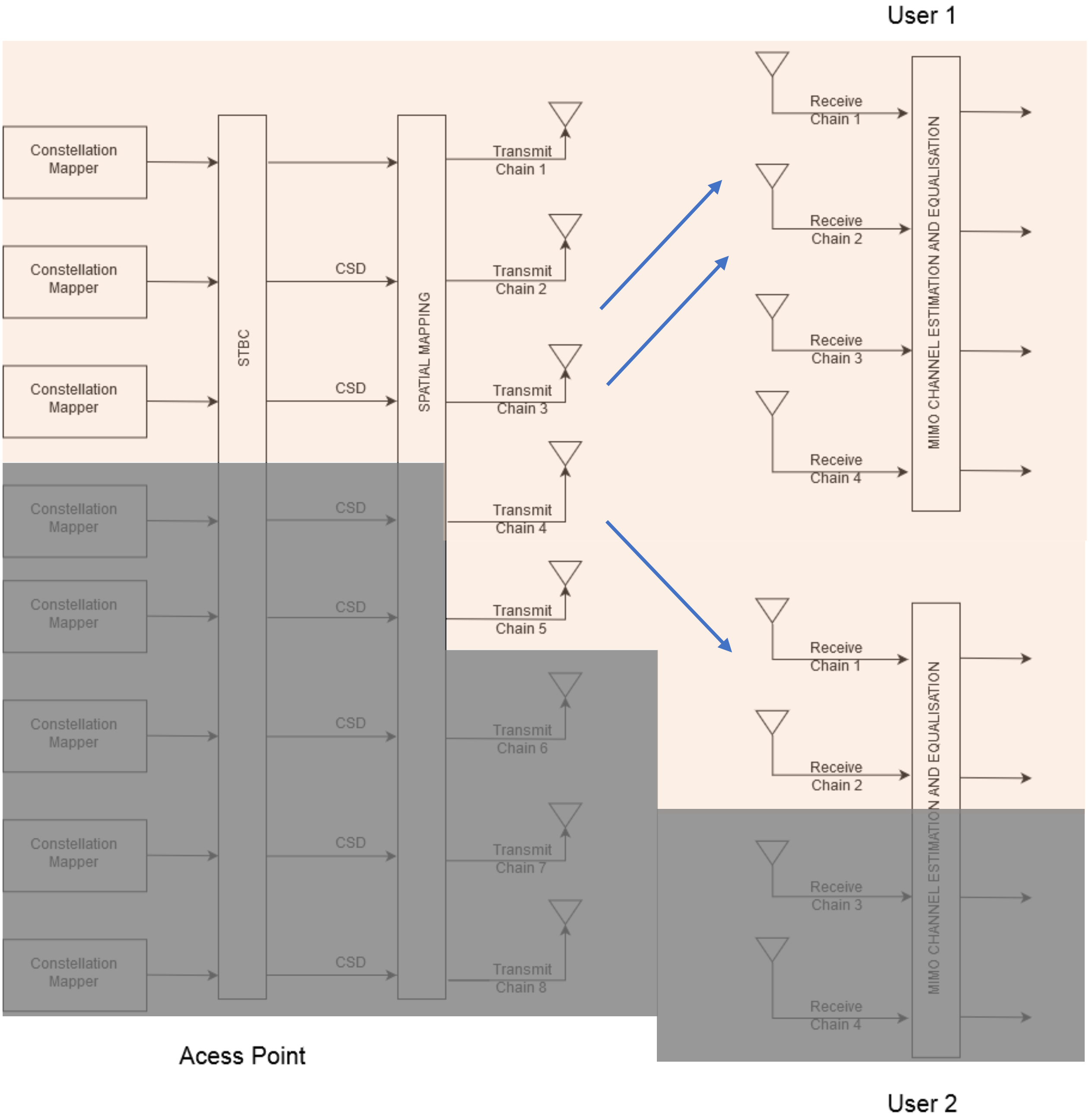


Fig 12.4: *MU-MIMO scenario where AP has 5 transmit chains, first user has 4 receive chains and second user has 2 receive chains.*

CHAPTER 13

MULTI-USER MIMO AND MASSIVE MIMO

The multi-user MIMO inherently undergoes interference due to presence of multiple users. Various algorithms are present in the literature for cancellation of the interference. In this project, two of the algorithms for MU-MIMO were implemented and used. The first algorithm utilises SVD based decomposition for generation of the feedback matrix. The second algorithm uses explicit feedback of the channel information. The feedback matrices are received by the transmitter from all the users and based on the feedback matrices, the transmitter utilises this information to form the precoder matrix that beamforms the space-time stream towards the desired user.

SVD based feedback

The Access Point sounds the channel with Null Data Packet (NDP) for all the users that are about to receive packet from the Access Point. The users estimate the channel using the channel sounding to get an estimate of the MIMO channel. It must be noted that the MIMO channel is formed between the space-time streams designated to the user and the receive chains of the user and not between transmit chain of AP and the receive chains. The user then performs Singular Value Decomposition of the channel received using the steps described in Chapter 12.

Each user then feeds back the V matrix in encoded form using Given's rotation as described in Chapter 6. The AP receives these beamforming weights of Given's rotation from each of the user and uses it form the V matrices fed back by each user. After receiving the V matrices from each user, the AP forms the precoder matrix using the equation given below.

$$P = \begin{bmatrix} V_1 & 0 & . & 0 \\ 0 & V_2 & . & 0 \\ . & . & . & 0 \\ 0 & 0 & 0 & V_{N_{user}} \end{bmatrix}$$

Where V_i denotes the V matrix returned by the ith user. It should be noted that that each of the V matrices are square matrix of dimension $N_{STS,i} \times N_{STS,i}$. Total number of space-time streams is equal to $N_{STS,total} = \sum_i N_{STS,i}$. So the dimension of P matrix is $N_{STS,total} \times N_{STS,total}$, which gets multiplied with the initial precoder matrix Q of dimension $N_{tx} \times N_{STS,total}$. The new precoder matrix Q1 is thud given by the equation

$$Q_1 = QP$$

For SVD based MU-MIMO this precoder matrix is used throughout the transmission of other packets till coherence time (unless the channel has changed significantly).

Channel inversion

The method of channel inversion based MU-MIMO precoding is applicable for single stream and single receive chain users. Like the previous case, the AP transmits an NDP for channel sounding. The receiver uses the NDP to get an estimate of the complete received channel coefficients. The size of this channel estimate matrix for each user is $1 \times N_{STS,total}$.

After estimating the channel, each user feedbacks the channel matrices to the user. The user then stacks these matrices in the form given below and generated a precoder matrix.

$$P = \begin{bmatrix} H_1 \\ H_2 \\ \vdots \\ H_{N_{user}} \end{bmatrix}$$

The P matrix has dimension $N_{STS,total} \times N_{STS,total}$. This matrix is then used to calculate the spatial mapping matrix using the equation given below.

$$Q_1 = QP^{-1}$$

The performance of channel inversion based MIMO equalisation is much better than the SVD based feedback mechanism demonstrated above. This is due to the fact that the explicit feedback uses information related to all the channels possible for finding the precoder, while SVD based feedback mechanism drops the information of other user channels.

Bit error rate vs Signal to Noise Ratio with channel inversion for number of transmit chain at AP being 4 and two users, each with one stream and one receive chain is given below.

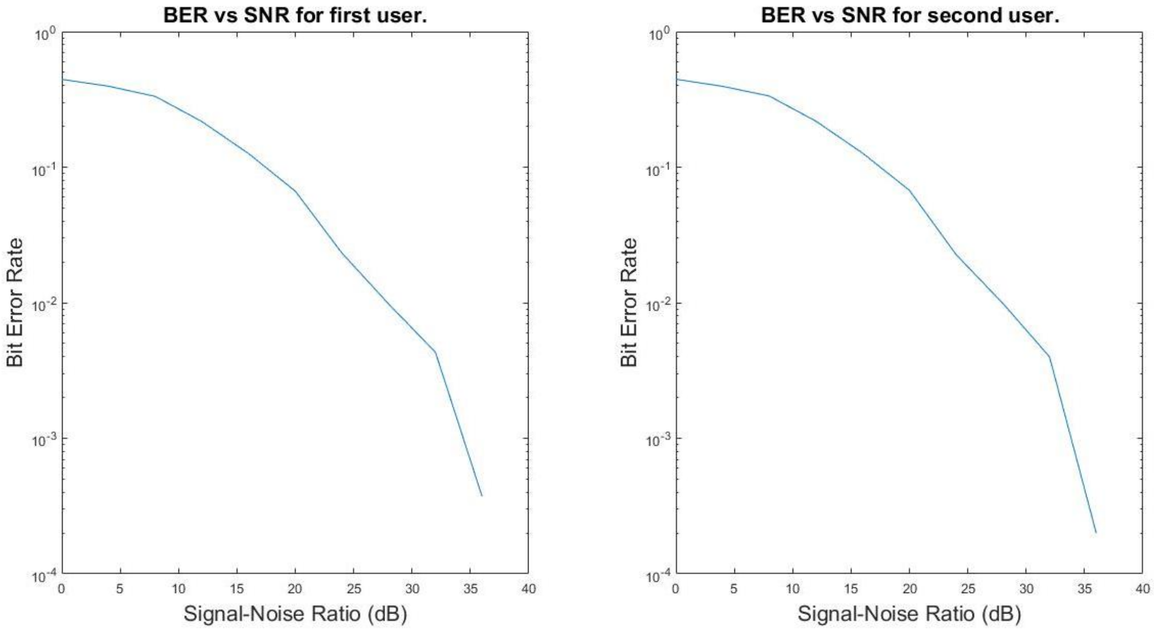


Fig 13.1: *SNR vs BER curve using QPSK for MU-MIMO system with two users.*

System equation for 802.11 based architecture.

In this part we discuss the equations related to the transmitter and receiver. The equations are formed with constellation mapper being the source of data. It should be noted that the equations do not take the blocks before constellation mapper into account and neither do they take the blocks after constellation mapper on the receiver side into account. The system model also doesn’t take into account the STBC scheme. To find the system model of STBC enabled transceiver, the data symbols (x) should be treated as the output of STBC encoder instead of constellation mapper.

Let there be N$_{\text{ofdm}}$ total OFDM symbols. Each of these symbols have 64 subcarriers. Out of them 52 carry data and 4 carry pilots. Others are zero valued. We denote subcarriers by k, then output of constellation mapper can be written as

$$x^{k}_{i_{STS},\ n_{ofdm}}$$

where $-28 \leq k \leq 28$,
$i_{STS} \in \{1,\ 2,\ ..\ N_{STS}\}$ is the number of space time streams in consideration, and $1 \leq n_{ofdm} \leq N_{ofdm}$. Here n$_{\text{ofdm}}$ denotes the OFDM symbol to which $x^{k}_{i_{ss},\ n_{ofdm}}$ belongs and provides the notion of time.

Then for a given n_{ofdm}, we concatenate $x^{k}_{i_{ss},\ n_{ofdm}}$ along i_{ss} to form a vector and further concatenate those vectors along k to form a matrix (denoted by X) that depends on n_{ofdm} (OFDM symbol).

$$X_{nofdm} = \begin{bmatrix} x^{-28}_{1,nofdm} & x^{-27}_{1,nofdm} & . & x^{+28}_{1,nofdm} \\ x^{-28}_{2,nofdm} & x^{-27}_{2,nofdm} & . & x^{+28}_{2,nofdm} \\ . & . & . & . \\ . & . & . & . \\ x^{-28}_{Nss,nofdm} & x^{-27}_{Nss,nofdm} & . & x^{+28}_{Nss\,nofdm} \end{bmatrix}$$

Each element of the $X_{n_{ofdm}}$ matrix is multiplied by a factor $c(i_{ss},\ k)$ where c is a function of the spatial stream to which it is getting multiplied and the subcarrier index. The output of the CSD block is denoted by $\hat{X}$. Then

$$\hat{x}^{k}_{i_{ss},\ n_{ofdm}} = x^{k}_{i_{ss},\ n_{ofdm}}.\exp(-j.2\pi.k.\Delta f.T^{i_{ss}}_{csd})$$

Where T^{iss}_{csd} = 0nS, -400nS, -200nS, and -600nS for i_{ss}= 1, 2, 3, and 4 respectively and $\Delta f = 312.5kHz$.

we denote the IFFT matrix (with GI) by W and the spatial mapping matrix by P, then the transmitted signal S can be given by

$$S_{nofdm} = P.\hat{X}_{nofdm}.W$$

Here P is of dimension $N_{tx} \times N_{sts}$
$\hat{X}_{nofdm}$ is of dimension $N_{sts} \times N_{ifft}$
W is of dimension $N_{ifft} \times (N_{ifft} + N_{gi})$
Hence, the size of S_{nofdm} is $N_{tx} \times (N_{ifft} + N_{gi})$ for each ofdm symbol.

The signal S is transmitted through a MIMO channel and it is received with by the receiver side where it undergoes waveform demodulation. After the proper removal of guard interval, windowing and waveform demodulation, the received signal can be written as

$$Y^{k}_{n_{ofdm}} = H^{(k)}.P.\hat{X}^{k}_{nofdm}$$

Where $\hat{X}^{k}_{nofdm}\ denotes\ the\ kth\ column\ \hat{X}_{nofdm}$ (the k^th subcarrier).

The $\hat{X}^{k}_{nofdm}$ can now be written as product of $C^{k}\ and\ X^{k}_{n_{ofdm}}$ where $X^{k}_{n_{ofdm}}$ denotes the kth column of X_{nofdm} matrix.

C^{k} is a diagonal matrix with elements
[$\exp(-j.2\pi.k.\Delta f.T^{1}_{csd})$, $\exp(-j.2\pi.k.\Delta f.T^{2}_{csd})$, ... $\exp(-j.2\pi.k.\Delta f.T^{Nss}_{csd})$]

So, we have

$$Y^k_{n_{ofdm}} = H^{(k)}.P.C^k.X^k_{nofdm}$$

The architecture of 802.11 encourages one to do channel estimation end to end using the formula given below

$$H^{(k)}.P.C^k = H^k_{eff}$$

And thus, the received signal on the effective channel forms a direct multiplication of matrices. As the received signal is in linear form, a large set of linear processing algorithms become applicable and the can be utilised for equalisation.

The equivalent form of the received signal is

$$Y^k_{n_{ofdm}} = H^k_{eff}.X^k_{nofdm}$$

Massive MIMO

A massive MIMO system uses large amount of antenna and hence transmit chain for beamforming than normal MIMO and MU-MIMO systems. This leads to huge diversity gain that is exploited to get better data rate and performance. T is a promising field that might play a crucial role in fulfilling the ever-increasing demand for the high-performance wireless communication network. A drawback of the massive MIMO is the huge channel state information feedback associated with it.

With large antenna arrays the typical signal processing techniques (e.g. maximum likelihood detection) get very complex because of large signal dimensions. The key question here is whether significant multiplexing gain be obtained with low-complexity and low-cost signal processing technique. Thomas Marzetta proved that by having a very large number of antennas at BS in comparison to the number of users being served the simple linear processing performs near optimal. Even the simple maximal ratio combining, the effects of fast fading, inter-cell interference, and uncorrelated noise, almost vanishes as the number of antennas at BS station is increased. MIMO systems, where the BS equipped with hundreds of antennas serves tens or hundreds of users simultaneously in same time-frequency block, also known as massive MIMO systems (very large MU-MIMO, hyper-MIMO, or full-dimension MIMO systems).

A modem based on 802.11 system was developed that supports massive MIMO along with MU-MIMO. The maximum number of transmitters supported is in hundreds for all practical purposes.

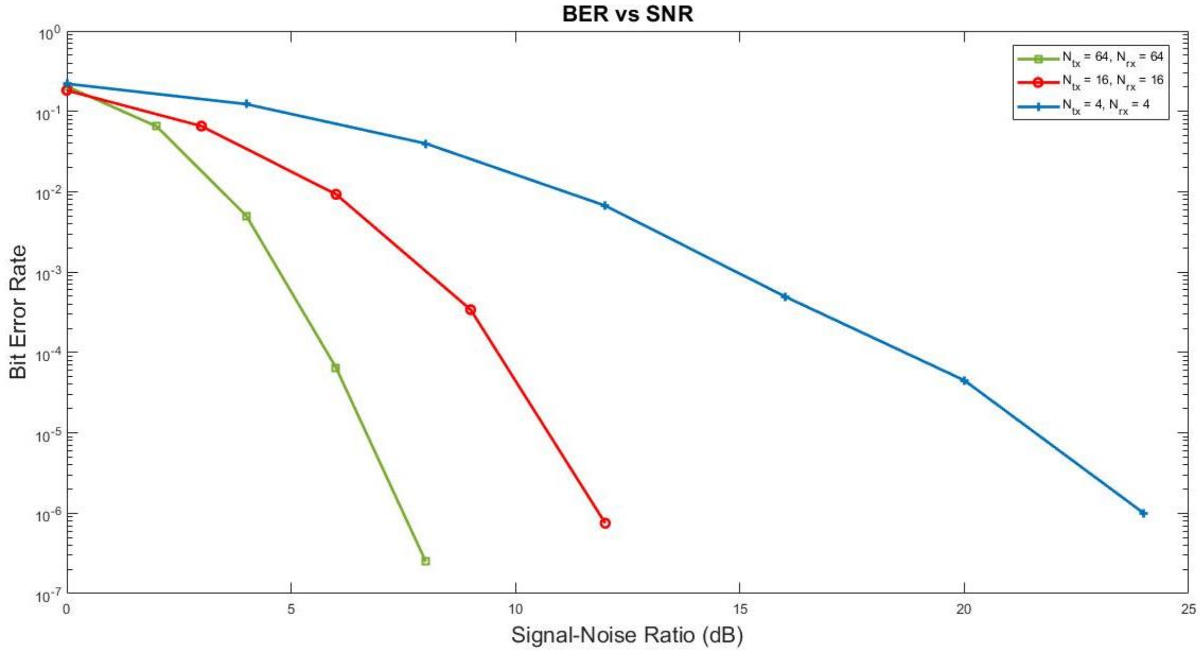


Fig 13.2: *SNR vs BER curve using 16-QAM for massive-MIMO system.*

CHAPTER 14

CONCLUSION

A good grasp of IEEE 802.11n and IEEE 802.11ac architecture was achieved as a part of Master Term Project. In the first part of MTP, Low Density Parity Codes and its decoder was implemented and it was found to be at par with the known results. Theoretical work was carried forward with the study and development of double sliding window energy-based packet detection. A theoretical derivation for receiver operating characteristic was undertaken and the simulation results were found to match the theoretical formulae developed thus verifying the work. Furthermore, an algorithm for the implementation of the double sliding window-based packet detection was proposed.

In the second half of the MTP transmitters and channel models, receivers along with their algorithms were realized for both 802.11n and 802.11ac as a part of this thesis. The preamble of the PHY layer PPDU was generated for SISO and MIMO, and MU-MIMO case. The transmitter consisting of LDPC(FEC), symbol mapper, STBC, CSD, MIMO encoder, OFDM, and windowing was implemented. The corresponding receiver with packet detection, synchronization, channel estimation and equalization, CSI feedback, MU-MIMO and massive-MIMO algorithms was also developed. With minimal changes, it is also possible to reconfigure the modem for other purposes too. The effect of Rayleigh and 802.11n TGn proposed MIMO channel models were studied as a part of the project.

The modem developed was tested with over the air setup using WARPLab. PPDUs were successfully modulated, transmitted, received and decoded for OTA testing. The modem has also been converted into a simulator targeted for academicians and students interested in learning wireless communication. The simulator was demonstrated with multiple MIMO modes as specified in 802.11n.

Various parts of the simulator can be used independently or in integration and the immediate future target would be to extend the simulator to encapsulate features of IEEE 802.11ax. However, the work done so far is conclusive and the simulator can be packaged and made ready for distribution if needed.

References:

1. IEEE P802.11n, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Enhancements for Higher Throughput. 2009

2. Eldad Perahia & Robert Stacey, "Next Generation Wireless LANs. Throughput, Robustness and Reliability in 802.11n", Cambridge University Press, 2008.

3. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Enhancements for Very High Throughput for Operation in Bands below 6GHz, IEEE P802.11ac/D1.0 Std., Jan. 2011

4. R. G. Gallager, "Low-density parity-check code," IRE Trans. Inform. Theory, vol. IT-8, pp. 21-28, Jan. 1962.

5. S. M. Alamouti, "A Simple Transmit Diversity Technique for Wireless Communications", IEEE JSAC, vol. 16, no. 8, Oct. 1998, pp.1451-58

6. A.A.M. Saleh and R.A. Valenzuela, "A statistical model for indoor multipath propagation," *IEEE J. Select. Areas Commun.*, vol. 5, 1987, pp. 128-137.

7. J. Heiskala & J. Terry, "OFDM Wireless LANs: A Theoretical and Practical Guide", 2nd ed. Sams Publishing, July 2001.

8. M. I. Rahman, Suvra Sekhar Das, Frank H.P Fitzek, "OFDM Based WLAN Systems", Technical Report R-04-1002; v1.2, ISBN 87-90834-43-7, ISSN 0908-1224, Aalborg University, Denmark , February 2005.

9. T. Schmidl and D. Cox, "Robust frequency and timing synchronization for OFDM," IEEE Trans. Commun., vol. 45, pp. 1613{1621, Dec. 997.

10. Steven M. Kay, "Fundamentals of Statistical Signal Processing", Volume II, Indian Edition, Pearson Publications.

11. Joshi Shailesh A, 14EE10023,"WLAN 802.11 Frame Detector Design", Electrical Engineering Department, Work done during the 8th semester B.Tech project in the spring semester 2018 under the guidance of Prof. Suvra Sekhar Das.

12. White Paper: LTE Release 12 "LTE Transmission Modes and Beamforming"

13. G. J. Foschini, “Layered space-time architecture for wireless communication in a fading environment when using multi-element antennas,” Bell labs technical journal, vol. 1, no. 2, pp. 41–59, 1996.

14. G. J. Foschini and M. J. Gans, “On limits of wireless communications in a fading environment when using multiple antennas,” Wireless personal communications, vol. 6, no. 3, pp. 311–335, 1998.

15. D. Nguyen, L.-N. Tran, P. Pirinen, and M. Latva-aho, “Transmission strategies for full duplex multiuser mimo systems,” in Communications (ICC), 2012 IEEE International Conference on. IEEE, 2012, pp. 6825–6829.

16. T. L. Marzetta, “Noncooperative cellular wireless with unlimited numbers of base station antennas,”IEEE Transactions on Wireless Communications, vol. 9, no. 11, pp. 3590–3600, 2010.

17. F. Boccardi, R. W. Heath, A. Lozano, T. L. Marzetta, and P. Popovski, “Five disruptive technology directions for 5g,”IEEE Communications Magazine, vol. 52, no. 2, pp. 74–80, 2014.

18. E. G. Larsson, O. Edfors, F. Tufvesson, and T. L. Marzetta, Massive mimo for next generation wireless systems,” IEEE Communications Magazine, vol. 52, no. 2, pp. 186–195, 2014

19. Hoefel, R.P.F.. (2012). IEEE 802.11n: On the Performance of Channel Estimation Schemes over OFDM MIMO Spatially- Correlated Frequency Selective Fading TGn Channels.

20. Q. H. Spencer, C. B. Peel, A. L. Swindlehurst and M. Haardt, "An introduction to the multi-user MIMO downlink," in IEEE Communications Magazine, vol. 42, no. 10, pp. 60-67, Oct. 2004.

21. A. Goldsmith, "Capacity Limits of MIMO Channels", *IEEE JSAC*, vol. 21, pp. 684-702, June 2003.

22. H. Weingarten, Y. Steinberg, S. Shamai, "The Capacity Region of the Gaussian MIMO Broadcast Channel", *Proc. Conf. Info. Sciences and Systems (CISS)*, 2004-Mar.

23. Tomoki Maruko, Takahiro Yamaguchi, Tomoki Yoshimura, Hiromichi Tomeba, Takashi Onodera, Fumiaki Maehara, "Efficient combination of multi-user MIMO THP and user selection based on spatial orthogonality", *Wireless Communications and Networking Conference (WCNC) 2016 IEEE*, pp. 1-5, 2016.

24. Israfil Bahceci, Mehedi Hasan, Bedri A. Cetiner, "Downlink Multi-User MIMO Transmission for Reconfigurable Antenna Systems", *Wireless Communications and Networking Conference (WCNC) 2017 IEEE*, pp. 1-6, 2017.

25. Bin Li, Yi Luo, Hui Shen, "A new precoding scheme using interference alignment on modulation signal for multi-user MIMO downlink", *Wireless Communications & Signal Processing 2009. WCSP 2009. International Conference on*, pp. 1-5, 2009.

26. Ruixue Liu, Zhao Li, Liyuan Xiao, "Proactive CCI evaluation and transmission gain based scheduling for MU-MIMO broadcast channel", *Wireless Communications & Signal Processing (WCSP) 2013 International Conference on*, pp. 1-6, 2013.

27. M. P. C. Fossorier, M. Mihaljevic, and H. Imai, “Reduced complexity iterative decoding of low density parity check codes based on belief propagation,” IEEE Trans. Commun., vol. 47, pp. 673-680, May 1999.

28. E. Eleftheriou, T. Mittelholzer and A. Dholakia, “Reduced-complexity decoding algorithm for low-density parity-check codes,” IEE Electronics Letters, vol. 37, pp. 102-104, Jan. 2001.

29. D. J. C. MacKay, “Good error-correcting codes based on very sparse matrices,” IEEE Trans. Inform. Theory, vol. 45, pp. 399-431, Mar. 1999.

30. S.-Y. Chung, G. D. Forney, Jr., T. J. Richardson, and R. Urbanke, “On the design of low-density parity-check codes within 0.0045 dB of the Shannon limit,” IEEE Commun. Lett., vol. 5, pp. 58-60, Feb. 2001.

31. J. Hagenauer, E. Offer, and L. Papke, “Iterative decoding of binary block and convolutional codes,” IEEE Trans. Inform. Theory, vol. 42, pp. 429-445, Mar. 1996.

32. P. Robertson, E. Villebrun, and P. Hoeher, “A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain,” Proc. Intl. Commun. Conf. ’95, pp. 1009-1013, June 1995.

33. D.J.C. MacKay, “Online database of low-density parity-check codes,” available at http://wol.ra.phy.cam.uk/mackay/codes/data.html

34. Seho Myung, Kyeongcheol Yang and Jaeyoel Kim, "Quasi-cyclic LDPC codes for fast encoding," in IEEE Transactions on Information Theory, vol. 51, no. 8, pp. 2894-2901, Aug. 2005.

35. Amr M. Otefa, Namat M. ElBoghdadly, and Essam A. Sourour “Performance Analysis of 802.11n Wireless LAN Physical Layer”, Published in: Information and Communications Technology, 2007. ICICT 2007. ITI 5th International Conference on Information and Communications Technology

36. F. Jameel, Faisal, M. A. A. Haider and A. A. Butt, "Massive MIMO: A survey of recent advances, research issues and future directions," *2017 International Symposium on Recent Advances in Electrical Engineering (RAEE)*, Islamabad, 2017, pp. 1-6.