MASTER OF SCIENCE IN ELECTRICAL AND ELECTRONIC ENGINEERING

Performance Evaluation and Enhancement of IEEE 802.11 WLAN over Multipath Fading Channels in GNU Radio and USRP Platform

Department of Electrical and Electronic Engineering
Islamic University of Technology (IUT)
Board Bazar, Gazipur-1704, Bangladesh.
October, 2018

Performance Evaluation and Enhancement of IEEE 802.11 WLAN over Multipath Fading Channels in GNU Radio and USRP Platform

by

Muhammad Morshed Alam

MASTER OF SCIENCE

IN

ELECTRICAL AND ELECTRONIC ENGINEERING

Department of Electrical and Electronic Engineering
Islamic University of Technology (IUT)
Board Bazar, Gazipur- 1704, Bangladesh.
October, 2018

CERTIFICATE OF APPROVAL

The thesis titled “**Performance Evaluation and Enhancement of IEEE 802.11 WLAN over Multipath Fading Channels in GNU Radio and USRP Platform**” submitted by Muhammad Morshed Alam, St. No. 142612 of Academic Year 2014-2015 has been found as satisfactory and accepted as partial fulfillment of the requirement for the Degree of MASTER OF SCIENCE in ELECTRICAL AND ELECTRONIC ENGINEERING on 25 October, 2018.

Board of Examiners:

1.

____ Chairman

Prof. Dr. Mohammad Rakibul Islam (Supervisor)
Professor,
Electrical and Electronic Engineering Department,
Islamic University of Technology (IUT), Gazipur.

2.

____

Prof. Dr. Md. Ashraful Hoque
Professor and Head, Member (Ex-officio)
Electrical and Electronic Engineering Department,
Islamic University of Technology (IUT), Gazipur.

3.

____

Prof. Dr. Syed Iftekhar Ali
Professor, Member
Electrical and Electronic Engineering Department,
Islamic University of Technology (IUT), Gazipur.

4.

____

Prof. Dr. Mohammed Imamul Hassan Bhuiyan Member (External)
Professor,
Electrical and Electronic Engineering Department,
Bangladesh University of Engineering and Technology (BUET)

DECLARATION OF CANDIDATE

It is hereby declared that this thesis report or any part of it has not been submitted elsewhere for the award of any Degree or Diploma.

Prof. Dr. Mohammad Rakibul Islam (Supervisor)
Professor
Electrical and Electronic Engineering Department,
Islamic University of Technology (IUT), Gazipur.

Muhammad Morshed Alam
Student No.: 142612
Academic Year: 2014-2015
Date: 25 October, 2018

Dedicated to my parents

ACKNOWLEDGEMENT

A deep gratitude to Dr. Mohammad Rakibul Islam, Professor, Department of Electrical and Electronic Engineering, Islamic University of Technology (IUT), Gazipur, for giving me an opportunity to develop this M.Sc. Engg. thesis under his supervision. Without his unwavering support and crucial insights, publishing the research contained in this thesis would have been impossible. The enthusiasm he effortlessly carries, even into his lectures, has been a constant source of motivation for me from the time I started my journey as master’s student, and for this I am forever grateful.

I am indebted to Dr. Md. Ashraful Hoque, Professor and Head, EEE Department, IUT for his valuable support throughout the two years of my M.Sc. Engg. course.

I am grateful to Dr. Syed Iftekhar Ali, Professor, EEE Department, IUT for his priceless counsel on my academics.

I am thankful to Dr. Mohammed Imamul Hassan Bhuiyan, Professor, EEE Department, BUET, for his precious advice on my research work.

I am incredibly grateful to Dr. Feroz Ahmed, Associate Professor, Independent University of Bangladesh (IUB), for giving me the opportunity to work at Telecommunication Laboratory of IUB and also for his valuable advice and continuous help regarding my thesis.

I would like to thank all the faculty members and staff of the department of EEE, IUT for their inspiration and help. I would like to thank my parents for their blessings and my family for having faith in me throughout the times of this thesis.

Finally, I would like to express my unparalleled gratitude to Almighty Allah for his divine blessing without which it would not have been possible to complete this thesis successfully.

LIST OF ABBREVIATION OF TECHNICAL TERMS

OFDM	Orthogonal Frequency Division Multiplexing
SDR	Software Define Radio
USRP	Universal Software Radio Peripheral
FDM	Frequency Division Multiplexing
FER	Frame Error Rate
BER	Bit Error Rate
AGWN	Additive White Gaussian Noise
WLAN	Wireless Local Area Network
ISI	Inter Symbol Interference
ICI	Inter Carrier Interference
LS	Least Square
LMS	Least Mean Square
GRC	GNU Radio Companion
WiFi PHY HIER	Wireless Fidelity of Physical Hierarchy
CP	Cyclic Prefix
QAM	Quadrature Amplitude Modulation
BPSK	Binary Phase Shift Keying
QPSK	Quadrature Phase Shift Keying
GI	Guard Interval
LOS	Line of Sight
NLOS	No Line of Sight
PDP	Power Delay Profile

SNR	Signal to Noise Ratio
BER	Bit Error Rate
FFT	Fast Fourier Transform
IFFT	Inverse Fast Fourier Transform
CE	Channel Estimation
LPI	Low-pass Filter Interpolation
FPGA	Field Programmable Gate Array
GPP	General Purpose Processor
DSP	Digital Signal Processor
PHY	Physical Layer
MAC	Media Access Control Layer
IDFT	Inverse Discrete Fourier Transform
DFT	Discrete Fourier Transform
FIR	Finite Impulse Response
LTE	Long Term Evolution
MIMO	Multiple Input and Multiple Output
PC	Pilot Carrier
SC	Sub Carrier
PMT	Polymorphic Type Message
PDF	Probability Density Function
PPDU	Physical Layer Protocol Data Unit
PLCP	Physical Layer Convergence Protocol
SWIG	Simplified Wrapper and Interface Generator
XML	Extensible Markup Language

ABSTRACT

This thesis focuses on the evaluation and enhancement of the Frame Error Rate (FER) performance of IEEE 802.11 a/g/p standard 5 GHz frequency band WLAN over Rayleigh and Rician distributed fading channels in the presence of Additive White Gaussian Noise (AWGN). Orthogonal Frequency Division Multiplexing (OFDM) based transceiver is implemented by using real-time signal processing frameworks (IEEE 802.11 Blocks) in GNU Radio Companion (GRC) and Ettus USRP N200 is used to process the symbol over the wireless radio channel. The Frame Error Rate (FER) is calculated for each sub-carrier conventional modulation schemes used by OFDM such as BPSK, QPSK, 16, 64-QAM with different punctuated coding rates. More precise SNR is computed by modifying the SNR calculation process of YANS and NIST error rate model to estimate more accurate FER. Here, real-time signal constellations, OFDM signal spectrums etc. are also observed to find the effect of multipath propagation of signals through flat and frequency selective fading channels. To reduce the error rate due to the multipath fading effect and Doppler shifting, channel estimation (CE) and equalization techniques such as Least Square (LS) and training based adaptive Least Mean Square (LMS) algorithm are applied in the receiver. The simulation work is practically verified at GRC by turning into a pair of Software Define Radio (SDR) as a simultaneous transceiver. The applied Low Pass Interpolation (LPI) technique based LS channel estimator gives better equalization of the transmitted symbols for the ideal static AWGN channel model as per the FER curve response for the SNR range from 5 to 25 dB. On the other hand, the similar estimator in the mobility condition considering the relative speed $v = 84\ m/s$ with associated normalized Doppler shift $f_d T_s = 0.0133$ in fading channel models, the transceiver prototype has no significant improvement in FER curve as this estimator has no adaptability with rapidly changing channel condition for similar values of SNR. It has been observed that after applying the proposed novel adaptive LMS channel estimation technique considering the different values of relative speed $v\ \epsilon\ \{84, 120, 150\}\ m/s$, normalized Doppler shift $f_d T_s \epsilon\ \{0.0133,\ 0.0188,\ 0.0235\}$, chosen modulation scheme 64-QAM (2/3) with 24 Mbps data rate the FER at receiver significantly decreased almost tends to zero while the SNR approached 25 dB for both Rayleigh (NLOS) and Rician (LOS) distributed fast time-varying wireless fading channel models.

Table of Contents

List of Figures

List of Tables

CHAPTER 1

INTRODUCTION

Software Define Radio (SDR) is a platform where the majority of the physical layer signal processing is done at software by using Field Programmable Gate Array (FPGA), General Purpose Processor (GPP) and Digital Signal Processor (DSP) instead of using complicated electrical hardware [1]-[3]. GNU Radio is an open source widely used software define radio platforms [3] which has digital signal processing blocks such as signal sources, sinks, filters, modulators, demodulators, encoders, decoders, equalizers, synchronizers and much more [6], [26]. The critical signal processing blocks are developed in C++ language [1]; however basic flow graphs are generated in python language [8]. Using this software user can build their own applications to transmit or receive data with the help of Universal Software Radio Peripheral (USRP) devices associated Radio Frequency (RF) TX/RX ports [1], [6].

OFDM is widely used in almost all wireless communication standards such as cellular standards like LTE advance and WiMAX, digital broadcasting standards known as DVB-T, IEEE 802.11 WLAN standards etc. [5], [9]. In OFDM, parallel high rate data symbols are loaded on comparatively low rate several numbers of closely spaced orthogonal sub-carriers [4]-[6] having safety guard interval between two adjacent symbols and each sub-carrier are performed with conventional digital modulation schemes such as BPSK, M-PSK, M-QAM [10] etc. For the precise symbol duration, the sub-carriers are independent to each other. As a result, for a particular center frequency of the symbol, only the corresponding sub-carriers have the peak power while others are zero [7], [19]. It provides higher resistance to multipath effects and gives high spectral efficiency [5]. IEEE 802.11 is basically set of a physical layer (PHY) and Media Access Control (MAC) layer specifications for wireless communication in 2.4, 5 and 60 GHz frequency bands [3], [18]. There are several 802.11 WLAN standards are existing among them few are given at Table 1.1.

Table 1.1 IEEE 802.11 PHY Layer standards [23]

Wireless LAN standard	**Radio Frequency Band (GHz)**	**Modulation**	**Bandwidth (MHz)**	**Highest Conceivable Data Rate (Mbps)**
IEEE 802.11 a	5	OFDM	20	54
IEEE 802.11 g	2.4	OFDM	20	54
IEEE 802.11 p	5	OFDM	10	54
IEEE 802.11 ac	5	MIMO-OFDM	20, 40, 80, 160	700-800

In wireless communication, multipath fading effect causes time and frequency dispersion of wireless radio signals as the mobile devices are not stationary [6]. These effects have the impact on the FER of any modulation schemes as mobile radio channels cause the significant fade or distort of the transmitted signal with compared to AWGN channel [9], [11]. Channel Estimation (CE) and equalization techniques can be applied to reduce the FER [8] and Inter Symbol Interference (ISI) created by multipath fading effects [13]-[14]. For a particular channel, ISI is introduced as the modulation bandwidth exceeds the coherence bandwidth due to the pulse broadening in time domain [12], [17]. As the mobile fading channel characteristics are random and time varying nature [11], the equalizers must have the adaptive characteristics [9] to track with the time varying components [14], [17]. At the beginning, a fixed length predefined training symbols are periodically sent by the transmitter that is known to the receiver so that the estimator can adapt to a minimum FER detection function and have the idea of the channel impulse response behavior caused by multipath [15]. Then the user's data symbols are transmitted following by the periodic training symbols [17] since the equalizer has the adaptive characteristics [16] it utilizes the adaptive Least Mean Square (LMS) technique to estimate the desired filter coefficients in order to compensate the distortion created by the multipath [3]. These predefined tanning symbols help the receiver end estimator to update with the suitable filter coefficients on worse case channel condition or even on rapidly time varying channel condition [19]-[20]. When the training sequences are finished the filter, coefficients are updated with nearly optimum values through a continuous consequential adaptive algorithm process to get the convergence of the estimator.

1.1 Motivation of Work

A growing number of present as well as future telecommunication systems deploy variants of OFDM as the transmission technology at the physical layer [5]. OFDM is a non-trivial frequency-division multiplexing scheme wherein the frequency band is split into several narrowband sub-carriers that overlap with each over [16]. Nevertheless, the use of harmonic functions for the sub-carriers establish an orthogonal basis that can be exploited in a transmission scenario. There is a general commonality with respect to the future expectations of telecommunications systems:

Dynamism: Mobility requirements are continuously increasing. Telecommunications systems where potentially both the transmitter and the receiver are in relative motion like vehicle to vehicle communication, mobile reception of terrestrial TV like Digital Video Broadcasting – Second Generation Terrestrial (DVB-T2) lead to requirement of coping with relative speeds of up to several 100 km/h [21].

High Frequency: Frequencies are a scarce resource and communication systems for the future have to work at higher carrier frequencies of several GHz up to several 10 GHz. Moreover, the 5th generation of wireless systems (5G) proposes a spectrum that aggregates frequency bands up to 86 GHz [21].

Sub- Carrier Spacing: The evolution of signal processing capabilities has led to a mass market application for multicarrier systems with up to 32k carriers while the channel raster has remained unchanged. This has resulted in a continuous decreasing carrier spacing [12].

A trend can thus be seen where telecommunication systems have to work with higher frequencies in a highly mobile environment with an ever-increasing number of sub-carriers. Although the use of OFDM offers several advantages, a Doppler shift caused due to the mobility affects the OFDM signal by destroying the orthogonality of the narrowband sub-carriers [6]. Thus, adjacent sub-carriers begin to interfere with each other causing ICI. Moreover, the use of higher frequencies along with a smaller sub-carrier spacing proportionally increases the effects of Doppler shifts at the receiver [5]. Thus, a robust and low-complexity channel estimation scheme plays an important role in overcoming the effects of a doubly selective multi-path fading channel and ensuring reliable performance of the receiver [9].

Employing an OFDM system in the PHY is addressing a limited part of the problem. Reliable communication between vehicles need to compensate for the highly dramatic environment [5]. A suite of protocols at each layer of the Open System Interconnection (OSI) need to be defined that will facilitate not only the communication between vehicles, but also between the vehicles and other network elements [15]. One such suite is defined by the Institute of Electrical and Electronics Engineers (IEEE) organization. IEEE 1609 is a family of standards for Wireless Access in Vehicular Environments (WAVE) [6]. It makes use of the IEEE 802.11p, which is a Medium Access Control Layer (MAC) and PHY specifications amendment to the IEEE 80211 standard [6]. Although the IEEE 802.11p is derived from the IEEE 802.11a, the letter is mainly restricted to operate in an indoor Environment [11].

Traditional channel estimation schemes designed for the IEEE 802.11a, like the Least Squares (LS) [12], work well under indoor environments but perform poorly under high Doppler shift environment [9]. But for highly time varying and Doppler shifting multi-path environment the traditional channel estimations scheme gives catastrophic performance in the receiver [5]. As a result, a channel adaptive estimator should have to design to reconstruct the data byte steams with minimum error rate in the receiver for worse channel condition [1].

1.2 Related Research

The Intelligent Transport Systems (ITS) makes use of V2V communication to allow automobiles to communicate with each other. The European Union (EU) has allocated the 30 MHz bandwidth from 5.875 GHz to 5.90 GHz for transport safety ITS applications in the year 2008 [17]. Although this frequency band is under contention, a lot of automotive manufactures are actively equipping their cars with this system. The standard of communication for ITS in both Europe and America is based on IEEE 802.11p as the MAC and PHY layer [19]. The IEEE 802.11 standards suite was originally specified for Wireless Local Area Networks (WLAN) commination [15].

In previous, researcher designing the IEEE 802.11 transceiver frame work in GNU Radio platform, but the testing is very limited to the low complex channel changing condition [2]. In vehicular environment the channel time varying condition, multi-path delay, fading effects and Doppler shift due to the relative motion between the transceiver is robust [5], [6]. Hence, the receiver in IEEE 802.11p should be more intelligent and channel estimator has to be designed in such way that it performs with low mean square error (MSE) [1]. The NIST and YANS error rate models are well established model to calculate the frame error rate model in IEEE 802.11 standard transceiver [18]. Here in this thesis, more precise SNR estimation is applied to calculate the BER and FER of the designed transceiver considering the reduction of energy due to the cyclic prefix, reduction of the net energy due to the pilot carriers and the ratio of used sub-carriers in the OFDM system.

The main goal of this thesis is to provide cognitive frame work to enable adequate channel estimation and equalization in all mobility conditions. This goal is achieved and will be proved through simulations. Along with this research also has to be performed on hardware applicability. In that case to implement this algorithm the USRP could provide an intermediate step in this direction on GNU Radio platform. The information of the delays and its corresponding coefficients could also be transferred back to the time domain that can be used to reconstruct a filter for each OFDM symbol, which might provide the better equalization.

1.3 Channel Estimation and Equalization

To support vehicular network the IEEE 802.11 standard was extended with a new operational mode that allowed intermediate communication without prior connection setup and an adapted OFDM based physical layer, which is very similar to IEEE 802.11a but all the timings are doubled. The change transforms the 20 MHz signal of IEEE 802.11a to the 10 MHz signal proposed for automotive applications. In a nutshell, the switch to 10 MHz made the signal more robust against delay spread but, at the same time, more sensitive to Doppler and time variability of the channel.

The foundations of OFDM was laid down by the honored paper of Shannon, where it was suggested that higher data-rate can be achieved by using a multi-carrier system [5]. Orthogonal multiplexing was then used to show transmission of data messages simultaneously through a linear band-limited transmission medium at a maximum data rate without inter-channel and inter-symbol interference [14]. The capacity of such a multi-tone channel was evaluated and shown that it is far better than a single tone-channel. There are lot of advantages of OFDM which makes it the preferred choice for the PHY layer, and mostly used in every wireless standard [15].

The OFDM signal undergoes adverse effects, like fading effects, when transmitted through the wireless channel. There are also certain practical impairments that are introduced, like timing offsets, in a communication system [5]. A protocol that is designed to compensate for these penalties is typically employed, and for this thesis, that protocol is the IEEE 802.11p. Synchronization is an important and necessary step in the receiver to coarsely correct the timing and frequency offsets [1]. Using a set of known sub-carriers to transmit information provides much better synchronization, in a multi-path environment, compared to the synchronization based only on the cyclic prefix [1]. The IEEE 802.11 wireless standard makes use of a particular structure, based on Pseudo-Noise sequences with good correlation properties, called as Short Preamble to achieve robust synchronization [1].

The synchronization scheme achieves coarse correction of frequency and timing offsets [2]. But for finer correction, channel estimation schemes are employed that make use of the Long Preamble [1]. Also, the channel might introduce inter-symbol and inter-carrier interference which need to be corrected before data can be decoded. The method involves transmitting known data on defined sub-carriers to estimate the multiplicative channel coefficients. Although the long preamble provides a defined set of values and positions in the frequency domain, its position in time domain is also defined, and due to IFFT being a linear function, its values in the time domain is also defined. This enables us to employ channel estimation in both domains. The LS channel estimator is a basic estimator to find the channel coefficients in the frequency domain. The computational complexity involved is minimal and the performance is also overly restricted especially applicable for indoor environment in static AGWN channel [1]. A comparison of different channel estimation schemes, based on performance and complexity, for IEEE 802.11p, shows, that schemes need more prior channel information to ensure preferable performance.

At the beginning of the MAC frame a fixed length predefined training symbols, known as Short Preamble and Long Preamble are periodically sent by the transmitter [1]. These preambles are known to the receiver. The receiver estimator compared the error rate by deriving a Mean Square Error (MSE) function and applied the iteration based adaptive minimization technique to get the Minimum Mean Square Error (MMSE) [1]. This training symbol based iterative process known as adaptive Least Mean Square algorithm filter. The LMS filter adaptively updates its filter coefficients using the pilot signal positions as long the minimization process is completed and the equalizer get the convergence.

1.4 Simulation Platform

Defined physical layer frames conforming to the specifications of the IEEE 802.11p standard need to be transmitted through a doubly selective channel [5]. These frames must then be forwarded to the different channel estimation and channel equalization blocks to understand their respective performance [1]. A simulation platform conforming to the specifications along-with the ability to simulate doubly selective channel are the essential ingredients required to verify the different schemes in this thesis. The GNU Radio Companion simulation platform is used based on the MAC and PHY layer specifications of the IEEE 802.11p to design transmitter and receiver and its performance has been evaluated [3]. The PHY layer specifications has also been flashed onto an FPGA and the performance has been assessed in. A lot of research in different platforms has already been performed on the simulation of IEEE 802.11p PHY layer. But, none of these endeavors have led to creating a publicly available simulation platform that can be incorporated readily to test the different schemes. However, A GRC model of IEEE 802.11 a/g/p frame transceiver can be designed using IEEE 802.11 a/g/p blocks that channel parameters can be altered and performing frame error rate computations at the same time. Although the simulation platform is available in MATLAB Simulink platform but performing the various types of channel estimation and equalization technique is a laborious process there.

GNU Radio is well known powerful simulation platform which is completely an open source Software Defined Radio (SDR) [26]. An SDR marks a paradigm shift towards accomplishing major part of functionality in software rather than the hardware design. A fundamental challenge of SDR is to provide the necessary computational capacity to process the waveform applications, in particular the complex and high data rate waveform especially for units with strict power and size limitations [6]. The open source signal processing platform, GNU Radio, in this regard has many contributions that make it easier to rapidly prototype the designed algorithms, test their performance and also estimate their complexity. The GNU Radio receiver supporting OFDM conforming to the IEEE 802.11 a/g/p standard was developed to serve as a basis of further experiments, measurements, and research on signal processing algorithms. An entire transceiver stack for IEEE 802.11p was constructed, tested extensively and was found to be in accordance to the well accepted models of Network Simulartor-3 (NS-3) [13]. This thesis also proven that the simulations with the USRP devices are in close relation to commercially available IEEE 802.11p devices.

1.5 Thesis Outline

This thesis report is prepared from the analysis of IEEE 802.11 PHY and MAC layer standard. According to the theoretical concepts the transceiver modeled in GNU Radio Companion (GRC) and USRP devices are used for practical implementations. Chapter 2 is accentuated on basics of OFDM, its channel response and multi-path influence on OFDM systems using some mathematical equations. It also contains the advantages and dis-advantages of adapting OFDM system. Following that chapter 3 describes the Physical and MAC layer standard of IEEE 802.11 and it also describes the transmitter and receiver structures in GRC. Different types of channels and their characteristics are interpreted in chapter 4. This chapter describes the channel modeling considering the multi-path fading environment and Doppler shifts in vehicular environment for mobile wireless channel condition.

The next chapter 5, describes the more precise FER computing process for IEEE 802.11 standard by modifying the SNR estimation process of established state-of-art model NIST and YANS for IEEE 802.11 wireless standard. In this chapter, FER vs. SNR curves are observed for different values of channel condition and types such as static AWGN, Rayleigh (NLOS) and Rician (LOS) distributed mobile wireless channel model. Here, symbol constellations and OFDM signal spectrums are observed for 16-QAM modulation scheme of symbol mapping in real-imaginary plane. This chapter also describes the channel estimation techniques for minimizing the Mean Square Error (MSE) and enhancing the receiver performance. This chapter illustrates the LS and training symbol based adaptive LMS channel estimator working algorithm for IEEE 802.11 frame equalization with lowest error rate. Here, FER performance is compared for both equalized and non-equalized IEEE 802.11 transceiver in multi-path faded and Doppler shifted environment considering the different values of SNR.

CHAPTER 2

OVERVIEW OF OFDM, GNU RADIO AND SDR

2.1 Overview of Orthogonal Frequency Division Multiplexing (OFDM)

Due to high spectral efficiency and robust performance at multipath fading channel OFDM is widely used in wireless communication. Let us consider, $x_1,\ x_2, \dots\dots,\ x_N$ are the data symbols in the frequency domain. These data symbols are transmitted to the IDFT operation after inserting the OFDM pilot symbols. Therefore, for the k^{th} data symbol the corresponding time domain data signal $x(n)$ is given below [1], [4]:

$$x(n) = \sum_{k=0}^{N-1} x(k) e^{\frac{j2\pi kn}{N}}, \qquad 0 \le n \le N-1 \tag{2.1}$$

Here, N is the total number of sub-carriers. $\sum_{n=1}^{N} x(n)^k\ (x(n)^j)$ *=0, here k $\neq$ j and * denotes the complex conjugate of the j^{th} sample. This equation confirms the orthogonal property of N OFDM sub-carriers to avoid Inter Carrier Interference (ICI) between them. The guard interval is provided between each time domain data symbols to avoid the ISI in OFDM system. Let the time domain samples $x(n)$ turn into $x_G(n)$ after the insertion of the guard interval and that should be transmitted over the wireless channel. The resulting transmitted symbol is:

$$x_G(n) = \begin{cases} x(N+n), & n = -N_G, -N_G+1, \dots\dots, -1 \\ x(n), & n = 0, 1, \dots\dots, N-1 \end{cases} \tag{2.2}$$

Here, N_G is the number samples in the guard interval. The transmitted signal will propagate through the frequency selective multipath fading wireless channel. Therefore, transmitted symbols can be modulated as a circular convolution between channel impulse response and transmitted data symbols. Let the receiver got the received signal as $y_G(n)$ [4] expressed as given below:

$$y_G(n) = x_G(n) * h(n,\tau) + w(n) = \int h(n,\tau) x_G(n-\tau) d\tau + w(n) \tag{2.3}$$

Here, $h(n,\tau)$ is the impulse response of the fading channel, $w(n)$ is the additive white Gaussian noise. Channel impulse response $h(n,\tau)$ consisting of L paths can be expressed as given below [1]:

$$h(n,\tau) = \sum_{j=0}^{L-1} h_j e^{j2\pi f_{Dj} t\frac{n}{N}} \delta(n-\tau_j)\,, 0 \le n \le N-1 = \sum_{j=0}^{L-1} a_n\, \delta\big(n-\tau_j\big) \qquad (2.4)$$

Here, L is the total number of possible propagation paths, f_{Dj} is the j^{th} path Doppler frequency shift which creates ICI of the received signals, h_j is the complex impulse response of the j^{th} path and τ_j is the j^{th} path time delay and a_n is the complex path attenuation factor which gives the time varying complex amplitude, defined as the exponential term $a_n = h_j e^{j2\pi f_{Dj} t\frac{n}{N}}$. Channel Doppler shift is chosen as per the jakes spectrum or a flat spectrum, $[-f_{Dmax}, f_{Dmax}]$, here maximum Doppler frequency $f_D = \frac{v f_c}{c}$. Normalized Doppler frequency is defined as $f_D T_s$, where T_s is the symbol period. We assume that the delay spreads are shorter than the cyclic prefix which allows reducing ISI between OFDM symbols.

After excluding the safety guard interval from $y_G(n)$, the received signal turns into $y(n)$ which are sent for the DFT operation to de multiplex the multi carrier signals and to get back again frequency domain data samples which have been transmitted over the fading channel [1], [11].

$$Y(k) = \left[\frac{1}{N}\sum_{n=0}^{N-1} y(n) e^{\frac{-j2\pi kn}{N}}\right], 0 \le k \le N-1 \qquad (2.5)$$

Finally, at the receiver end output the de multiplexed signal can be expressed as given below:

$$Y(k) = X(k)H(k) + W(k), \qquad 0 \le k \le N-1 \qquad (2.6)$$

Here $X(k)$ is the transmitted signal, $H(k)$ is the channel transfer function represented in the frequency domain and $W(k)$ is the FFT of Gaussian noise $w(n)$.

Advantages of OFDM is much more complicated compared to traditional multiplexing schemes, but it does prove definite gains, in terms of data transmission, over the traditional schemes.

- Typically, transmission specifications are severely stringent on the bandwidth and amount of power that can be transmitted by a system. OFDM is one of the best methods to exploit this limited spectrum by using the narrow-band carries [5].
- The input data stream in OFDM is converted to a lower bit rate parallel steam, which increase the symbol duration, in turn making the OFDM system more resistant to multi-path effects [23].
- The lower bit-rate symbols also result in lower ISI. The use of cyclic prefix in OFDM contributes to making the entire OFDM symbol more robust against the multi-path effects.
- Integrated Circuit (IC) for IFFT and FFT are fast, reliable and economical. This makes the OFDM implementation in a communication system exceptionally competitive even in the commercial market [16].

Although there are many advantages to using an OFDM scheme, there are few disadvantages that also have to be taken in consideration before selecting it as the medium of transmission.

- Carrier frequency offsets and drifts can be dangerous to an OFDM signal. The orthogonality between the sub-carriers are lost due to these frequency shifts. Doppler effects result in frequency shifts, and these shifts are not a constant value for all frequencies [8].
- OFDM has a high Peak to Average Power Ratio (PAPR). The lower bit-rate combined with the cyclic prefix and high spectral efficiency usually means that the PAPR will be large. This puts higher demands on the amplifiers in the system [16].

In conclusion, OFDM offers several advantages but the major drawback is its susceptibility to Doppler shifts that destroy the orthogonality of the narrowband sub-carriers due to ICI. High mobility in a multipath propagation environment leads to a doubly selective or a time-varying multipath channel that deteriorates the performance of an OFDM system. Moreover, the estimation and the subsequent equalization of such a channel is a non-trivial task since the channel sate is continuously changing and the equalizer must be constantly redesigned to match the channel condition. Therefore, a robust channel estimation scheme is essential to compensate for the disturbances experienced by the OFDM signal, when passing through a doubly selective channel.

2.2 GNU Radio Architecture

GNU Radio is a free and open-source software development tool-kit that provides the signal processing blocks to implement software radios [3]. It can be used with readily-available low-cost external RF hardware to create software-defined radios, or without hardware in a simulation-like environment. It is widely used in hobbyist, academic and commercial environments to support both wireless communications research and real-world radio systems.

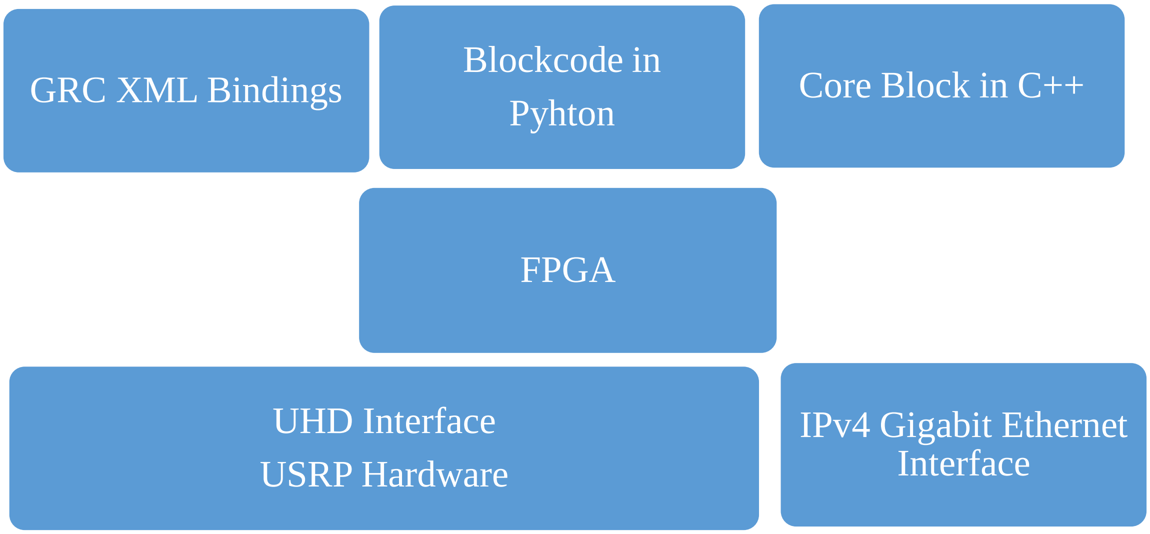


Fig. 2.1 GNU Radio Programming Stack

It has a graphical user interface called as the "GNU Radio Companion", that has drag and drop signal processing blocks that can be connected together to simulate basic signal processing operations. Custom-made blocks can be created and used in a similar way to build more complex and user-defined signal processing schemes. To create custom blocks, one has to know C++ programming, as most of the performance critical blocks are written in C++ [26]. It is almost important to mention that knowledge of Python programming language, use of "CMakeLists", and the working of Extensible Markup Language (XML) files are essential to build a complete framework on the GNU Radio platform. Many protocols have been simulating on the GNU Radio, like Bluetooth, IEEE 802.15.4 or WPAN, and GSM [22]. The use of GNU Radio with USRP can help researchers in avoiding the closed source firmware/drivers of commercial chip-sets by providing an ability for full customization at physical and data-link-layers.

2.3 Experimental Setup of GNU Radio, USRP and SDR Toolkit

The Ettus Research USRP family of Software Defined Radios (SDRs) are versatile devices that allow users to transmit and receive many different and custom waveforms at various frequencies and settings on a common hardware platform [7]. Commercial, academic, and military customers employ the flexible and reusable USRP hardware for research, wireless exploration, and field deployments [2]. In this thesis, we used Network (N200) Series for designing the IEEE 802.11 standard transceiver and FER computation in different channel models. It can connect to a host computer via a Gigabit Ethernet connection.

<table>
<tr><td rowspan="3">**Linux Host Machine**
(OS: Ubuntu 16.04 LTS)</td><td>**GNU Radio (Simulator)**</td><td>GRC Bindings (XML)</td><td>Block (python)</td><td>SWIG (Interfacing between C / C++ and python)</td><td>Core Blocks (C / C++)</td></tr>
<tr><td colspan="5">SW Interfaces to UHD C / C++ API
UHD Driver (C / C++)</td></tr>
<tr><td colspan="5">**Hardware Interface** (Gigabit Ethernet Port) / IPv4: 192.168.10.2</td></tr>
<tr><td colspan="2" rowspan="4">**Host Based USRP**</td><td colspan="4">Packet Tx / Rx</td></tr>
<tr><td colspan="4">FPGA (Verilog)</td></tr>
<tr><td colspan="4">Digitizers</td></tr>
<tr><td colspan="4">RF Front End (Tx / Rx)</td></tr>
</table>

Fig. 2.2 GNU Radio, UHD and USRP stack

USRPs are transceivers, meaning that they can both transmit and receive RF signals. UHD provides the necessary control used to transport user waveform samples to and from USRP hardware as well as control various parameters (e.g. sampling rate, center frequency, gains, etc) of the radio. The USRP Hardware Driver (UHD) is a user-space library that runs on a General-Purpose Processor (GPP) and communicates with and controls all of the USRP device family. The majority of the UHD code base is open source, including code that executes on the host, as well as code targeted to the USRP hardware (FPGA and microcontroller firmware). As dual-licensed software, UHD is available under the open-source GNU Public License version 3 (GPLv3), as well as an alternate, less-restrictive license offered only by Ettus Research. UHD can be installed on any Linux and UNIX based kernel. UHD GPP driver and firmware code is written in C/C++ while the code developed for the FPGA is written in Verilog [20]. There is a C/C++ API that can interface to other software frameworks, as in the case of GNU Radio, or a user can simply build custom signal processing applications directly on top of the UHD C/C++ API as shown in Figure 1.2.

Therefore, UHD library is basically an interface to connect to and to connect to and send and receive data between the Ettus Research. To use the UHD blocks, the Python namespaces is in gnuradio.uhd, which would be normally imported as [20]:

```
from gnuradio import uhd
```

The relevant blocks are listed in the UHD Interface group. The most important components are the “gr::uhd::usrp_source” and “gr::uhd::usrp_sink” blocks, which act as receivers/transmitters. Both are derived from “gr::uhd::usrp_block”, which defines many of the shared functions between those blocks. The UHD source and sink blocks are available in GRC under the section of UHD. These blocks take input some important parameters to process the signal over the wireless channels. Those parameters details are given at Table 2.1 in details.

The following command keys are understood by both UHD Source and Sink:

Table 2.1 Some Important Parameters of UHD Source and Sink Block in GRC [20]

Command name	**Value Type**	**Description**
Chan	Int	Specifies a channel. If this is not given, either all channels are chosen, or channel 0, depending on the action. A value of -1 forces 'all channels', where possible.
Gain	double	Sets the Tx or Rx gain (in dB). Defaults to all channels.
Freq	double	Sets the Tx or Rx frequency. Defaults to all channels. If specified without lo_offset, it will set the LO offset to zero.

lo_offset	double	Sets an LO offset. Defaults to all channels. Note this does not affect the effective center frequency.
Tune	tune_request	Like freq, but sets a full tune request (i.e. center frequency and DSP offset). Defaults to all channels.
lo_freq	double	For fully manual tuning: Set the LO frequency (RF frequency). Conflicts with freq, lo_offset, and tune.
dsp_freq	double	For fully manual tuning: Set the DSP frequency (CORDIC frequency). Conflicts with freq, lo_offset, and tune.
direction	string	Used for timed transceiver tuning to ensure tuning order is maintained. Values other than 'TX' or 'RX' will be ignored.
rate	double	See usrp_block::set_samp_rate(). *Always* affects all channels.
bandwidth	double	See usrp_block::set_bandwidth(). Defaults to all channels.
time	timestamp	Sets a command time. See usrp_block::set_command_time(). A value of PMT_NIL will clear the command time.
mboard	Int	Specify mboard index, where applicable.
antenna	string	See usrp_block::set_antenna(). Defaults to all channels.

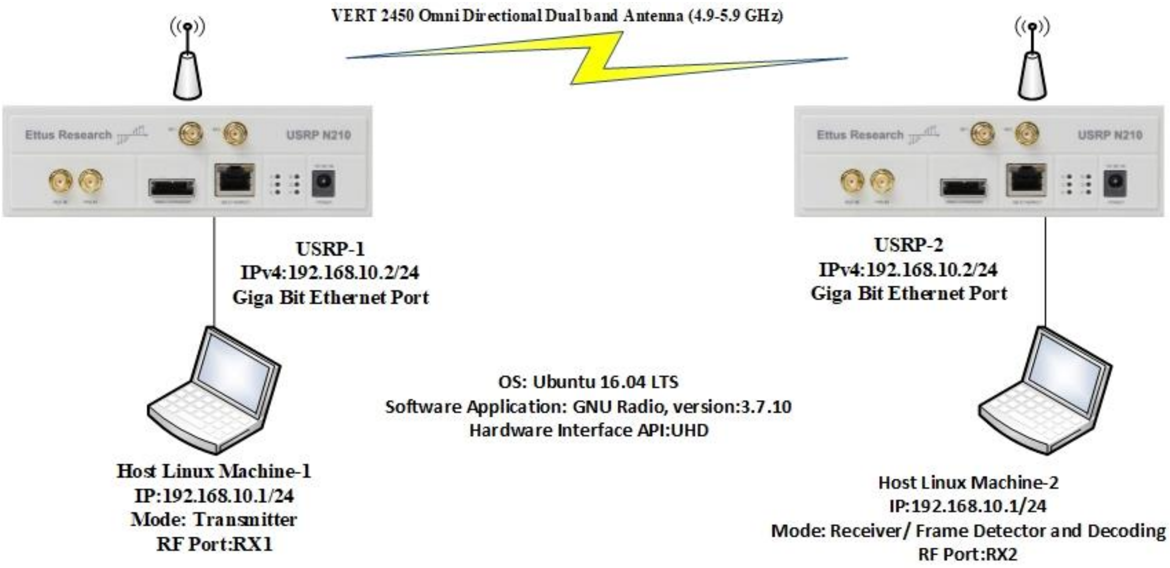


Fig. 2.3 Experimental testbed for SDR GNU Radio and USRP

Figure 2.3 illustrates the experimental testbed scenario that has been used to simulate IEEE 802.11 transceiver in lab environment. The Linux host machines generates and process the detection and decoding of IEEE 802.11 a/g/p standard frames. The USRP devices process the RF signal over the wireless medium. Since the device is in lab environment, there were a complete line of sight between the transmitting and receiving antenna. There will be no obstacles that may create multi-path fading effect. Therefore, a time and frequency selective multipath environment should be created by using the software defined radio. Here, GNU radio creates a time varying, doppler shifted channel model that affects the transmitted OFDM symbols to analyze the performance in real life scenario.

Table 2.2 describes the list of hardware and software requirements with their specific version and model. This thesis simulation testbed is done using the hardware's and software's given below:

Table 2.2 Experiment Setup Hardware and Software Details

Component	**Type**
CPU / Processor /RAM	Intel Core-i3, RAM-8GB, 64-bit, CPU-2.3 GHz.
NIC	Giga Bit Ethernet LAN Card
Operating System	Ubutnu Desktop 16.04 LTS
GNU Radio	Version 3.7.10
SDR	Ettus Research N200
Daughter Board	CBX Full Duplex Transceiver (operating frequency range: 1.2 GHz-6 GHz, Bandwidth: 40MHz)
Antenna	VERT 2450 Omni Directional Dual band Antenna (2.4 to 2.48 GHz and 4.9 to 5.9 GHz and 3dBi Gain)

CHAPTER 3

IEEE 802.11 a/g/p ARCHITECTURE AND TRANSCEIVER DESIGN IN GNU RADIO

The IEEE 802.11p, historically, was defined as the PHY standard for WAVE and Dedicated Short Range Communication (DSRC) [9]. Some of the frequency bands allocated for DSRC applications in different regions are shown in Table 3.1. The Federal Communications Commission (FCC) allocated 75 MHz of spectrum in the 5.9 GHz band for use by ITS vehicle safety and mobility applications. In this frequency bands, the IEEE 802.11p created 7 different channels, each having a 10 MHz bandwidth as shown in Figure 3.1 [15]. The control channel is used to transmit safety messages, and service channels are used for data transmission. It offers data exchange among vehicles and between vehicles and road side infrastructure (V2I) within a range of 1km while using a transmission rate of 3 Mbps to 27 Mbps up to vehicle velocity of 260 km/h. Although, the bit rate of the primary standard (IEEE802.11a) was up to 54Mbps, the bit rate of IEEE 802.11p has been reduced to half because of the limitations of PHY layer in coping with the high mobility vehicular characteristics [7].

The main aspects of the IEEE 802.11p under consideration in this thesis are the PHY layer specifications. The IEEE 802.11p, as mentioned in section 1.1, is next version of IEEE 802.11a standard which is widely used for WiFi. The OFDM packet frame of IEEE 802.11p is shown in Figure 3.2, which is analogous to the IEEE 802.11a packet. PHY layer in IEEE 802.11p is quite similar to IEEE 802.11a but with some additional features to handle the vehicular environment, so the priority changes depending upon the situation it is implemented.

Table 3.1 Spectrum Allocation for WAVE/DSRC Application [15]

Country/Region	Frequency Bands (MHz)	Reference Documents
ITU-R (ISM band)	5725-5875	Article 5 of Radio Regulations
Europe	5795-5815, 5855/5875, 5905/5925	ETS 202-663, ETSI EN 302-571, ETSI EN 301-893
North America	902-928, 5850-5925	FCC 47 CFR
Japan	715-725, 5770-5850	MIC EO Article 49

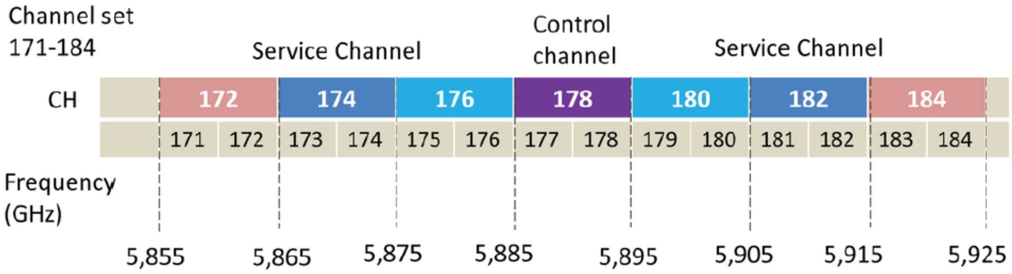
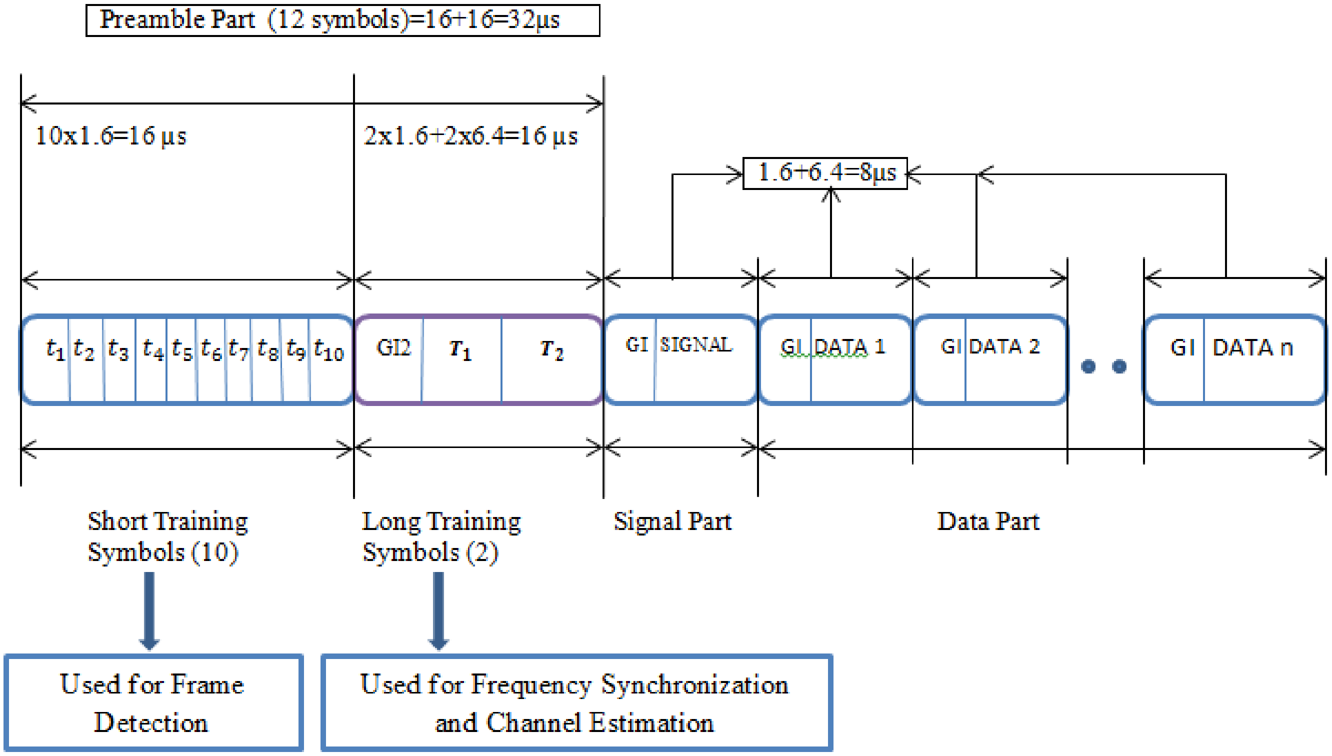


Fig. 3.1 IEEE 802.11p Spectrum [15]

To overcome the difficulty in communication in WAVE, changes done in PHY layer of IEEE 802.11p compared to IEEE 802.11a as shown in Table 3.4. In vehicular environments, the maximum multi-path delay is more expected to be more than the indoor environment. Therefore, the guard time in IEEE 802.11p is doubled. The data rate in IEEE 802.11p for similar modulation schemes, compared to IEEE 802.11a, is halved as the bandwidth is also halved [1]. This also has a consequence on the symbol period, which is also doubled compared to IEEE 802.11a. The changes are in accordance to the requirements of the current urban vehicular environment where the Doppler shifts are not substantial enough to adversely affect the OFDM transmission.

The beginning of a MAC frame has to be properly detected by a receiver in order to correctly interpret the data samples. The receiver needs to estimate the exact sample positions; therefore, each MAC frame is extended by a fixed preamble. One purpose of the preamble is to enable the receiver to detect the start of a frame and to obtain a training symbol synchronization to ensure that the individual data symbols of the following MAC frame are correctly aligned in time and signal amplitude. This synchronization procedure is called the Physical Layer Convergence Protocol (PLCP). The PLCP header and the data symbols along with few additional fields are combined together to form a single Physical Layer Protocol Data Unit (PPDU) as shown in below Figure 3.3 [12].



Preamble Part (12 symbols)=16+16=32µs
10x1.6=16 µs
2x1.6+2x6.4=16 µs
1.6+6.4=8µs
t_1 t_2 t_3 t_4 t_5 t_6 t_7 t_8 t_9 t_{10}
GI2
T_1
T_2
GI SIGNAL
GI DATA 1
GI DATA 2
GI DATA n
Short Training Symbols (10)
Long Training Symbols (2)
Signal Part
Data Part
Used for Frame Detection
Used for Frequency Synchronization and Channel Estimation

Fig. 3.2 IEEE 802.11p frames comprise a short and a long training sequence for synchronization; a signal field, containing information about the length and encoding of the frame; and the data symbols, carrying the actual payload.

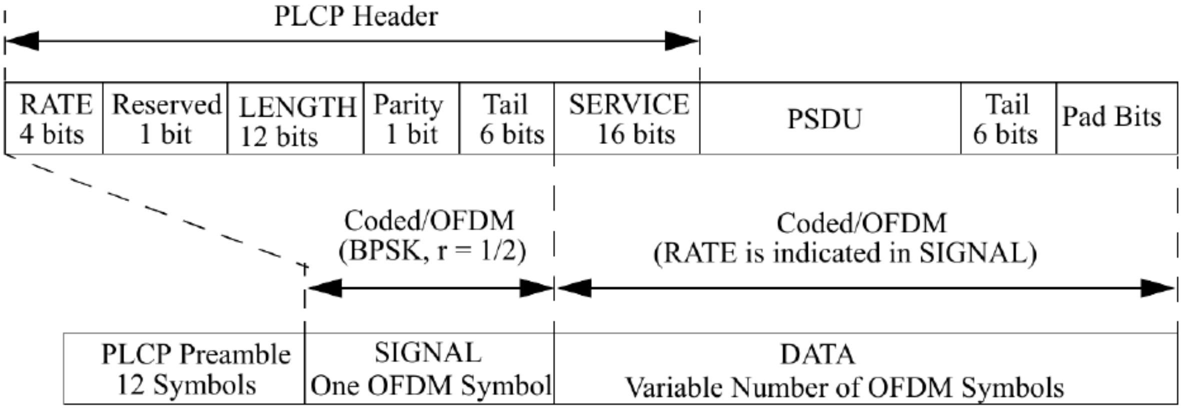


Fig. 3.3 PPDU Frame Format [12]

3.1 Short Training Sequence

A fixed sequence of 10 short symbols is transmitted in the first part of the PLCP preamble. This is also called as the “short training sequence” or the “short preamble”. Automatic Gain Control (AGC) or signal amplitude convergence, coarse timing acquisition and coarse frequency acquisition are accomplished by adopting this short preamble. The size of short training symbol is 10 number of samples and each symbol is 1.6 μs long at 10 MHz frequency. 10 such symbols put together is 16μs long, which is the total duration of the shot training sequence that exits in the MAC frame. The short training sequence is generated by computing the 64-FFT of the complex sequence given in Equation 3.4. The numbering of the sub-carriers in a 64-OFDM symbol are done from -32 to +31, totaling to 64 sub-carriers. The sub-carrier labeled as “0” is called as the Direct Current (DC) sub-carrier and is always set to the value zero. It corresponds to the center frequency of the Radio Frequency (RF) signal being transmitted [6].

3.2 Long Training Sequence

The “Long Preamble” or “Long Training Sequence” in the PLCP header is used for channel estimation, better symbol alignment and also more precise frequency offset corrections [1], [14]. They can also be used to track the channel variations from one OFDM frame to another, but as they are placed only at the beginning of an OFDM frame, channel variations within the OFDM frame cannot be tracked using the Long Preamble. There are actually two Long Preambles in the PLCP header. Each long preamble is 6.4 μs long consisting of 52 non-zero sub-carriers and 12 null sub-carriers. Between the short preamble and the long preamble, two guard interval sequences that are the cyclic prefix of the long preamble added, where each of cyclic prefix is 1.6 μs long. Therefore, the PLCP preamble consists of one short training sequence of 16μs, two cyclic prefix sequences of 1.6μs each, and two long preambles of 6.4μs each, accumulating to a total duration of 32μs as given in Figure 3.2 [1]. The long training sequence is generated by computing the 64-FFT of the sequence given in Equation 3.5.

3.3 Pilots

In each of OFDM data symbols following the PLCP header, there are 4 pilot sub-carriers that can be used for further channel estimation and correction. As the PLCP header is at the beginning of the OFDM frame, they provide a very good estimate of the channel at their respective time period. However, we expect the channel to change within the OFDM frame in a high mobility environment, and here the aforementioned pilots can be used to trace the changes of the channel and adjust the estimated channel matrix accordingly. The sub-carriers defined for the pilots are at the positions -21, -7, 7, 21 and their initial values are set to $(1+0j),(1+0j),(1+0j)\ and\ (-1+0j)$ respectively [1], [3]. These values are again influenced by Pseudo Noise (PN) sequence that is generated by using the generator polynomial given in Equation 3.1 [3].

$$m(x) = x^7 + x^4 + 1 \tag{3.1}$$

The 127-bit sequence generated repeatedly with an initial state of all ones is given in Equation 3.2. In the PN sequence, all zeros are replaced with the value of "+1" and all ones are replaced with the value of "-1" to obtain the polarity sequence show in Equation 3.3.

$$\boldsymbol{pn} = (\begin{matrix} 00001110 & 11110010 & 11001001 & 00000010 \\ 00100110 & 00101110 & 10110110 & 00001100 \\ 11010100 & 11100111 & 10110100 & 00101010 \\ 11111010 & 01010001 & 10111000 & 1111111) \end{matrix} \tag{3.2}$$

$$\boldsymbol{p_{0,126}} = (\begin{matrix} 1,1,1,1,-1,-1,-1,1, & -1,-1,1,1,-1,1,1,-1, & -1,-1,1,1,-1,1,1,-1, & 1,1,1,1,1,1,-1,1, \\ 1,1,-1,1,1,-1,-1,1, & 1,1,-1,1,-1,-1,-1,1, & -1,1,-1,-1,1,-1,-1,1, & 1,1,1,1,-1,-1,1,1, \\ -1,-1,1,-1,1,-1,1,1, & 1,1,-1,1,-1,-1,-1,1 & -1,1,-1,-1,1,-1,1,1, & 1,1,1,1,-1,-1,1,1, \\ -1,-1,-1,-1,-1,1,-1,1, & 1,-1,1,-1,1,1,1,-1, & -1,1,-1,-1,-1,1,1,1, & -1,-1,-1,-1,-1,-1,-1) \end{matrix} \tag{3.3}$$

Each symbol in the polarity sequence is used for the entire OFDM symbol, which means the pilots in the entire OFDM symbol is multiplied by "+1" or "-1". Table 3.2 illustrates that operation. Incremental phase offset corrections, or timing offset corrections that are confined to a single OFDM symbol are also estimated by using the pilots and linear regression method. It is important to note that the position of the pilots in the entire OFDM symbol might also contribute to the accuracy of such conditions. Therefore, uniformly spread and equally spaced positions are designed as pilots in most standards.

Table 3.2 Polarity of Pilots

OFDM Symbol	Polarity	Pilots			
		-21	-7	7	21
0	1	$(1+0j)$	$(1+0j)$	$(1+0j)$	$(-1+0j)$
1	1	$(1+0j)$	$(1+0j)$	$(1+0j)$	$(-1+0j)$
2	1	$(1+0j)$	$(1+0j)$	$(1+0j)$	$(-1+0j)$
3	1	$(1+0j)$	$(1+0j)$	$(1+0j)$	$(-1+0j)$
4	-1	$(-1+0j)$	$(-1+0j)$	$(-1+0j)$	$(1+0j)$
5	-1	$(-1+0j)$	$(-1+0j)$	$(-1+0j)$	$(1+0j)$
6	-1	$(-1+0j)$	$(-1+0j)$	$(-1+0j)$	$(1+0j)$
7	1	$(1+0j)$	$(1+0j)$	$(1+0j)$	$(-1+0j)$

3.4 Guard Interval

The guard interval is used in OFDM to combat multi-path effect and avoid ISI. The guard interval here, is also called as the cyclic prefix, where a few defined numbers of samples from the end of the OFDM symbol is prefixed to the start of the symbol. The number of samples taken as cyclic prefix in IEEE 802.11p is defined to be 16 samples, or 25% of the entire OFDM symbol duration. This makes the cyclic prefix 1.6µs long, having 16 samples, which is twice as long as the guard interval in IEEE 802.11a. As long as the distance between the different paths of the multi-path effects stay below the length of the cyclic prefix, the ISI can be compensated [17].

3.5 Guard Band

The guard band that is located at the outer limits of the frequency band is used to adhere to the strict spectral mask defined by the standardizing committees. In IEEE 802.11p the first 6 sub-carriers, and the last 5 sub-carriers are set to zero to create the guard band, which is the same as IEEE 802.11a. Although this reduces the maximum data rate that can be achieved, it is important to avoid aliasing, so that the adjacent channels shown in figure do not interfere with each other [17].

3.6 Signal Field

Another integral part of the PLCP header is the SIGNAL field that follows the OFDM training symbols described above. Within the signal field, the RATE field conveys the type of modulation and the coding rate for the entire frame and its values are given in Table 3.3. This means that the modulation scheme and code rates for the data sub-carries do not vary from one OFDM symbol to another and is consistent for entire frame. Additionally, the schemes are also fixed for each sub-carrier [17].

The LENGTH field also exists within the signal field, which is employed to transmit the size of the MAC frame. The encoding is performed with Binary Phase Shift Keying (BPSK) modulation of the sub-carriers and using convolution coding at a coding rate of ½, which restricts the size of the signal field to 24 bits as shown in Figure 3.4. The entire coding procedure, which includes convolution encoding, interleaving, modulation mapping processes, pilot insertion and OFDM modulation, are performed for the signal field as per the 'BPSK ½' guidelines [19].

Table 3.3 RATE Field within the SIGNAL Field

R1-R4	**Rate (Mbps)**
1101	3
1111	4.5
0101	6
0111	9
1001	12
1011	18
0001	24
0011	27

RATE (4 bits)					LENGTH (12 bits)													SIGNAL TAIL (6 bits)					
R1	R2	R3	R4	R	LSB											MSB	P	0	0	0	0	0	0
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23

Transmitting Order

Fig. 3.4 SIGNAL Field Definition [22]

The four bits 0 to 3 represents the RATE, as shown in Figure 3.4. Bit 4 is reserved for the future use. Bits 5-16 encode the LENGTH. The 12-bit unsigned integer value in the length filed is equal to the number of octets (bytes) that the MAC frame is requesting the PHY layer to transmit. Bit 17 is a positive parity (even parity) bit for 0-16 bits. The bits 18-23 constitute the SIGNAL TAIL field, and all 6 bits are set to 0. The receiver decodes the signal field to identity the modulation scheme and the code rate of the frame. If this field is erroneous at the receiver then the entire frame is affected due to the wrong choice of demodulation or code rate. The BER calculations at the receiver end is unnecessarily elevated for the channel estimation schemes. Therefore, fixed data in the signal field is transmitted through the simulation platform, but the values are ignored at the receiver. Instead, modulation scheme and code rate are initially defined for the entire frame at the receiver so that the BER results are indubitable and consistent over multiple simulations runs. Additionally, this method is unchanged for all the channel estimation schemes under investigation.

3.7 Data Field

The DATA field, shown in Figure 3.3, consists of a SERVICE field at the beginning. This is 16 bits long and it is mainly used to initialize the descrambler at the receiver as shown in Figure 3.6. This is followed by the MAC layer data symbols aggregated into the Physical Layer Service Data Unit (PSDU), which contains the data that is to be transmitted over the channel. The 6 TAIL bits shown in Figure 3.3 are required to return the convolution encoder to the zero state. Additionally, PAD bits are vital to ensure adequate number of bits for complete mapping onto the OFDM symbols.

Scrambler Initialization							Reserved Service Bits (R: Reserved)								
0	0	0	0	0	0	0	R	R	R	R	R	R	R	R	R
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Transmitting Order

Fig. 3.5 SERVICE Field Definition [22]

The data field, composed of SERVICE, PSDU, tail and pad parts are scrambled with a length-127 frame-synchronous scrambler. The generator polynomial is the same as given in Equation 3.1. Hence, the sequence generated is also identical to Equation 3.2. The data descrambler is illustrated in Figure 3.6.

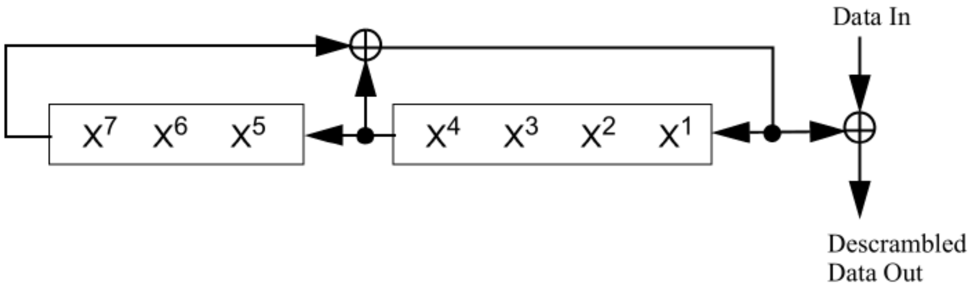


Fig. 3.6 Data Descrambler Structure [23]

The entire data field is also encoded using the convolution coder with the generator $g_0 = 133_8$ and $g_1 = 171_8$. Combining this with specific puncturing patterns results in different code rates. The convolution coder is illustrated in Figure 3.7 for a constraint length of k=7.

After convolution encoding and puncturing, interleaving of the data is performed at the transmitter. Generally, the errors due to noise in wireless channels are burst errors, which means that a lot of errors occur within a small amount of time. To circumvent this effect, interleaving is implemented that pseudo randomly distributes portions of data over the entire stream.

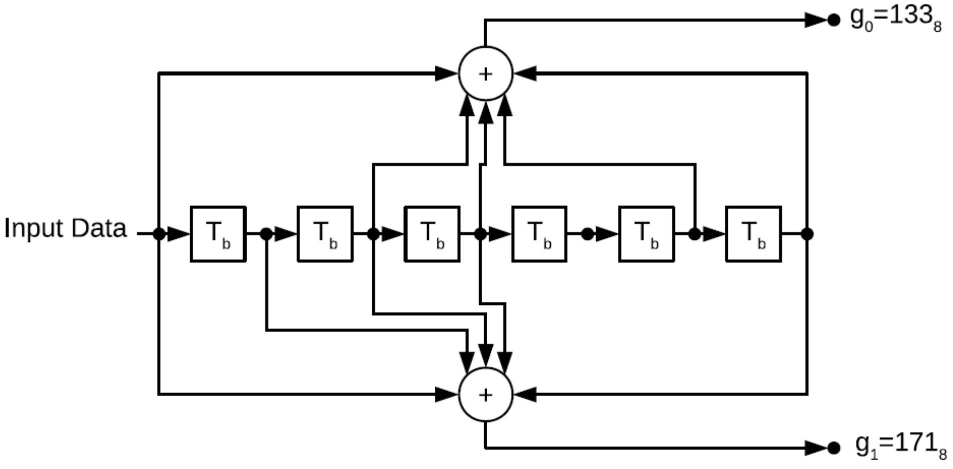


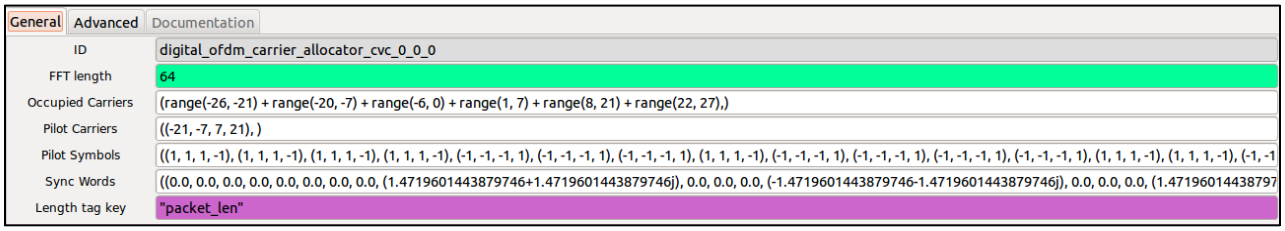
Fig. 3.7 Convolution Encoder [23], [22]

Combining this with convolution encoding, which can also compensate for burst errors, ensures considerable reduction in the BER. The modifications in the data field, which are essential for an end-to-end communication system, have substantial positive contributions towards the performance of the channel estimation schemes. The SERVICE field, which is used to initialize the descrambler, can be considered as additional data that is being transmitted and is indeed inconsequential to the channel estimation scheme. The scramble is utilized to randomize data, such that a lengthy stream of “1” s or “0” s does not occur continuously. It simply uses to avoid long sequence with same value. The descrambler performed the inverse process. These methods also contribute in reduction of BER which could distort the genuine performance of the channel estimation schemes in the simulation platform.

Furthermore, the effect of convolution coding and interleaving are significant enough to deceive the researcher into believing that the gain in performance might be due to the developments in the channel estimation scheme. Attributing the performance to the channel estimation and equalization schemes alone becomes problematic and inaccurate. To avoid such points of contention and represent the legitimate performance of the various channel estimation schemes, all modifications to the data field are ignored in the simulation platform. Therefore, a block of unmodified data prefixed with the PLCP header is transformed into OFDM symbols, as per the guidelines introduced before, and transmitted over the channel. At the receiver, the received data, after channel estimation and channel equalization, is compared with the template that is transmitted to calculate the BER performance. This method ensures that the measured performance is a sincere characteristic of the channel estimation scheme and is consistent over several iterations. Additionally, it is encouraging to note that, all the above methods used for modifying data can only improve the BER performance rather than degrade it.

3.8 Design Implementation in GNU Radio Companion

The IEEE 802.11 a/g/p standard transmitter specifies total 64 sub-carriers (N) for the 20 MHz (802.11 a/g) or 10 MHz bandwidth (802.11p), among them data sub-carriers $N_D = 48$ are used for data, 4 pilot sub-carriers N_P are assigned between data sub-carriers as per comb pilot insertion method to assist receiver in frequency offset detection and compensation, rest 12 sub-carriers are kept as null N_0 which is used to avoid Direct Current (DC) offset and frequency leakage to other frequency bands [1]-[3]. In case of every non-null sub-carrier first two symbols are kept for pilots were known as block type pilot symbols allocation to estimate channel coefficient. This type of carrier allocation known as block-comb combination method and it is very efficient to deal with quasi-static channels [10].



ID	digital_ofdm_carrier_allocator_cvc_0_0_0
FFT length	64
Occupied Carriers	(range(-26, -21) + range(-20, -7) + range(-6, 0) + range(1, 7) + range(8, 21) + range(22, 27),)
Pilot Carriers	((-21, -7, 7, 21),)
Pilot Symbols	((1, 1, 1, -1), (1, 1, 1, -1), (1, 1, 1, -1), (1, 1, 1, -1), (-1, -1, -1, 1), (-1, -1, -1, 1), (-1, -1, -1, 1), (1, 1, 1, -1), (-1, -1, -1, 1), (-1, -1, -1, 1), (-1, -1, -1, 1), (-1, -1, -1, 1), (1, 1, 1, -1), (1, 1, 1, -1), (-1, -1
Sync Words	((0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, (1.4719601443879746+1.4719601443879746j), 0.0, 0.0, 0.0, (-1.4719601443879746-1.4719601443879746j), 0.0, 0.0, 0.0, (1.47196014438797
Length tag key	"packet_len"

Fig. 3.8 IEEE 802.11 a/g/p OFDM sub-carrier allocator configuration

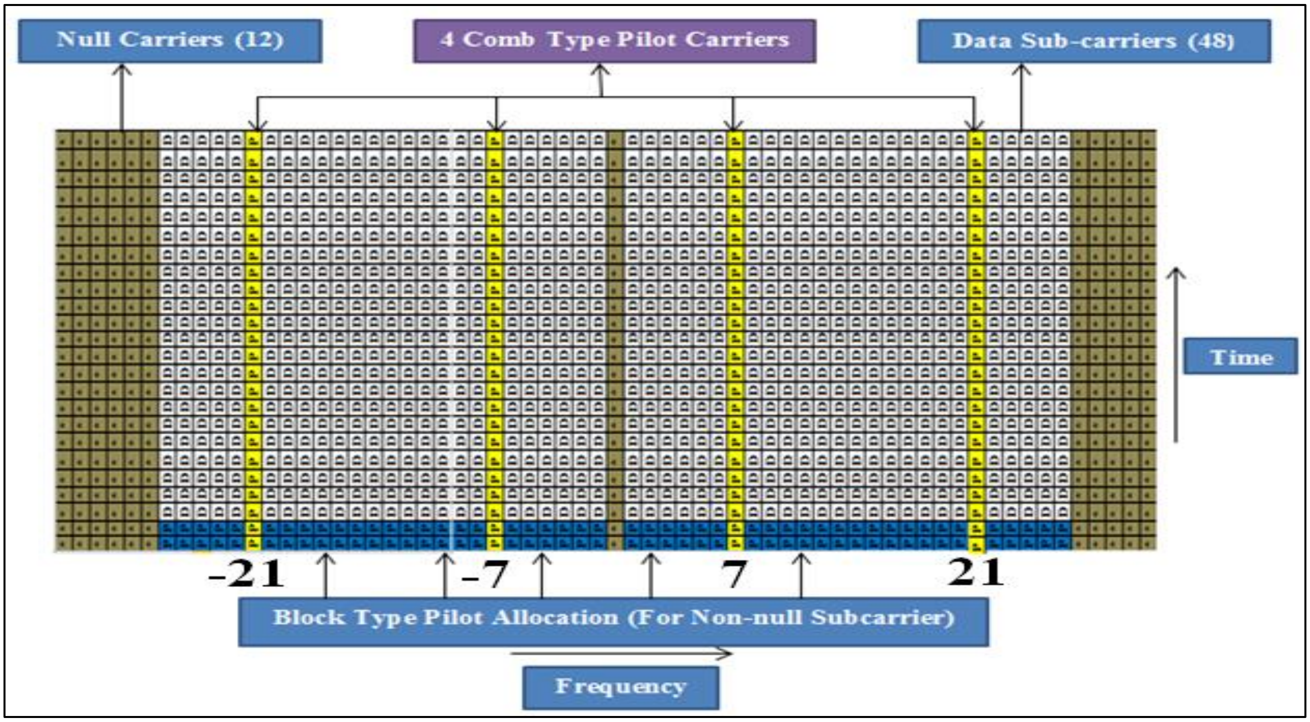
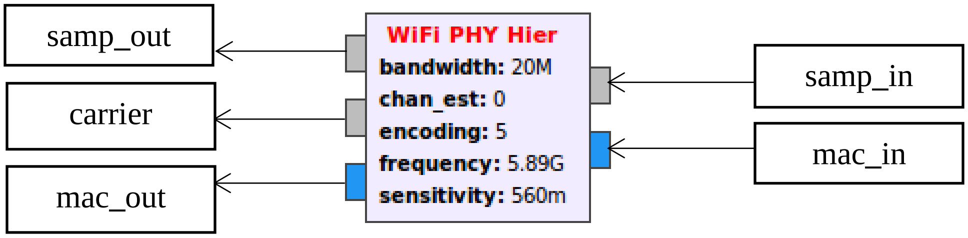

Null Carriers (12)
4 Comb Type Pilot Carriers
Data Sub-carriers (48)
Time
-21
-7
7
21
Block Type Pilot Allocation (For Non-null Subcarrier)
Frequency

Fig. 3.9 Sub-carrier's allocation and pilot's insertion (comb and block types) in IEEE 802.11p

The receiver will get multiple copies of the signal with different strength even at the different time due to the reflection, scattering of the transmitted signal over the fading channels which may introduce the frequency offset in the sub-carriers. As a result, they may lose the orthogonal property resulting inter carrier interference. To remove the multipath effects guard interval N_G known as a cyclic prefix is used at the beginning of each OFDM symbol. The last $N_G = 16$ samples are copied from the time domain OFDM symbol x and put at the beginning of the x. As a result, $N + N_G = 80$ samples $(x_{N-N_G+1}, \dots \dots, x_N, x_1, \dots \dots, x_N)$ are transmitted per OFDM symbol [12].

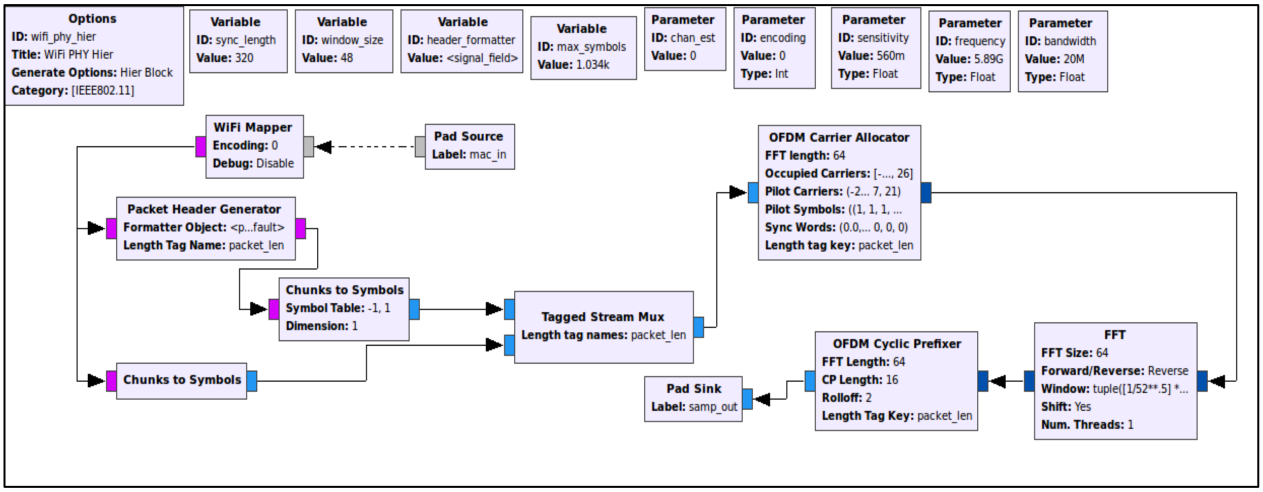
The physical layer is encapsulated in WIFI PHY Hierarchical block which is the integrated systems for the transmitter and receiver in GNU Radio Companion given at Figure 3.10**.**



bandwidth: 20M
chan_est: 0
encoding: 5
frequency: 5.89G
sensitivity: 560m

Fig. 3.10 IEEE 802.11 WIFI PHY HIER Block (TX and RX)

Basically, the transmitter consists of an encoder, OFDM carrier allocator, Inverse Fast Fourier Transform (IFFT) operation and cyclic prefix adder. In hierarchical block, WIFI Mapper is responsible for data scrambling, interleaving and convolution encoding. Packet Header Generator generates the header of the frame including the signal and services field.



Options
ID: wifi_phy_hier
Title: WiFi PHY Hier
Generate Options: Hier Block
Category: [IEEE802.11]

Variable
ID: sync_length
Value: 320

Variable
ID: window_size
Value: 48

Variable
ID: header_formatter
Value: <signal_field>

Variable
ID: max_symbols
Value: 1.034k

Parameter
ID: chan_est
Value: 0

Parameter
ID: encoding
Value: 0
Type: Int

Parameter
ID: sensitivity
Value: 560m
Type: Float

Parameter
ID: frequency
Value: 5.89G
Type: Float

Parameter
ID: bandwidth
Value: 20M
Type: Float

WiFi Mapper
Encoding: 0
Debug: Disable

Pad Source
Label: mac_in

Packet Header Generator
Formatter Object: <p...fault>
Length Tag Name: packet_len

Chunks to Symbols
Symbol Table: -1, 1
Dimension: 1

Chunks to Symbols

Tagged Stream Mux
Length tag names: packet_len

OFDM Carrier Allocator
FFT length: 64
Occupied Carriers: [-..., 26]
Pilot Carriers: (-2... 7, 21)
Pilot Symbols: ((1, 1, 1, ...
Sync Words: (0.0,... 0, 0, 0)
Length tag key: packet_len

Pad Sink
Label: samp_out

OFDM Cyclic Prefixer
FFT Length: 64
CP Length: 16
Rolloff: 2
Length Tag Key: packet_len

FFT
FFT Size: 64
Forward/Reverse: Reverse
Window: tuple([1/52**.5] *...
Shift: Yes
Num. Threads: 1

Fig. 3.11 IEEE 802.11 standard packet transmitter

The frame header is BPSK modulated by the upper chunks to symbols block (symbol table: 1, -1) and the remaining frame is modulated by the bottom chunks to symbols block as per the preferred modulation schemes. The chunks to symbol block convert the incoming byte stream to the desired complex symbol based on the type of modulation that has been chosen. The available modulation schemes are BPSK, QPSK, 16-QAM and 64-QAM. Finally, the header is integrated to the remaining of the frame with a new stream tag by using Tagged stream MUX block. Then, multiplexed frames are transmitted to the OFDM carrier allocator block for the insertion of pilot sub-carriers uniformly inside the data sub-carriers and IFFT block is used to convert the symbols from frequency domain to the time domain. At last, OFDM Cyclic Prefixer block is used to add guard interval in the time domain before each OFDM symbol of the frames to avoid ISI. A short OFDM training symbols consist of 12 sub-carriers which are given below [23]:

$$S_{-26,26} =$$

$$\sqrt{\frac{13}{6}}\begin{bmatrix} 0, 0, 1+j, 0, 0, 0, -1-j, 0,0,0,1+j, 0,0,0, -1-j, 0,0,0, -1-j, 0,0,0, 1+j, 0, 0\ ,0\ , 0, 0, 0, 0, -1-j, 0 \\ , 0,0, -1-j, 0,0,0, 1+j, 0,0,0,1+j, 0,0,0, 1+j, 0,0,0, 1+j, 0,0 \end{bmatrix} \quad (3.4)$$

The factor $\sqrt{\frac{13}{6}}$ is used to normalize the average power of the OFDM symbol. The predefined long training symbols are 53 sub-carriers and it has a zero value at DC. The long training symbols are given below [23]:

$$L_{-26,26} = [1,1,-1,-1,1,1,-1,1,-1,1,1,1,1,1,1,-1,-1,1,1,-1,1,-1,1,1,1,1,0,1,-1,-1,1,1,-1,1,-1,1,-1, \\ -1,-1,-1,-1,1,1,-1,-1,1,-1,1,-1,1,1,1,1] \tag{3.5}$$

Table 3.4 IEEE 802.11 a/g/p Wireless Standard Parameters and Values [5], [23]

Parameter	Notation	IEEE 802.11a/g Values	IEEE 802.11p values
Total OFDM Sub-carrier	N_{SC}	64	64
Pilot Sub-carrier	N_P	4	4
Data Sub-carrier	N_{DSC}	48	48
Preamble Duration		16µs	32µs
Guard Time	GI	0.8µs	1.6µs
Bit Rate (Mbps)		6,9,12,18,24,26,48,54	3,4.5,6,9,12,18,24,27
Modulation Type		BPSK, QPSK, 16/64-QAM	BPSK, QPSK, 16/64-QAM
Bandwidth (MHz)	B	20MHz	10MHz
Symbol Period	$T_{OFDM} = \frac{N_{sc}+G}{B}$	4µs	8µs
Sub-carrier Frequency Spacing	$\Delta f = \frac{B}{N_{SC}}$	0.3125 MHz	0.15625MHz

Figure 3.2 describes the packet preamble structures of IEEE 802.11p. Each IEEE 802.11 a/g/p frames consists of preamble including 10 short training symbols (t_1 to t_{10}) of each 1.6 µs located at the beginning of each frame used to detect starting of a frame, 2 long training symbols (T_1 and T_2) of each 6.4µs used for channel equalization and channel estimation. The rest of the frame is the signal field and data the field. Signal field contains the information about modulation schemes, coding rates etc. by using only one OFDM symbol and the data field generally used for transmitting data. Guard interval is used to avoid inter symbol interference. The frame detection algorithm is based on the autocorrelation of the short training sequence which is 16 samples repeating 10 times. The receiver feats this cyclic pattern of short training symbols and calculates

the autocorrelation value a of the incoming baseband sample stream s with lag 16 by summing up the autocorrelation coefficients over an adjustable window N_{win} [2].

$$a(n) = \sum_{k=0}^{N_{win}-1} s(n+k)\bar{s}(n+k+16) \tag{3.6}$$

Here $\bar{s}$ express the complex conjugate of s. Experimental result showed that adjustable window N_{win}= 48 gives better performance and the summation over this window turns into moving average which acts as a low pass filter. The autocorrelation is high at the start of each IEEE 802.11 a/g/p frame due to the cyclic property of short training sequence. The autocorrelation $a(n)$ is normalized with the average power $p(n)$ such that the receiver was independent of the absolute level of incoming samples [2].

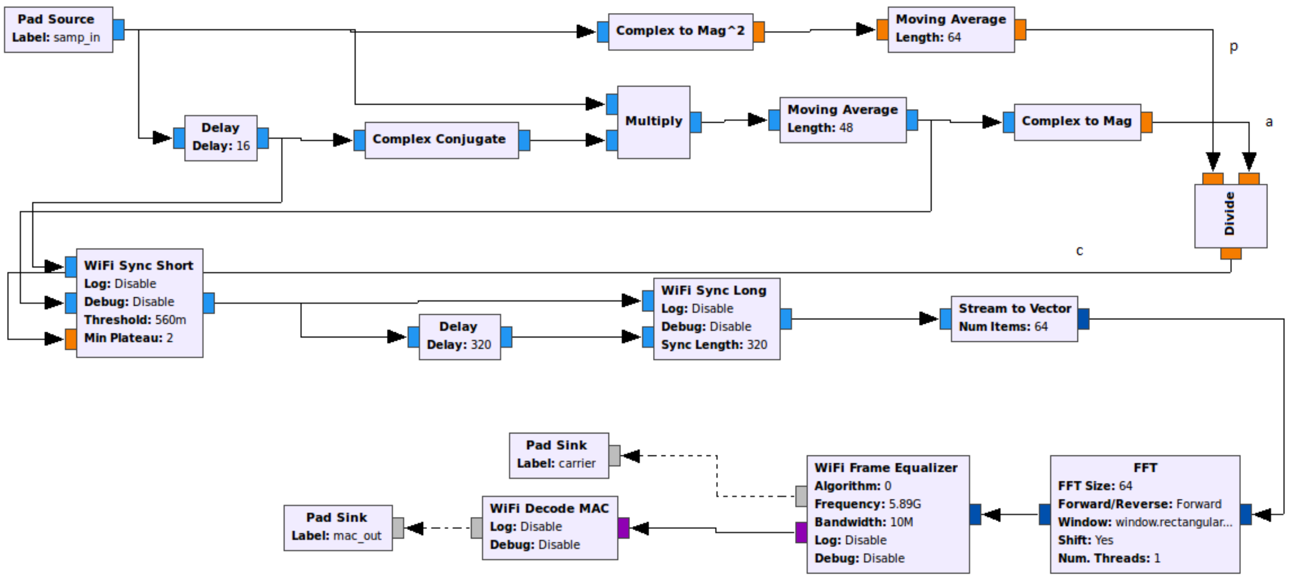
$$p(n) = \sum_{k=0}^{N_{win}-1} s(n+k)\bar{s}(n+k) \tag{3.7}$$

Finally, the normalized autocorrelation coefficient c can be expressed as [2]:

$$c(n) = \frac{|a(n)|}{p(n)} \tag{3.8}$$

Here, $|a(n)|$ denotes the magnitude of $a(n)$. Here the Figure 3.12 given below expressed the IEEE 802.11 standard frame receiver schematics at GNU Radio companion. These schematics have basically 2 parts one is frame detection and another part is frame decoding.

The calculation of autocorrelation coefficients can be composed from simple mathematical functions that are readily available in GNU Radio. An overview of the receiver flow graph in GRC is depicted in Figure 3.13, where marked the signal power (p), the autocorrelation (a) and the autocorrelation coefficient (c). The blocks make heavy use of vectorized instruction through GNU Radio Vectorized Library of Kernels (VOLK), allowing a typical PC to process even 20 MHz signals without dropping any samples.



Pad Source
Label: samp_in

Complex to Mag^2

Moving Average
Length: 64

Delay
Delay: 16

Complex Conjugate

Multiply

Moving Average
Length: 48

Complex to Mag

WiFi Sync Short
Log: Disable
Debug: Disable
Threshold: 560m
Min Plateau: 2

Delay
Delay: 320

WiFi Sync Long
Log: Disable
Debug: Disable
Sync Length: 320

Stream to Vector
Num Items: 64

Pad Sink
Label: carrier

WiFi Frame Equalizer
Algorithm: 0
Frequency: 5.89G
Bandwidth: 10M
Log: Disable
Debug: Disable

FFT
FFT Size: 64
Forward/Reverse: Forward
Window: window.rectangular...
Shift: Yes
Num. Threads: 1

Pad Sink
Label: mac_out

WiFi Decode MAC
Log: Disable
Debug: Disable

Fig. 3.12 IEEE 802.11 standard packet receiver at GNU Radio Companion. The signal power (p), autocorrelation between short preambles (a) and autocorrelation coefficient (c) are annotated.

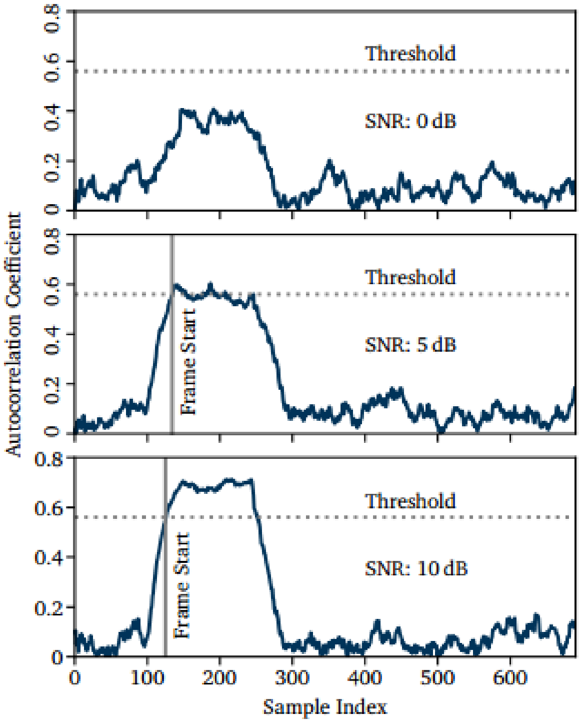


Fig. 3.13 The autocorrelation coefficients (c) during the start of a frame at different values of SNR= 0, 5 and 10 dB respectively. The frame detection is triggered once (c) exceeds the predefined threshold line that is represented in dotted line.

After that, normalized autocorrelation coefficients $c(n)$ and the received samples from the USRP are sent to the “WiFi Sync Short” block as input samples which act as like as a valve. During the starting of the frame “WiFi Sync Short” block detects a plateau and then it pipes a fixed number of samples into the rest of the signal processing pipeline, otherwise, it drops the samples.

Once a frame is detected we have to derive the exact position of the OFDM symbols in order to align the FFT in the receiver and decode the frame. “WiFi Sync Long” block is used for frequency offset correction and symbol alignment by using the characteristics of the long preamble. This block has 2 inputs, one is without delay, and other is delayed by 320 samples. 320 samples are total number of samples in the PLCP Preamble of IEEE 802.11p, which is 32µs long and each sample is 100ns.

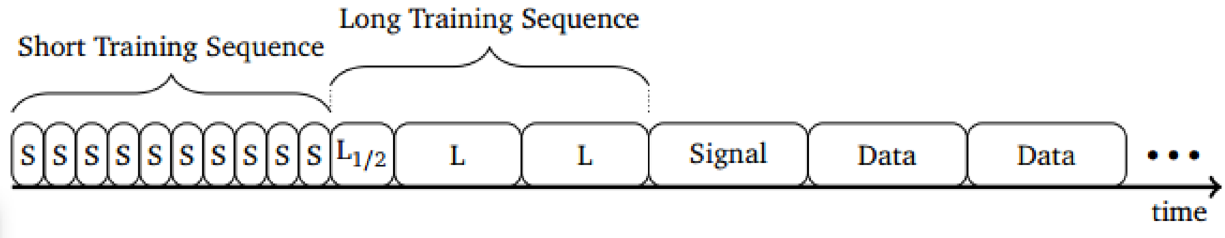


Fig. 3.14 IEEE 802.11p Frame Structure [3]

To align the symbol here a matched filter is employed to cross-correlate the sample stream $s(n+k)$ with the known pattern of the long training sequence $L_p(k)$. The typical course of the cross-correlation function is shown in Figure 3.15 for SNR= 10 dB. To determine the frame start, the receiver orders the values of the cross-correlation by their magnitude and searches for the three peaks N_p matches using the $\arg max_3$ operator as shown in Equation 3.9 that selects the indices of three highest values as Long Preamble has 3 filed values as shown in Figure 3.14.

$$N_p = \underset{n\epsilon\{0,\ldots\ldots,N_{preamble}\}}{\arg max_3} \sum_{k=0}^{63} s(n+k)\ \overline{L_p(k)} \tag{3.9}$$

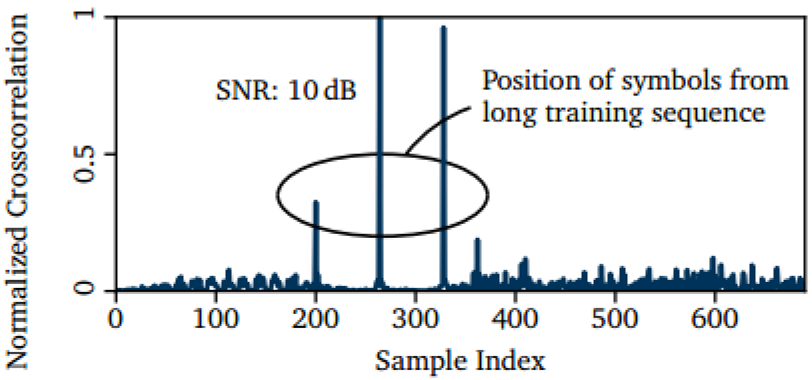
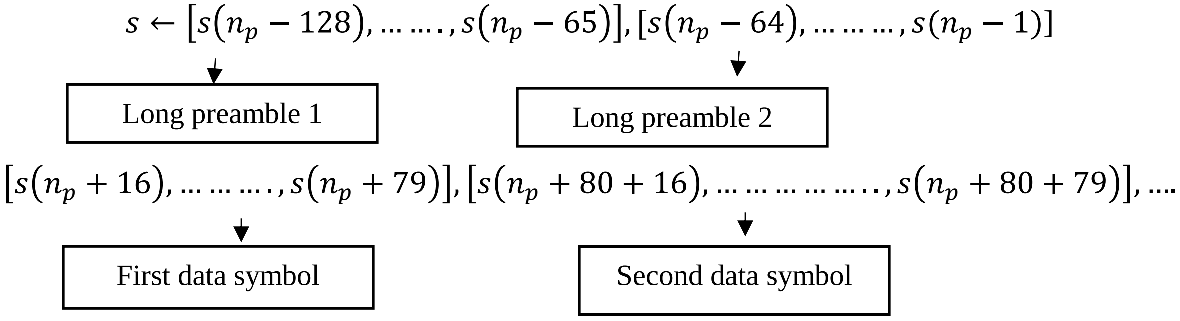


Fig. 3.15 Cross-correlation of a frame sample stream with the known pattern of the long preamble, as calculated by the receiver to determine the symbol alignment.

The spikes in Figure 3.15 indicate the position where this pattern matches with the input stream signal. The first OFDM data symbol starts 64 samples after the last peak, which is the peak with the highest sample index.

$$n_p = \max(N_p) + 64 \tag{3.10}$$

As shown as Figure 3.14, the long preamble symbols are sent back to back, whereas the data symbols use a cyclic-prefix of 16 samples. There for the symbol alignment of transmitted sample stream s can be expressed as shown in Figure 3.16.



$$s \leftarrow [s(n_p-128), \dots \dots., s(n_p-65)], [s(n_p-64), \dots \dots \dots, s(n_p-1)]$$

$$[s(n_p+16), \dots \dots \dots., s(n_p+79)], [s(n_p+80+16), \dots \dots \dots \dots \dots., s(n_p+80+79)], \dots.$$

Fig. 3.16 Symbol index alignment as per the cross-correlation output and frame structure

The transmitter and receiver oscillators are clocking at slightly different frequencies due to this there will be frequency offset between two devices. For the proper OFDM transmission, the transmitter and receiver must be synchronized with the same clocking. The short training preambles are used to estimate the frequency offset between. Let us consider $s(n)$ represents a sample of the short preamble and $s(n)$ corresponds to the sample $s(n+16)$ due to its cyclic property. The frequency offset df can be expressed as the average of the argument of the product $s(n)\overline{s}(n+16)$ for all the 16 samples which are given below [2]:

$$df = \frac{1}{16} arg\{ \sum_{n=0}^{N_{short}-1-16} s(n)\bar{s}(n+16)\} \tag{3.9}$$

Here N_{short} is the length of the short training samples. The calculated frequency offset df is applied to every long training preamble and data symbols as well to get the perfect synchronization. Once the frame alignment is done, the cyclic prefix samples are removed along with the short preamble, and data is forwarded to the next block. The "Stream to Vector" block collects this data and converts it to a parallel steam of complex symbols which become the input to the FFT. In the next block, the time domain samples are converted to the frequency domain using FFT block. After the FFT block data samples are sent to "OFDM Equalize Symbol" block for the channel estimation and phase offset correction. The Phase offset is introduced between the transmitter and receiver as the sampling time is not synchronized properly and symbol alignment in the receiver is not correct. Linear regression method is used to estimate the phase offset between OFDM samples and it is linear with the frequency. Let us consider, there are a set of N order pairs (x_0, y_0), (x_1, y_1), (x_2, y_2) …….. , (x_{N-1}, y_{N-1}) and through this points using LS method, a regression line $\hat{y} = a + b\hat{x}$ is computed. IEEE a/g/p defines 4 pilot sub-carriers which are used to compute the values of $a\ and\ b$ to equalize the phase offset of the OFDM data symbols in the physical layer frame. The final stage of the receiver is the decoding of the payload using "WiFi Decode Mac" block which performed the demodulation, de-interleaving, convolution decoding, puncturing and descrambling [1]-[3] to get a stream of byte.

CHAPTER 4

WIRELESS CHANNEL MODELS

In communications, a channel refers to the physical medium that forms the connection between the transmitter and receiver. For IEEE 802.11p, we make use of microwaves in the 5 GHz domain to form the connection between the transmitter and receiver in the wireless medium. There are many physical phenomena, like reflection, refraction, and diffraction that affects the signal passing through the wireless medium. However, there are many other effects in the wireless medium, like fading, that are best modeled mathematically or statistically. Generally, the actual channels in the environment are too complex to be modeled perfectly, hence we make use of statistical channels that have been developed in real world measurements. These channels models have been tested extensively and are accepted in the community as the standard models on which simulations can be performed.

4.1 Additive White Gaussian Noise (AWGN) Channel

The AWGN channel model is one of the most common channels which assumes that the noise is additive, has a very large bandwidth and is a superposition of many independent sources. In this channel model, it is assumed that there are no multi-path or Doppler shifts. It only assumes that there is additive noise that has a probability density function same as the Gaussian distribution given in Equation 4.1. Here, z is a random variable representing the noise amplitude, $p(z)$ represents the probability of the random variable, and σ^2 represents the variance of the Gaussian distributions [25].

$$p(z) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-z^2}{2\sigma^2}\right) \tag{4.1}$$

However, there is a delay when the signal passes through the AWGN channel, which is the path delay between the transmitter and the receiver. This is nothing but a timing offset that is easily corrected by the synchronization block in the receiver. The variations in the noise amplitude, as it is assumed to be independent and identically distributed, are only scalars for each sample. The LS estimator is extremely efficient fir such a channel, as the channel is also assumed to be stationary for the entire OFDM frame. The received symbols y_i are the sum of the transmitted symbols x_i and the noise symbols z_i, where z_i has values that are drawn from a Gaussian distribution, as shown in Equation 4.2 [25].

$$y_i = x_i + z_i \tag{4.2}$$

Evaluating the performance of the channel estimation and equalization schemes for a purely AWGN channel is unproductive in this this thesis, as the LS scheme has already been established to compensate such channels entirely. The major focus of this thesis is evaluating the performance of the schemes under much more complex channels.

4.2 Time Selective Fading Channel

A general channel model is represented in Equation 4.3. It shows that the received symbols $y(t)$ is the convolution of the time dependent impulse response of the channel $h(t,\tau)$ and the signal that travels through the channel $x(t-\tau)$ [13], [25].

$$y(t) = \int_{-\propto}^{\propto} h(t,\tau)\,.\,x(t-\tau)d\tau \tag{4.3}$$

In the time selective channel, the characteristics of the channel vary over time, which means that the signal can undergo different perturbations within a single OFDM frame. If there is a multi-path environment, then there is constructive and destructive interference of multiple reflections, and this is usually Rayleigh distributed in magnitude. The complex gain of such a channel for each path is represented in Equation 4.4. Where $a_n(t)$ is the attenuation and $\theta_n(t)$ is the phase rotation for the nth reflected path. Here, $a^{\sim}(t)$ is then the complex gain which is often assumed to have a Rayleigh distributed magnitude [25].

$$a^{\sim}(t) = \sum_{n=1}^{N} a^{\sim}(t) = \sum_{n=1}^{N} a_n(t)e^{j\theta_n(t)} \tag{4.4}$$

The channel impulse response can then be written as given Equation 4.5, where $\delta(t)$ is the Dirac delta function or unit-impulse function. Since $a^{\sim}(t)$ is time varying then the received signal is also time varying.

$$h^{\sim}(t) = a^{\sim}(t)\delta(t) \tag{4.5}$$

Coherence time is defined as the period over which the channel time response is stable. Coherence time is approximately calculated using Equation 4.6, where f_D is the maximum Doppler frequency and $2f_D$ is the Doppler spread in the channel. If the symbol duration is longer than the coherence time of a channel, then this might result errors within the symbol. The coherence time is also inversely proportional to Doppler spread, implicating that higher Doppler results in shorter coherence time [25].

$$T_{coherence} \approx \frac{1}{2f_D} \tag{4.6}$$

However, when the Doppler shift is minuscule compared to the bandwidth of the channel, then the effect of Doppler on the spectrum of the received signal is negligible, and such channels are called as frequency-flat channels. A channel model defined as "frequency-flat and time selective channel" is used commonly to test multi-path effects on a signal. Such a channel is also referred to as "flat-fading" channel. The performance of the LS channel estimation scheme under a flat-fading channel is favorable. Considering its low complexity, the LS scheme is an optimal choice for channel estimation when the Doppler shift is insignificant. But in highly time an frequency selective model, need to design an optimum channel estimation method that is adaptive with channel model and it should able to update it coefficients according to the channel behavior to minimize the error rate the in receiver.

4.3 Frequency Selective Fading Channel

A frequency selective channel will have different characteristics for the different frequency signals that are passing through the channel. The channel impulse response of such a channel is given in Equation 4.7 [25], where the complex gains $a_l^{\sim}$ are assumed to be constant, and τ_l represents the delay associated with the lth path.

$$h^{\sim}(t,\tau) = \sum_{l=1}^{L} a_l^{\sim}\,\delta(\tau - \tau_l) \tag{4.7}$$

This channel is time-invariant, but has a frequency dependent response, hence called as frequency selective channels. Coherence bandwidth is then defined as the frequency separation at which the attenuation of two frequency domain samples of the channel become decorrelated. If the bandwidth of a symbol is bigger than the coherence bandwidth, the channel transfer function has to be corrected to compensate the errors. The maximum delay that can be simulated to avoid any ISI is equal to the guard interval or cyclic prefix, in IEEE 802.11p which is 1.6 µs long.

4.4 Doubly Selective Fading Channel

The channel that is selective in both the time and frequency domain is called as a doubly selective channel. Consider the complex gains $\tilde{a_l}(t)$ for each path l to be a time-varying complex value, and τ_l to represent the delay associated with the lth path. Then the channel response can be given by Equation 4.8 [15].

$$h^{\sim}(t,\tau) = \sum_{l=1}^{L} a^{\sim}(t)\delta(\tau - \tau_l) \tag{4.8}$$

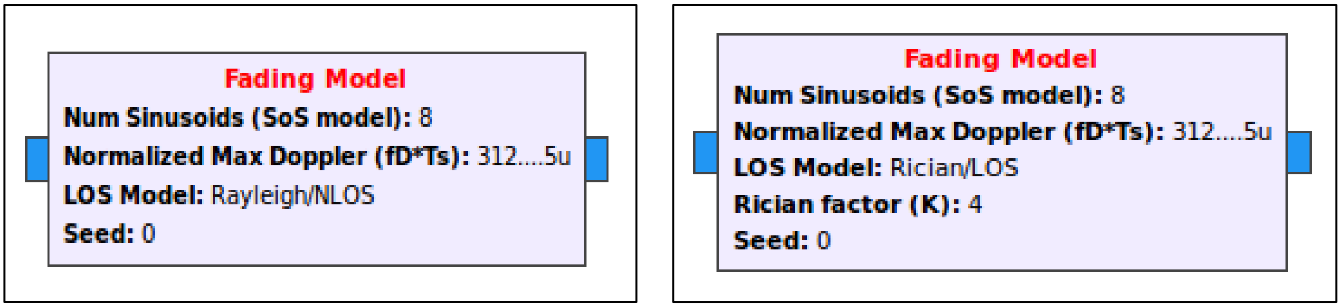
The continuous signal $x(t)$ is then transmitted over the noisy, doubly selective channel. In Equation 4.9, $y(t)$ is the received signal after passing through the doubly selective channel. $h^{\sim}(t,\tau)$ is the channel's time-varying impulse response, and $z(t)$ is the complex noise.

$$y(t) = \int_{-\propto}^{\propto} h^{\sim}(t,\tau)\,.\,x(t-\tau)d\tau + z(t) \tag{4.9}$$

The impulse response of the channel $h^{\sim}(t,\tau)$ will have complex coefficients that are relatively correlated to each other within the defined coherence time and coherence bandwidth of the channel. This means that the channel is selective in both time and frequency domain and hence such a channel is called as doubly selective channel. Most standardized mobile channel models belong to this category, and this thesis aims to improve the performance under such channels. The use of higher frequencies proportionally increases the effects Doppler shifts at the receiver, and combining them with smaller sub-carriers spacing in the OFDM symbols results in severe degradation of performance in the traditional channel estimation methods like LS.

4.5 Channels Modeling in GNU Radio Companion (GRC)

Multipath causes the propagation of radio signals transmitted from one antenna to another through two or more paths. In urban areas, fading arises due to the different height of the transmitting and receiving antennas as there is no particular line of sight (NLOS) to propagate the signal from one antenna to another [14]. But fading may happen even there is a presence of a Line of Sight (LOS) due to the reflection, scattering etc. of the transmitted signal from the ground and surrounding area objects. The receiving antenna will get a resultant signal which can vary widely in amplitude or even in phase based on the distribution of the intensity and the bandwidth of the transmitted signal. The amplitude variations of multipath fading signals are followed by Rayleigh and Rician distributions. Multipath propagation can be categorized as large scale and small-scale fading. Small scale fading can be expressed as rapid fluctuations of the amplitude or phase of the transmitted radio signal over a very short period of time or short travel distance [25]. Large scale fading is the consequence of signal attenuation due to the propagation over long distances and diffraction around large objects. Multipath small-scale fading effect causes few rapid changes in radio channels media such as rapid changes in signal strength level, time dispersion produced by multipath propagation path delays and random frequency modulation due to varying Doppler shifts of different multipath signals [17]. The small-scale fading can be expressed as a linear filter with the time varying impulse response of a wireless channel. Time variation is introduced due to the receiver motion in space and filtering nature is introduced due to the summation of amplitudes, delays of multiple arriving waves at any instant of time. However, based on the time delay spread small scale fading can be categorized as flat fading and frequency selective fading. As per Doppler spread small scale fading can be categorized as fast fading and slow fading. In GNU Radio, the fading model blocks exist for both Rayleigh and Rician distributions. The models approximate a fading channel using the sum of sinusoids method (SOS) for the number of anticipated multipath components. Generally, the number of sinusoids value is set to 8 as it is considered a decent value [26]. It takes other input parameters such as the normalized value of Doppler frequency shift, a line-of-sight (LOS) parameter which is false for Rayleigh distribution (NLOS) and true for Rician distribution (LOS), the Rician factor K and a random seed to the noise generators [21].

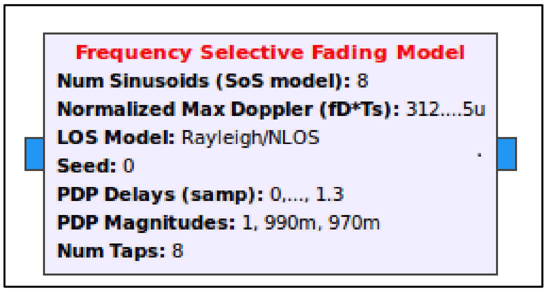


Fading Model
Num Sinusoids (SoS model): 8
Normalized Max Doppler (fD*Ts): 312....5u
LOS Model: Rayleigh/NLOS
Seed: 0

Fading Model
Num Sinusoids (SoS model): 8
Normalized Max Doppler (fD*Ts): 312....5u
LOS Model: Rician/LOS
Rician factor (K): 4
Seed: 0

Fig. 4.1 The Rayleigh (NLOS) and Rician (LOS) model block at GNU Radio [1]

Frequency selective fading model takes the input of power delay profile (PDP) to design a time varying multiple taps impulse response filtered channel model. In this fading model, PDP is given as a vector to provide the time delay of each impulse. The PDP magnitudes define the corresponding magnitudes to each time delay. A number of possible taps length of the filter is also assigned to simulate the channel fading behavior and to interpolate the power delay profile [21].



Frequency Selective Fading Model
Num Sinusoids (SoS model): 8
Normalized Max Doppler (fD*Ts): 312....5u
LOS Model: Rayleigh/NLOS
Seed: 0
PDP Delays (samp): 0,..., 1.3
PDP Magnitudes: 1, 990m, 970m
Num Taps: 8

Fig. 4.2 Frequency Selective Fading Model block at GNU Radio

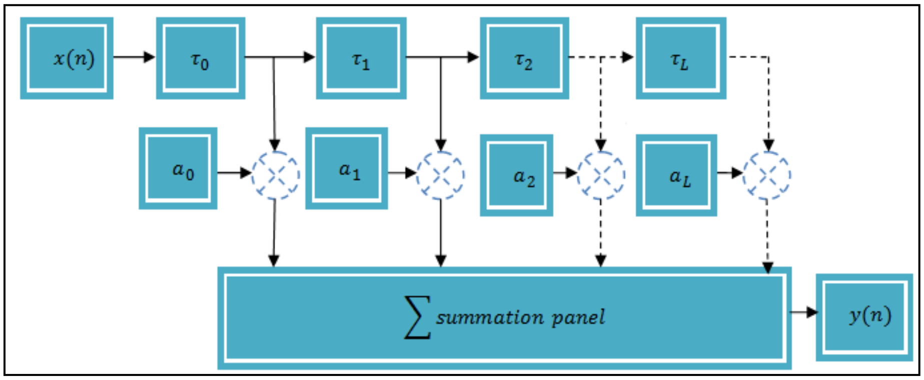
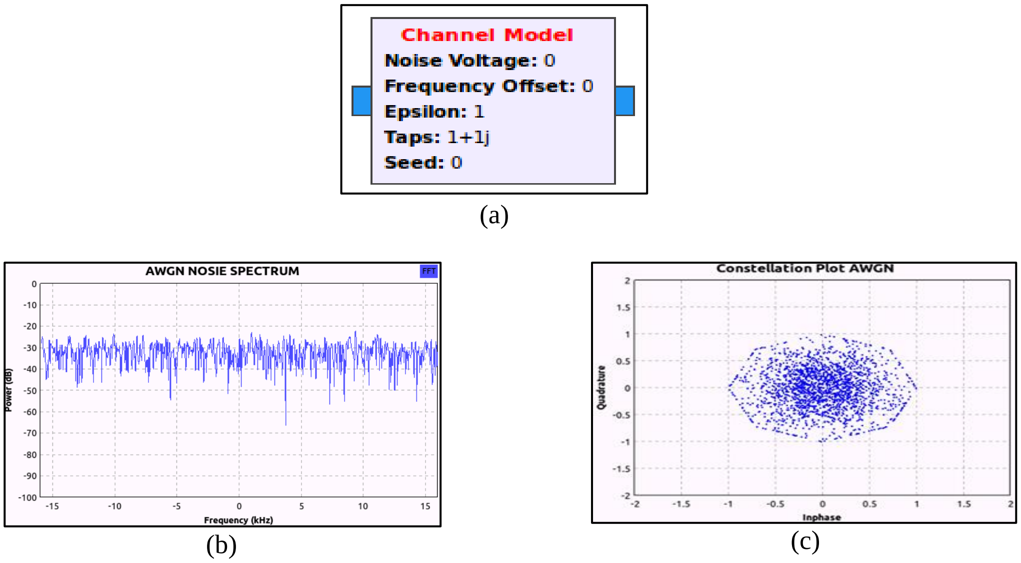


Fig. 4.3 L-tapped delay line FIR filter to describe the fading channel model characteristics. (τ_L is the delay of each tap during propagation, a_L is the tap gain which causes the magnitude variations of the impulses) [1]

Table 4.1 Wireless Channel Modeling Parameters for GRC [20], [26]

Parameter Setting	Descriptions
Normalized Max Doppler	Normalized max Doppler frequency is defined as $f_D T_s$, where f_D is the max Doppler frequency used in the fading model, $f_D = \frac{v f_c}{c}$ and T_s is the symbol period used in the OFDM system. It values vary within 0 to 1.
Number of Sinusoids (SOS)	Number of Sinusoid used in the frequency selective fading simulation
LOS model	Defines whether the fading model should include a line of sight component. LOS => Rician distribution, NLOS => Rayleigh distribution
Seed	A random number of generator seed for the noise source
Rician Factor (K)	Ratio between the deterministic signal power and the variance of multi-path
PDP Delays	A list of factional sample delays making up the power delay profile. Use to design a time varying multiple taps impulse response filter chrematistics channel model
PDP Magnitudes	A list of magnitudes corresponding to each delay time in the power delay profile
Num Taps	The length of the filter to interpolate the power delay profile

Additive White Gaussian Noise (AWGN) is the random basic noise model that exists in nature. This noise spectrum is uniform for all frequencies and the amplitude variation follows Gaussian Probability Density Function (PDF) [12]. Due to this reason, it has a uniform frequency spectrum shown in Figure 4.4 (b) and constellation diagram show the higher density at the center and less density outer from the center which is shown in Figure 4.4 (c). Additions of this noise to a transmitted radio signal during propagation through the wireless causes the signal amplitude variations. The channel model simulates AWGN with a simple static multipath environment using some parameters such as frequency offset; timing offset using epsilon, taps of FIR filter to add delay to multipath and a random number of generator seed for the noise source [20].



Channel Model
Noise Voltage: 0
Frequency Offset: 0
Epsilon: 1
Taps: 1+1j
Seed: 0

(a)

(b)

(c)

Fig. 4.4 (a) AWGN Channel Model Block at GNU Radio (b) AWGN noise spectrum (c) Constellation diagram of AWGN [1]

CHAPTER 5

PROPOSED PRECISE FER COMPUTATION OF IEEE 802.11 a/g/p TRANSCEIVER PROTOTYPE

Generally, NIST and YANS error rate models are so much popular to analyze the performance of an IEEE 802.11 standard WLAN. These models compute SNR based on the parameters used in simulation model such as noise figure, noise floor, path loss model etc. The FER is computed based on the applied modulation scheme, coding rate, and frame size. Let us assume that convolution coding and hard decision Viterbi decoding are applied to the data symbols and transmitted over an AWGN channel model. In IEEE 802.11 a/g/p the bit error probability is upper bound by following Chernoff Bound as per the NIST error rate model [18]:

$$P_b < \frac{1}{k} \sum_{d=d_{free}}^{\infty} B_d P(d)\,, P(d) = [4p(1-p)]^{\frac{d}{2}} \tag{5.1}$$

Here, d_{free} is the free distance of the convolution code, multiplication factor B_d is the total number of non-zero information bits that are in error while an incorrect path is chosen for the specified hamming distance d, $P(d)$ is the probability of selecting a weight d output sequence as the transmitted code sequence, p is the simply the un-coded BER and k is the number of information bits per clock cycle.

5.1 The precise estimation of SNR and $\frac{E_b}{N_0}$

The signal to noise power ratio (SNR) in AWGN is estimated using the equation given below [13]:

$$SNR[dB] = P_{rec} - N_F - N_1 \tag{5.2}$$

Here, P_{rec} is the received signal power at the receiving antenna, N_F is the noise figure and $N_1$is the thermal noise floor. YANS model estimated the ratio between the energy per bit (E_b) and noise power spectral density (N_0) as per the Equation 5.3 [13].

$$\frac{\mathrm{E_b}}{\mathrm{N_o}} = \mathrm{SNR}\frac{\mathrm{B_t}}{\mathrm{R_b}} \tag{5.3}$$

Here B_t is the noise bandwidth and R_b is the raw bit rate calculated by the number of bits per OFDM symbol over the symbol interval time. This model does not account for the reduction of energy due to the cyclic prefix (CP) and the reduction of the net energy due to the pilot carriers which are basically not used for carrying information. On the other hand, NIST model estimated as given in Equation 5.4 [13].

$$\frac{\mathrm{E_b}}{\mathrm{N_0}} = \frac{\mathrm{SNR}}{\log_2(\mathrm{M})} \tag{5.4}$$

Here, M= number of signal constellations. NIST model does not account for the ratio of used sub-carriers in OFDM system and CP.

However, in order to calculate more accurate BER and FER, more precise estimation of $\frac{\mathrm{E_b}}{\mathrm{N_0}}$ are required which given at Equation 5.5 [1]:

$$\frac{E_b}{N_0} = SNR\left(\frac{N_{FFT}}{N_{DC} + N_{PC}}\right) x \left(\frac{1}{N_{CBPS}}\right) \tag{5.5}$$

Here, N_{FFT} is the FFT length, N_{DC} is the number of data sub-carriers, N_{PC} is the number of pilot carriers and N_{CBPS} is the number of coded bits per symbol in each OFDM sub-carrier.

5.2 Precise Estimation of FER in AGWN Channel Model

In section 5.1, the more accurate estimation of $\frac{E_b}{N_0}$ is computed considering the reduction of energy due to the cyclic prefix, reduction of the net energy due to the pilot carriers and the ratio of used sub-carriers in the OFDM system. The FER can be expressed as given at Equation 5.6 [1].

$$FER = 1 - (1 - P_b)^L \tag{5.6}$$

Here, L is the aggregated MAC frame size. The standard size of L is basically 441 bytes. BER (P_b) for the M-PSK modulated signal in AWGN channel can be expressed as given below [24]:

$$P_{M-PSK} = \frac{2}{\max(log_2 M, 2)} \sum_{k=1}^{\max({}^{M}/_{4},1)} Q(\sqrt{\frac{2E_b log_2 M}{N_0}} \sin\frac{(2k-1)\pi}{M}) \tag{5.7}$$

For M=2, the above equation reduces to BER of BPSK modulation $P_{BPSK} = Q(\sqrt{\frac{2E_b}{N_0}})$, here the error function Q is expressed as given below [24]:

$$Q(x) = \frac{1}{2\pi}\int_x^{\infty} \exp\left(-\frac{t^2}{2}\right)dt = \frac{1}{2}erfc\left(\frac{x}{\sqrt{2}}\right) \tag{5.8}$$

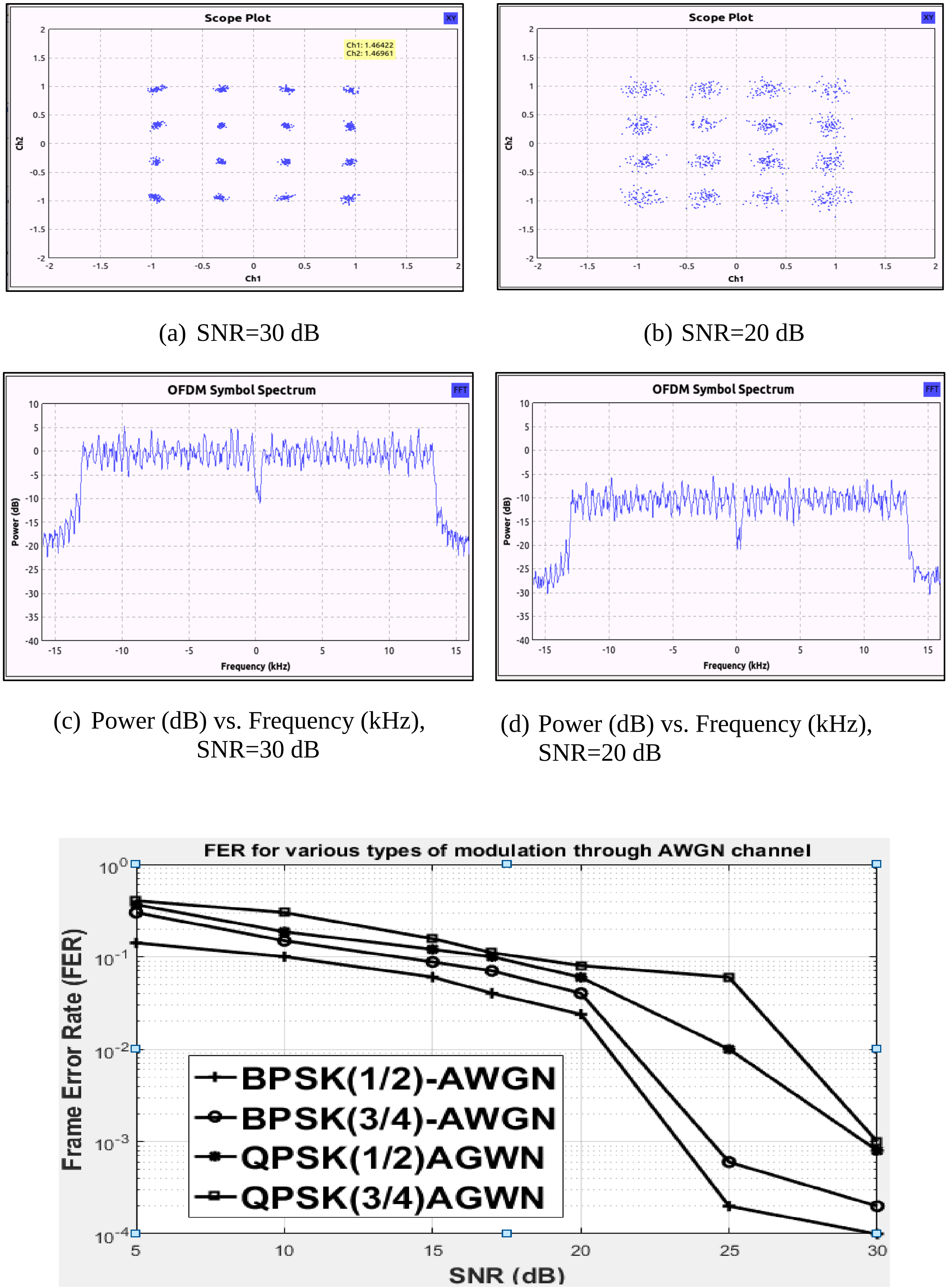
BER (P_b) for the M-QAM modulated signal in AWGN channel can be expressed as given below [24]:

$$P_{M-QAM} = \frac{4}{log_2M}\ Q\left[\sqrt{\frac{3(log_2M)E_b}{(M-1)N_0}}\right] \tag{5.9}$$

To estimate the performance of the receiver FER is calculated for different values of SNR (dB) expressed in terms of $\frac{E_b}{N_0}$ as given as Equation 5.5.

5.3 FER Computation of IEEE 802.11 a/g/p Transceiver for AWGN Channel Model

Let us consider that, the data are transmitted over AWGN channel by choosing 16-QAM modulation scheme for each sub-carrier with rate 24 Mbps. The outputs of the simulated model at GNU Radio during real-time data transmission over AWGN channel using USRP devices are given in Fig. 5.1. According to the above equations no. (5.9) and (5.6), both BER and FER are the functions of SNR and $\frac{E_b}{N_0}$. As a result, with decreasing SNR (dB) the coded and mapped constellations points that are carrying the data symbols mentioned at Figure 5.1 {(a), (b)} are spreading from the center and almost try to overlap with each other due to the effect of AWGN for changing in value of SNR from 30dB to 20dB. It causes the ISI and affects to maintain the orthogonal property between the OFDM sub-carriers. To reduce the FER channel estimation and equalization should be investigated at the receiver end. Since the signal power decreases the OFDM spectrum power decreases but AGWN has the lowest impact on sub-carrier orthogonality as the spectrum shape remains almost similar as shown as Figure 5.1 {(c), (d)}. The Figure 5.1 {(e), (f)} illustrates the FER vs. SNR (dB) curve. There, it is noticed that with the increment of coding rate or even data rate the FER values were getting higher. As a result, to reduce FER better estimation and equalization is required in the receiver end.



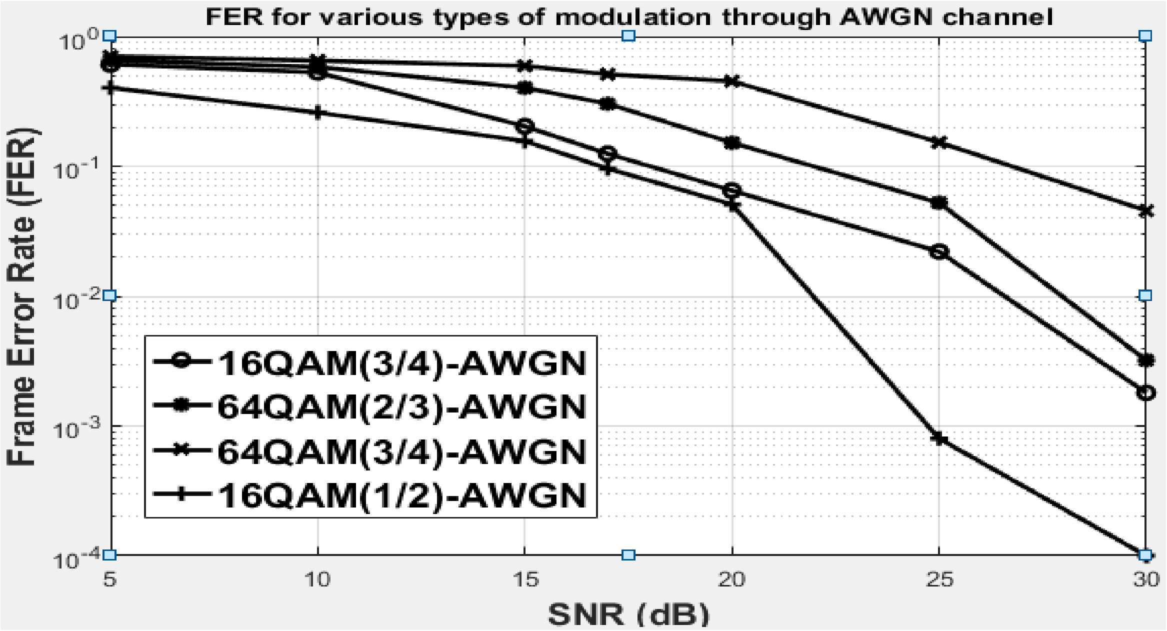
(a) SNR=30 dB

(b) SNR=20 dB

(c) Power (dB) vs. Frequency (kHz), SNR=30 dB

(d) Power (dB) vs. Frequency (kHz), SNR=20 dB

(e) FER vs. SNR (dB), M-PSK



(f) FER vs. SNR (dB), M-QAM

Fig. 5.1 [(a), (b)] 16-QAM (1/2) constellation points for SNR=30 dB and 20 dB respectively with 24 Mbps data rate; **[(c), (d)]** OFDM spectrum for SNR = 30 dB and 20 dB respectively; [**(e), (f)]** FER performance of OFDM for M ∊ {2,4}-PSK and M ∊ {16, 64}-QAM with different coding rate ∊ {1/2, 2/3, ¾} over AWGN channel

5.4 FER Computation of IEEE 802.11 a/g/p Transceiver for Mobile Wireless Fading Channel Model

Mobile wireless channel considered as the flat fading Rayleigh channel with Jake spectrum. A particular frame will be successfully transmitted when it is fully in the inter-fading state. Let us consider some random variable given below: t_i = inter fading duration, T_i is the mean value of t_i, t_f = fading duration, T_f is the mean value of t_f, t_{FD}= frame duration and T_{FD} is the mean value of t_{FD}. A particular threshold value of the received signal power is used to determine the frame either is in the fading state or inter fading state. If any part of the frame is in the fading state, then the frame delivery will be unsuccessful. The threshold is known as fading margin $m = \frac{R_{req}}{R_{rms}}$, here R_{req} is the required received power level and R_{rms} is the mean received power level. FER can be written as [12]:

$$FER = 1 - \frac{T_i}{T_i + T_f} P(t_i > T_{FD}) \tag{5.10}$$

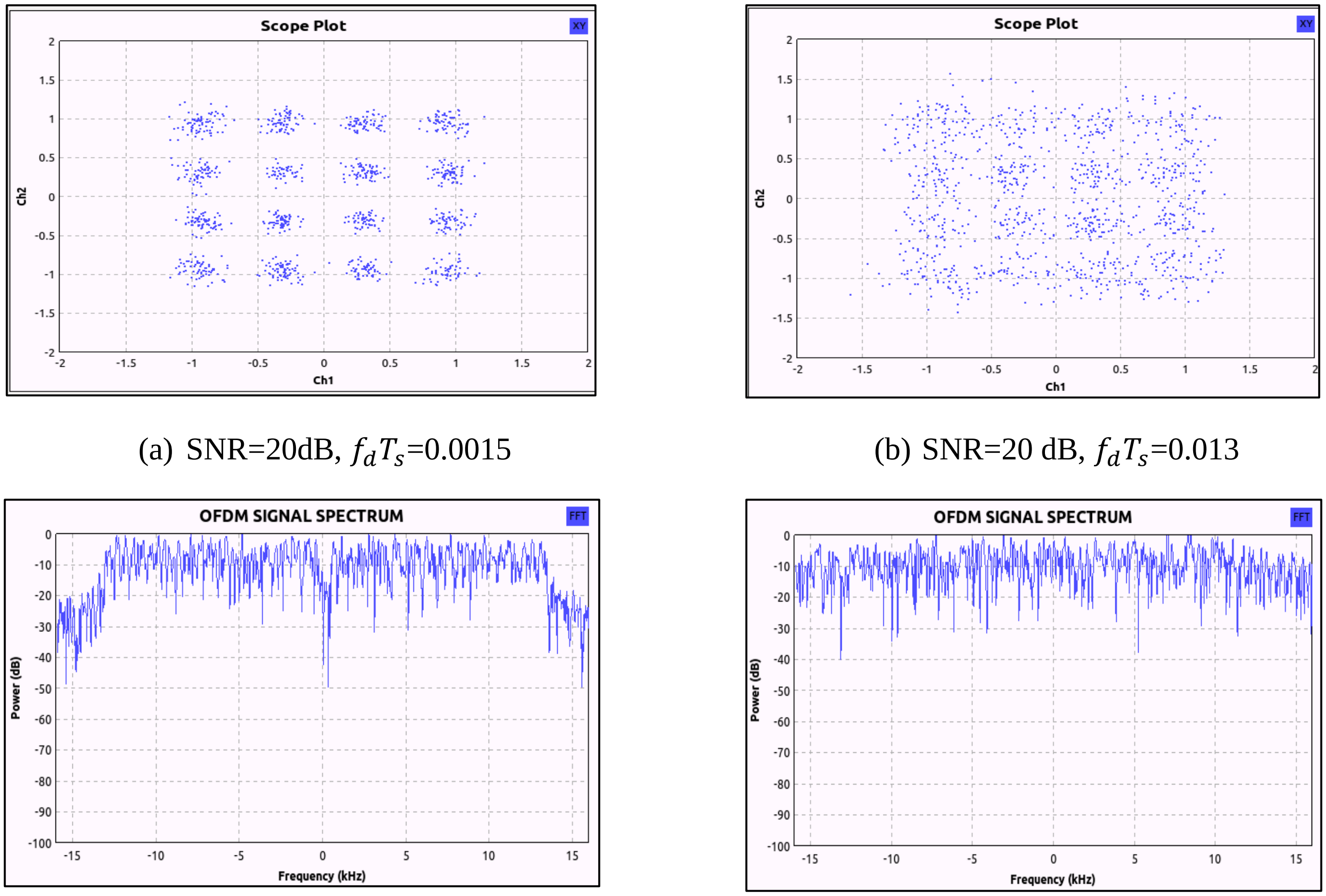
For both t_i and t_f, exponential distributions are assumed and the probability of $P(t_i > T_{FD})$ can be written as $e^{-\frac{T_{FD}}{T_i}}$. The FER in Rayleigh fading channel can be expressed as given below [12]:

$$FER = 1 - e^{(-m - f_m\sqrt{2\pi m}T_{FD})} \tag{5.11}$$

f_m is the maximum Doppler frequency, defined as $f_m = \frac{v}{\lambda} = \frac{f_c v}{c}$. Here, $v =$ velocity of the mobile (ms^{-1}), f_c= carrier frequency 5.89 GHz and $c = 3x10^8\ ms^{-1}$. Normalized Doppler frequency can be defined as $f_D = f_m T_s,$ here T_s is the symbol period.

As a result, FER in wireless channel depends on Doppler frequency shift, fading margin and frame duration. Again, to estimate the performance of the receiver, FER is calculated for different values of SNR (dB) where the selected data rate is 24 Mbps, the chosen modulation scheme is 16-QAM and transmission medium is the time and frequency selective fading channel. The channel is simulated by introducing a tapped delay line FIR filter with an exponentially decaying average power delay profile (PDP). The number of taps given in FIR filter is 8 and tap characteristics is chosen as Rayleigh distributed (NLOS) and Rician distributed (LOS) exclusively. A particular normalized Doppler shift ($f_D T_s$) is added to get the relative characteristics while the receiver is not stationary. In case of Rician distribution, the transmitted signal experienced a dominant nonfading signal component due to the presence of a line of sight (LOS) path between the transceiver. In this context, many weaker random multipath signals arriving at different angles produced by reflection, diffraction or scattering effect are superimposed on the dominant signal. The ratio between the deterministic signal power and the variance of multipath known as the Rician factor K is set equal to 4 during simulation.

The outputs of the simulated model at GRC during real-time data transmission over AWGN and frequency selective fading channel model (Rayleigh Distributed-NLOS, Rician Distributed-LOS) using USRP devices are given respectively at Figure 5.2 and 5.3**.** The constellations diagrams at Figure 5.2 and 5.3 {(a), (b)} shows that for the increasing Doppler shifts the coded and mapped information data symbols carried by the constellation's points of 16-QAM are spreading from the center and overlapping to each other. The OFDM signal spectrums are also losing the ideal shapes meaning that the orthogonal property between multiple sub-carriers are affected. Therefore, proper channel estimation is required for the error rate and ISI minimization. The FER vs. SNR (dB) curves given at Figure 5.2 and 5.3 showed comparatively less FER for less puncturing rate {M-PSK (1/2), M-QAM (1/2)} as well as for the lower data rate. Once the coding puncturing rate is increased with changing the higher modulation scheme to get the higher data rate, FER is increased accordingly.

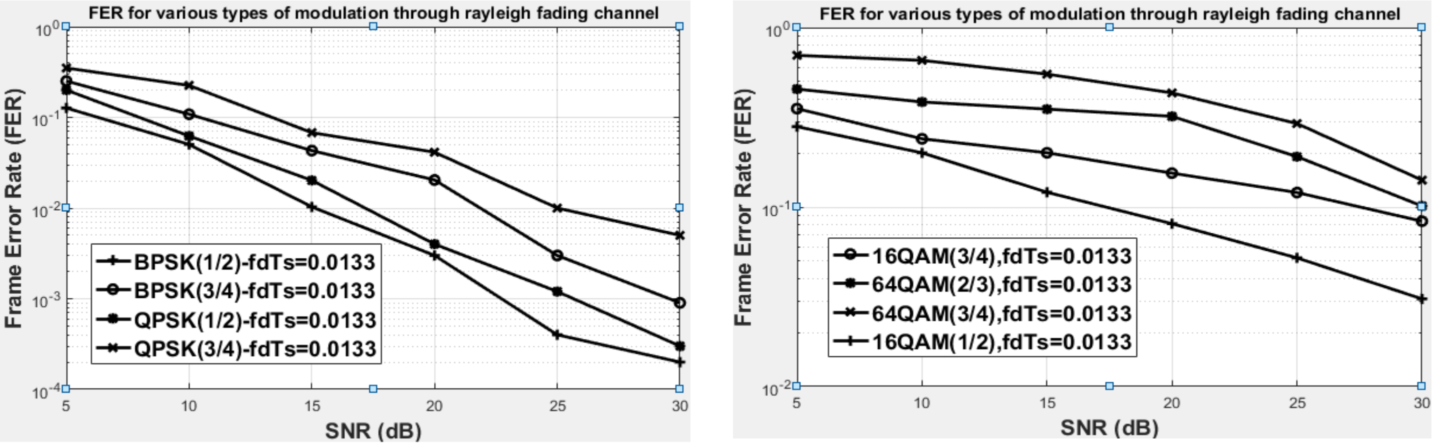


(a) SNR=20dB, $f_d T_s$=0.0015

(b) SNR=20 dB, $f_d T_s$=0.013

(c) Power (dB) vs. Frequency (kHz), SNR=20 dB, $f_d T_s$=0.0015

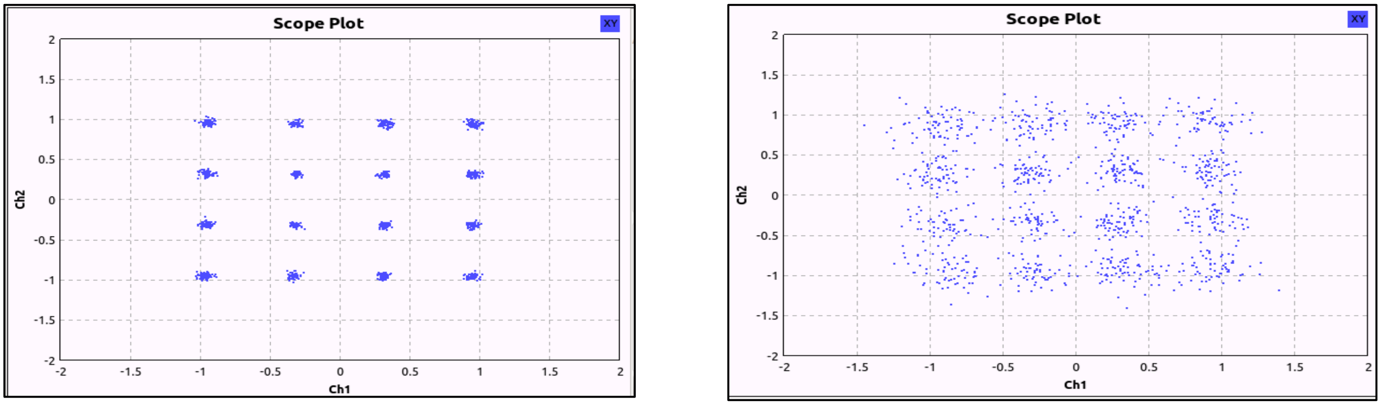
(d) Power (dB) vs. Frequency (kHz), SNR=20 dB, $f_d T_s$=0.0133



(e) FER vs. SNR (dB), M-PSK, $f_d T_s$=0.0133

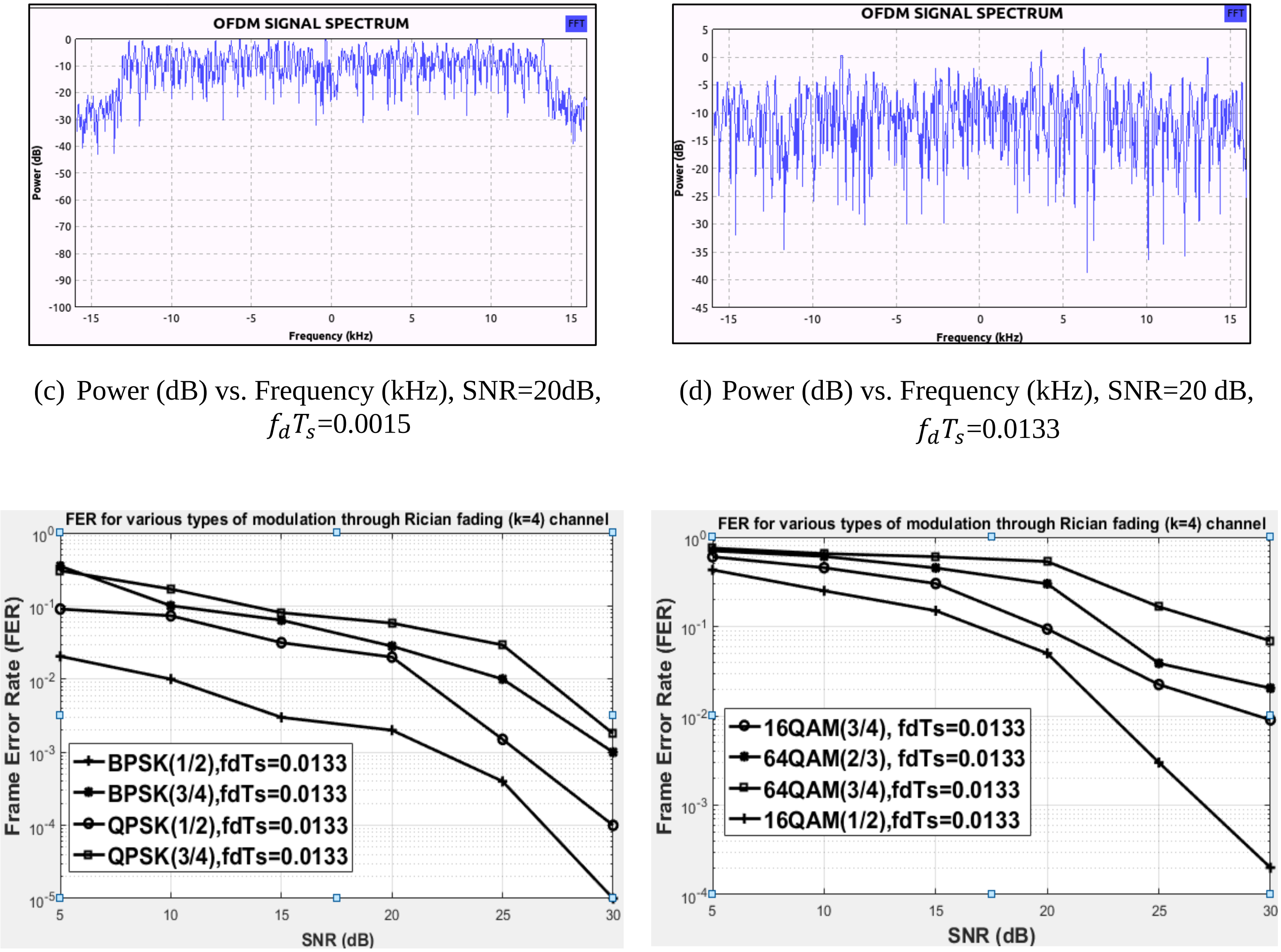
(f) FER vs. SNR (dB), M-QAM, $f_d T_s$=0.0133

Fig. 5.2 [(a), (b)] 16-QAM Constellation Points **[(c), (d)]** OFDM symbol spectrum [for Rayleigh distributed frequency selective fading channel having SNR=20 dB and $f_d T_s$=0.0015, 0.0133 respectively] [(**e), (f)]** FER performance of OFDM for M ∊ {2,4}-PSK and M ∊{16, 64}-QAM with different coding rate ∊{1/2, 2/3, ¾}, $f_d T_s$=0.0133



(a) SNR=20dB, $f_d T_s$=0.0015

(b) SNR=20 dB, $f_d T_s$=0.0133



(c) Power (dB) vs. Frequency (kHz), SNR=20dB, $f_d T_s$=0.0015

(d) Power (dB) vs. Frequency (kHz), SNR=20 dB, $f_d T_s$=0.0133

(e) FER vs. SNR (dB), M-PSK, $f_d T_s$=0.0133

(f) FER vs. SNR (dB), M-QAM, $f_d T_s$=0.0133

Fig. 5.3 **[(a), (b)]** 16-QAM Constellation Points **[(c), (d)]** OFDM symbol spectrum [for Rician distributed frequency selective fading channel having SNR=20 dB, Rician Factor K =4 and normalized Doppler frequency $f_d T_s$=0.0015, 0.0133] [**(e), (f)]** FER performance of OFDM for M ∊ {2,4}-PSK and M ∊{16, 64}-QAM with different coding rate ∊ {1/2, 2/3, ¾}, $f_d T_s$=0.0133

CHAPTER 6

PROPOSED CHANNEL ESTIMATION (CE) TECHNIQUES

The Channel Estimation (CE) process is the approximation and characterizing the effect of physical medium on the transmitted data symbols. It allows the implementation of coherent demodulation technique. CE technique basically uses the pilot reference signals of OFDM to estimate the channel characteristics. This technique minimizes the Mean Square Error (MSE) in order to get more accurate data symbols at the receiver end by utilizing the less computational method. Based on the help of the position of neighboring pilot sub-carries both time and frequency-domain interpolation are required to get the CE of each data sub-carrier in time-frequency plane as the pilots are assigned combined block-comb allocation method. If the receiver has the prior information of the transmitting data over the channel, it can utilize that knowledge in the estimation process to get the more accurate data. Predefined training symbols are used at the beginning of each IEEE 802.11 standard WLAN data packets with safety guard interval to get the convergence of the estimator.

6.1 Least Square (LS) Channel Estimation (CE) for static AGWN channel

IEEE 802.11 a/g/p uses two long preambles $T_1 \; and \; T_2$ in the Least Square (LS) CE process. Let us consider for each OFDM frame the received long preambles of a particular OFDM symbol are denoted as $Y_{T_1} \; and \; Y_{T_2}$ and predefined frequency domain long training symbols are $X_{T_1} and \; X_{T_2}$. Then the LS CE transfer function is expressed as given below [4], [9]:

$$\widetilde{H_{LS}}(k) = \frac{1}{2}\left(\frac{Y_{T_1}}{X_{T_1}} + \frac{Y_{T_2}}{X_{T_2}}\right) \qquad (6.1)$$

Above estimation is used to equalize the rest of the OFDM symbols. LS CE algorithm is expressed as the ratio between the output $Y(k)$ and input data $X(k)$ sequence. The LS estimation of the channel transfer function $\widetilde{H_{LS}}(k)$ can be written as given below [10], [17]:

$$\widetilde{H_{LS}}(k) = \frac{Y(k)}{X(k)} = Y(k)X^{-1}(k) = \left[\frac{y_0}{x_0}, \frac{y_1}{x_1}, \dots., \frac{y_{N-1}}{x_{N-1}}\right]^T = \frac{X(k)H(k) + W(k)}{X(k)}$$

$$= H(k) + \frac{W(k)}{X(k)} \qquad (6.2)$$

The LE CE of the channel transfer function at pilot positions can be expressed as [1]:

$$\widetilde{H}_{LS,P}(k_p) = \frac{Y_P(k_p)}{X_P(k_p)} = X_P{}^{-1}(k_p)Y_P(k_p) = \left[\frac{Y_P(0)}{X_P(0)}\ \frac{Y_P(1)}{X_P(1)}, \dots \dots .., \frac{Y_P(N_P-1)}{X_P(N_P-1)}\right]^T \qquad (6.3)$$

Here, a number of pilot signal $N_P = 4$, $Y_P(k_p)$ is the received signal in the pilot positions (-21, -7, 7, 21) and $X_P(k_p)$ is the known transmitted pilot signal. The LS pilot estimated transfer function $\widetilde{H}_{LS,P}(k_p)$ is used to determine the channel estimate transfer function at each neighboring sub-carrier. As the pilots are assigned in both time and frequency domain, therefore, 2-D channel interpolation is required to compute the channel transfer function at each sub-carrier $\widetilde{H_{Ls}}(k)$ from $\widetilde{H}_{LS,P}(k_p)$ that can be expressed as [1]:

$$\widetilde{H_{Ls}}(k) = \sum_{n_p\ k_p} F[(n,k); (n_p, k_p)]. \widetilde{H}_{LS,P}(k_p) \qquad (6.4)$$

Here $F[(n,k); (n_p, k_p)]$ is the estimation filter which coefficients are needed to be estimated to get the estimation at a particular OFDM sub-carrier. The term (n,k) defines the position of particular OFDM channel that has to be estimated, n denotes the OFDM symbol and k is the sub-carrier index. The filter coefficients can be estimated by using a low-pass filter interpolation (LPI) technique at both time and frequency domain.

A signal $x(k)$ having bandwidth B can be restored properly from its samples $x(n)$ using an ideal low pass filter with bandwidth $W = \frac{F_S}{2}$ only if the sampling frequency $F_S \geq 2B$. The channel transfer function $\widetilde{H_{LS}}(k)$ is sampled with the rate $f_{s,time} = \frac{F_{SR}}{f_t}$, where F_{SR} is the symbol rate (symbols/sec) and f_t is the pilot separation along with time domain. The channel maximum Doppler frequency f_m defines the rate of change of the channel across the time direction. A low-pass filter can be used for the time-domain interpolation by avoiding aliasing only when $f_m \leq \frac{F_{SR}}{2f_t}$. For the frequency domain interpolation channel transfer function $\widetilde{H_{LS}}(k)$ is sampled with the frequency $f_{s,freq} = f_f.\Delta f$, where f_f is the pilot's separation along with frequency domain, Δf is the sub-carrier spacing. The time period is calculated from the sampling time $\frac{1}{f_f.\Delta f} = \frac{T_{SWCP}}{f_f}$, where T_{SWCP} is the OFDM symbol period without the cyclic prefix; it gives the periodic replicas of the channel impulse response $h(n,\tau)$.

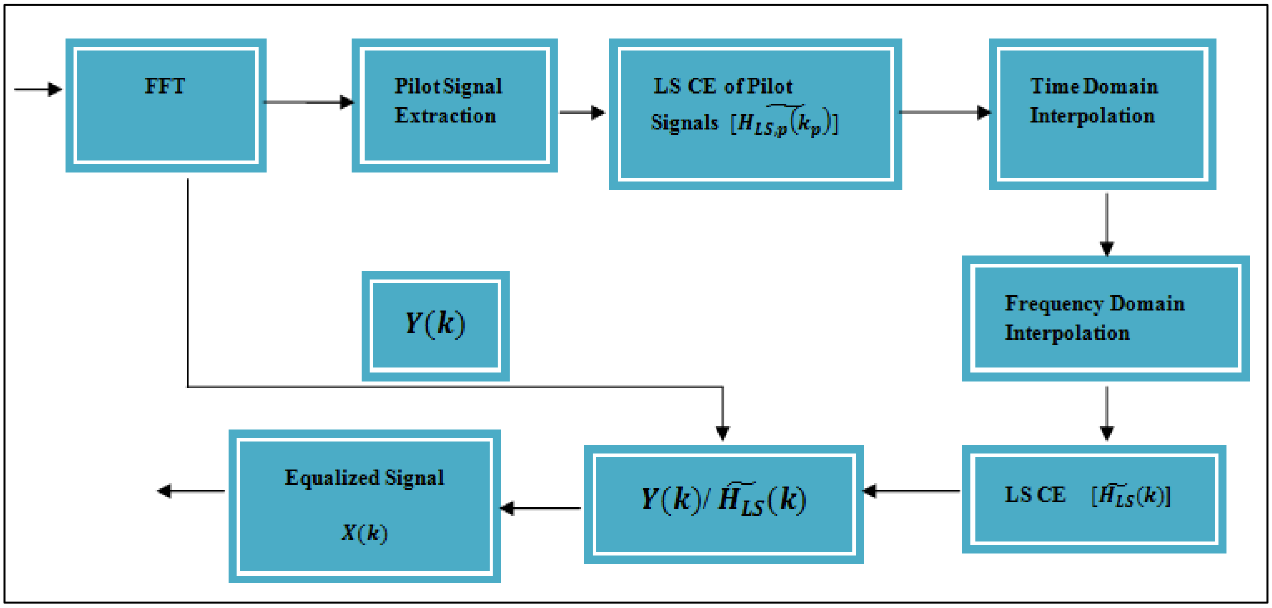

FFT
Pilot Signal Extraction
LS CE of Pilot Signals $[\widetilde{H_{LS,p}(k_p)}]$
Time Domain Interpolation
Frequency Domain Interpolation
$Y(k)$
LS CE $[\widetilde{H_{LS}}(k)]$
$Y(k)/\widetilde{H_{LS}}(k)$
Equalized Signal
$X(k)$

Fig. 6.1 LS CE from pilot signal using LPI technique [1]

Therefore, a low pass filter can be used along with frequency domain for channel interpolation without aliasing only when maximum delay spread of the channel $\tau_{max} \leq \frac{T_{SWCP}}{f_f}$.

6.2 LS Channel Estimator Performance Evaluations for Different Channel Models

The Figure 6.1 gives the FER performance comparisons for IEEE 802.11 a/g/p transceiver modeled in GRC after applying LS CE technique during data symbol transmission over AWGN and Rayleigh distributed (NLOS) channel models. From the Figure 6.2 it has been observed that LS CE gives better performance only for the static AWGN channel as FER decreases suggestively for the AWGN channel that represents in blue dotted line. On the other hand, Doppler frequency changing environment there are no changes in FER enhancement more over it affects the receiver performance severely for high spread Doppler shift. The channel response derived from above equations are used for the equalization by considering that the channel is stationary for the whole packet. If the transmitter or receiver is in motion then there will be the presence of frequency and phase offset of the transmitting signal. As a result, at the receiver end, LS estimation is not efficient to minimize the FER.

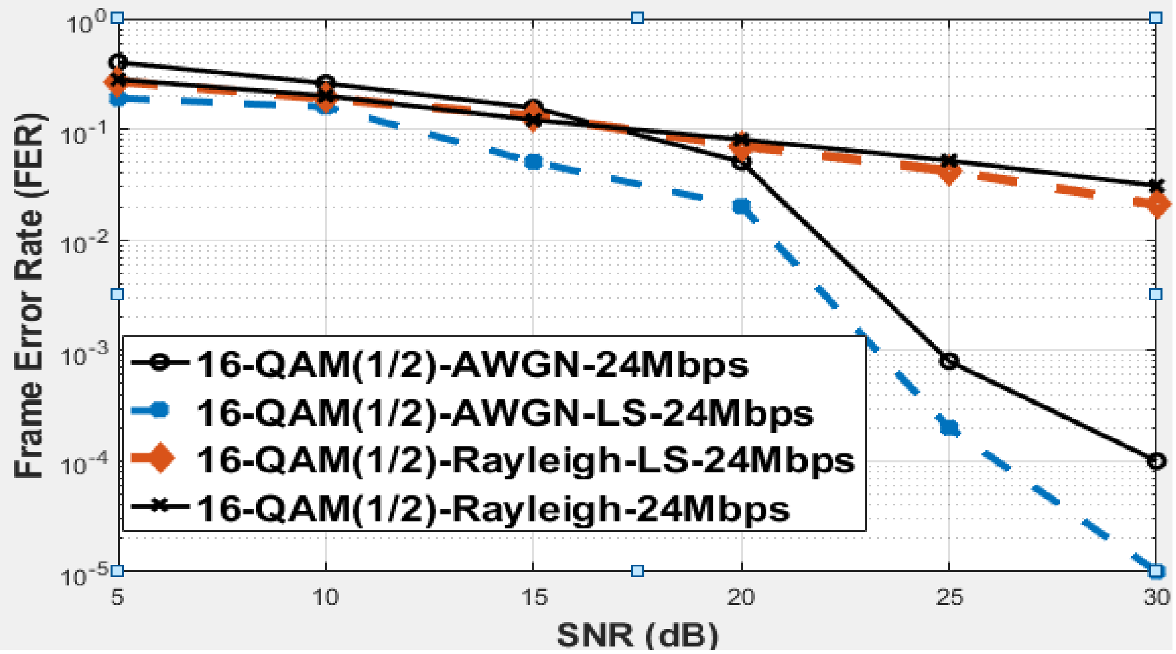


Fig. 6.2 FER performance of LS CE for 16-QAM over AWGN and Rayleigh fading channel ($f_d T_s = 0.0133$)

6.3 Adaptive Least Mean Square (LMS) Channel Estimator for Mobile Wireless Fading Channel

To improve the estimator performances iterative based CE method has been investigated known as Least Mean Square (LMS) algorithm. The rapidly changing time varying channel requires adaptive estimation process as it has to continuously up-to-date its filter coefficients following the channel conditions. Adaptive algorithm enhances the performance by minimizing the mean square error (MSE) between the desired equalizer output and the actual equalizer output [8]. The unknown multipath fading channel has to be equalized and properly estimated by using an adaptive filtering algorithm which has the knowledge of prior transmission of data known as a periodic training sequence. This adaptive process is continued until the MSE of LMS algorithm is zero and equalizer are converged. Let us consider that signal $x(n)$ is transmitted over the time varying wireless fading channel which is described through an L-tapped FIR filter having time varying complex coefficients a_{nL}. The received signal at the channel filter output can be expressed as:

$$y(n) = \sum_{l=0}^{L-1} a(n,l)x(n-l) + w(n) \tag{6.5}$$

Here, $x(n-L)$ is the complex symbol strained from the x constellation of the L paths at time instant $n-L$, $a(n,l)$ is the complex channel tapped coefficients and $w(n)$ is the AWGN noise. The LMS algorithm can be mathematically expressed as the vector notation as given below:

The transmitted signal, $x(n) = [x(n)\ \ x(n-1)\ \ x(n-2)\ \ \ldots\ldots\ldots\ldots\ldots.\ x(n-L)]^T$, the received signal, $y(n) = [y(n)\quad y(n-1)\quad y(n-2)\ldots\ldots\ldots\ldots\ldots.\ y(n-L)\ ]^T$, the filter complex coefficients at different tap positions, $a_n = [a_{0n}\quad a_{1n}\quad a_{2n}\quad \ldots\ldots\ldots\ldots\ldots..\ a_{Ln}]^T$, AWGN vector $w(n) = [w(n)\quad w(n-1)\quad w(n-2)\ldots.w(n-L)]^T$, the adaptive filter tap weights is $\widehat{a_n}$. The received signal can be written in vector form as $y(n) = a_n^T x(n)$. The error signal that has to be minimized adaptively and update the filter tap weights accordingly in the estimation process can be expressed as given below:

$$e(n) = d(n) - y(n) = d(n) - {a_n}^T x(n) \tag{6.6}$$

Here, $d(n)$ is the predefined known transmitted training symbol that basically used for channel equalization and error approximation. The error signal $e(n)$ requires to compute the cost function $E[e(n)e^*(n)]$ or MSE function and minimization of the cost function is done by applying the steepest descent algorithm. By detecting the prior known training sequence adaptive filter can compute the possible error signal and minimize the cost function by driving the tap weights through iteration process until the next training sequence is sent. At time instant n the mean square error (MSE) can be calculated as given below:

$$|e(n)|^2 = d^2(n) + a_n^T x(n)x^T(n)a_n - 2d(n)x^T(n)a_n \tag{6.7}$$

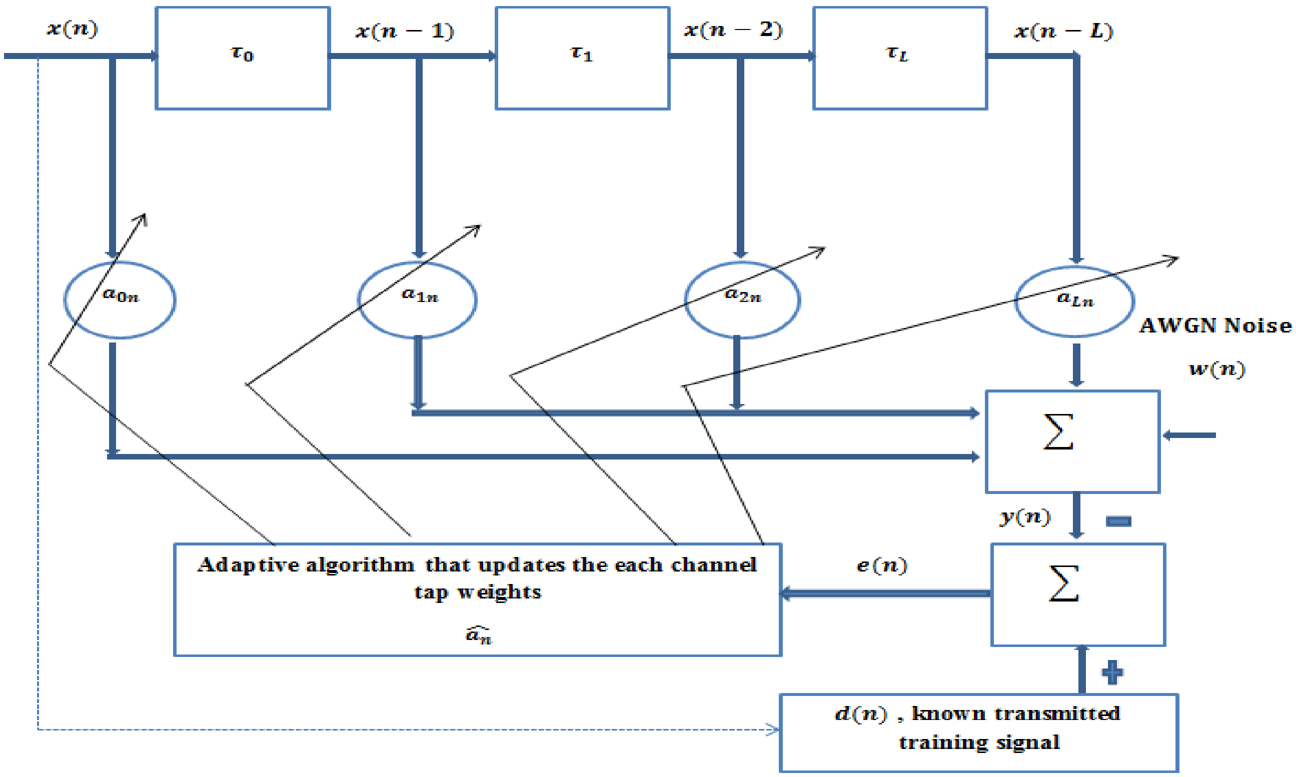


Fig. 6.3 Adaptive CE using LMS algorithm [1]

Now taking the expected mean of the above equation MSE can be written as [1]:

$$E[|e(n)|^2] = E[d^2(n)] + a_n^T E[x(n)x^T(n)]a_n - 2E[d(n)x^T(n)]a_n \qquad (6.8)$$

The cross-correlation vector P_{CC} between the desired response $d(n)$ and the input signal $x(n)$ can be expressed as:

$$P_{CC} = E[d(n)x(n)] = E[d(n)x(n) \quad d(n)x(n-1) \;\; d(n)x(n-2) \ldots \ldots \ldots d(n)x(n-L)\;]^T \qquad (6.9)$$

The input correlation matrix R_{CR} is defined as given below:

$$R_{CR} = E[x(n)x^T(n)]$$

$$= \begin{bmatrix} x^2(n) & x(n)x(n-1) & x(n)x(n-2) \ldots \ldots \ldots . \ldots & x(n)x(n-L) \\ x(n-1)x(n) & x^2(n-1) & x(n-1)x(n-2) \ldots \ldots . . & x(n-1)x(n-L) \\ x(n-L)x(n) & x(n-L)x(n-1) & x(n-L)x(n-2) \ldots \ldots . . & x^2(n-L) \end{bmatrix} \qquad (6.10)$$

The MSE can be written as given in Equation 6.11.

$$MSE = E[x^2(n)] + a_n^T R_{CR} a_n - 2P_{CC}^T a_n \tag{6.11}$$

Minimization of this MSE equation in terms of tap weight a_n is conceivable to adaptively tune the equalizer to curtail the ISI in the received signal. Let the cost function $J(a_n)$ as the MSE which is basically the function of tap gain vector a_n. Minimization process is done by taking the derivative and putting the derivative equal to zero as given below:

$$\frac{\partial}{\partial a_n} J(a_n) = -2P_{CC}^n + 2R_{CR}^{nn} a_n = 0$$

$$P_{CC}^n = R_{CR}^{nn} \widehat{a_n}, \quad \hat{a}_n = R_{CR}^{nn^{-1}} P_{CC}^n \tag{6.12}$$

Eventually, the Minimum Mean Square Error (MMSE) or the optimal cost function in terms of optimal tap gain vector $\hat{a}_n$ can be written as given below:

$$J_{opt} = J(\hat{a}_n) = E[d(n)d^*(n)] - {P_{CC}^n}^T \hat{a}_n \tag{6.13}$$

The LMS is computed through the n iterative processes of minimization of the cost function to get the convergence state of the equalizer and adaptive adjustment of tap weight vector which can be summarized by the equations given below:

$$y(n) = a^T(n)x(n) \tag{6.14}$$

$$e(n) = x(n) - y(n) \tag{6.15}$$

$$a_L(n+1) = a_L(n) - \alpha e^*(n) y_L(n) \tag{6.16}$$

L is the number of delays taps in the fading channel model, α is the step size which defines the rate of convergence and stability of the LMS algorithm, $a_L(n+1)$ denotes the weight vector to be computed at iteration $(n+1)$. The stability of α is defined within the range $0 < \alpha < \frac{2}{\sum_{j=1}^{L} \lambda_j}$, here λ_j is the jth Eigen value of the convergence matrix R_{CR}^{nn}.

6.4 The Adaptive LMS Algorithm Implementation in GNU Radio Companion

The theoretical implementation process for adaptive LMS estimator that has been discussed at section 6.3 can be applied in GNU Radio for IEEE 802.11 a/g/p transceiver. The GNU Radio IEEE 802.11 a/g/p signal processing blocks have a C++ API for this LMS decision directed equalizer [20]. The API documentation is standardized at file "**frame_equalizer_impl.h"**. It uses the LMS adaptive filter as equalizer named as "**lms.h".** It is a set of weights to correlate against the inputs, and a decision is then made from this output which is known as error function minimization step by changing the tap weight vector. The error in the decision is used to update the tap weight vector. This tap updated continues till the error function approaches to zero and estimator tap coefficients get convergence for the mobile multi-path fading channel conditions. The "**lms.h"** library is added to the "**frame_euqlizer_impl.h"** file according to the C++ syntax **#include "equalizer/lms.h"** [27].

This file also maps the constellations according to the chosen modulation schemes using the function **message_port_register_out(pmt::mp("symbols"))** and the chosen modulation schemes are called using **constellation_{modulation_schemes}** name as given below:

```
 d_bpsk = constellation_bpsk::make();

 d_qpsk = constellation_qpsk::make();

d_16qam = constellation_16qam::make();

d_64qam = constellation_64qam::make();
```

The options for mapping the symbols in the real and imaginary planes are BPSK, QPSK and 16/64-QAM modulation schemes. The equalizer mode can be switched from LS to LMS mode in the live channel. To do this following function is used.

```
frame_equalizer_impl::set_algorithm(Equalizer algo) {

	switch(algo) {

	case LS:

		dout << "LS" << std::endl;
```

```
        d_equalizer = new equalizer::ls();

        break;

    case LMS:

        dout << "LMS" << std::endl;

        d_equalizer = new equalizer::lms();

        break;
```

The adaptive **lms.h** takes input of the mapped transmitting complex symbols, known portion of transmitted symbols, filer taps and chosen modulation schemes as per the format given below under its public base and under private base it takes the input of variable double SNR value . The portion of C++ function is illustrated below:

```
class lms: public base {

public:

    virtual void equalize(gr_complex *in, int n, gr_complex *symbols, uint8_t *bits,
boost::shared_ptr<gr::digital::constellation> mod);

private:

    double get_snr();
```

The final raw C++ adaptive filter signal processing code is written at “**lms.cc”**. It gives the tap weight vector of filter coefficients according to the minimization condition of the error function by using a simple for loop and condition checking that described elaborately in Appendix A, Section A.5.

The whole algorithm process for data symbol transmitting, channel model designing and data receiving including LMS channel estimation and equalization is described through the Figure 6.4. The transmitter and receiver stack at GRC are illustrated at Figure 6.5 and 6.6 respectively.

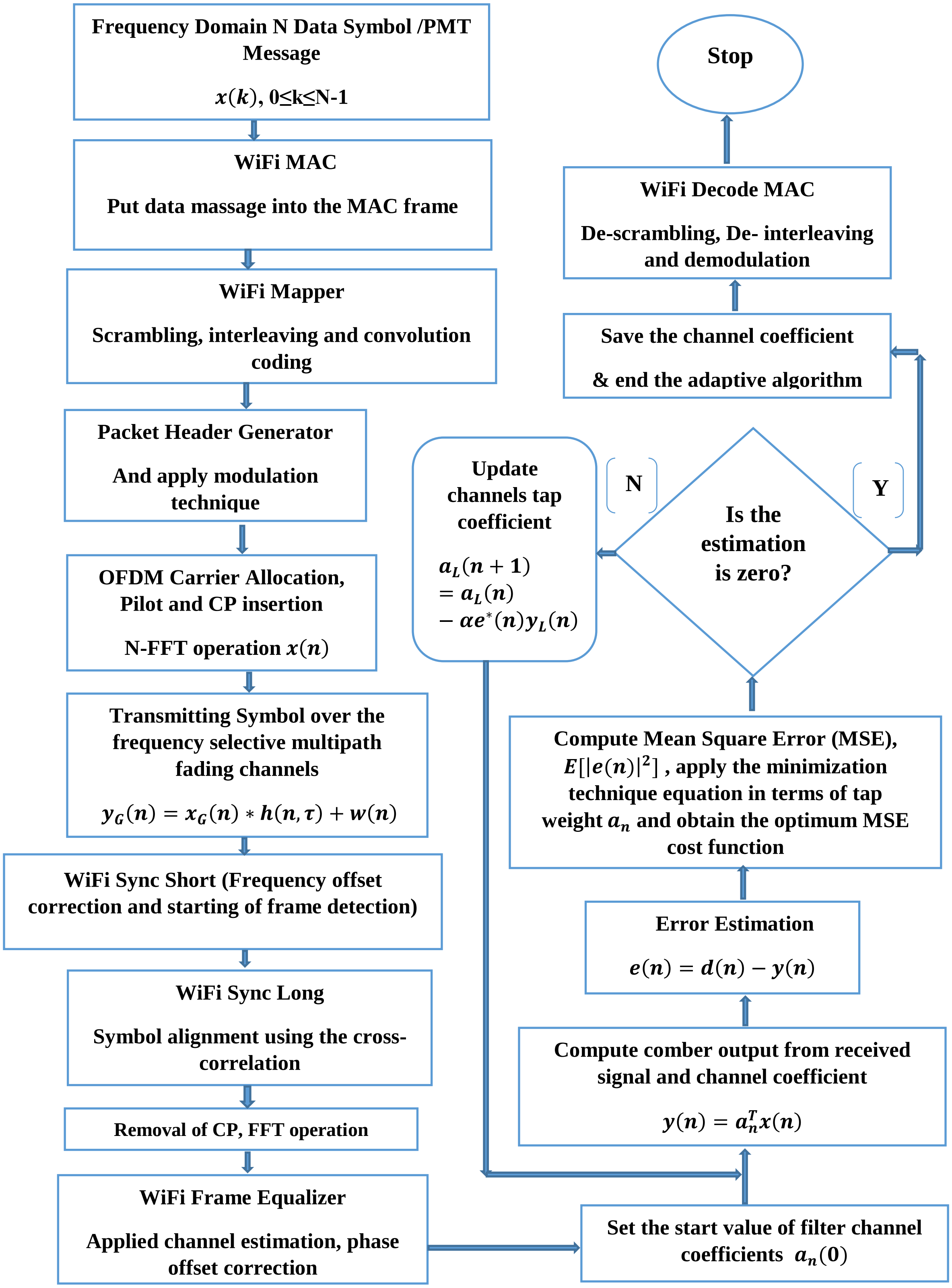
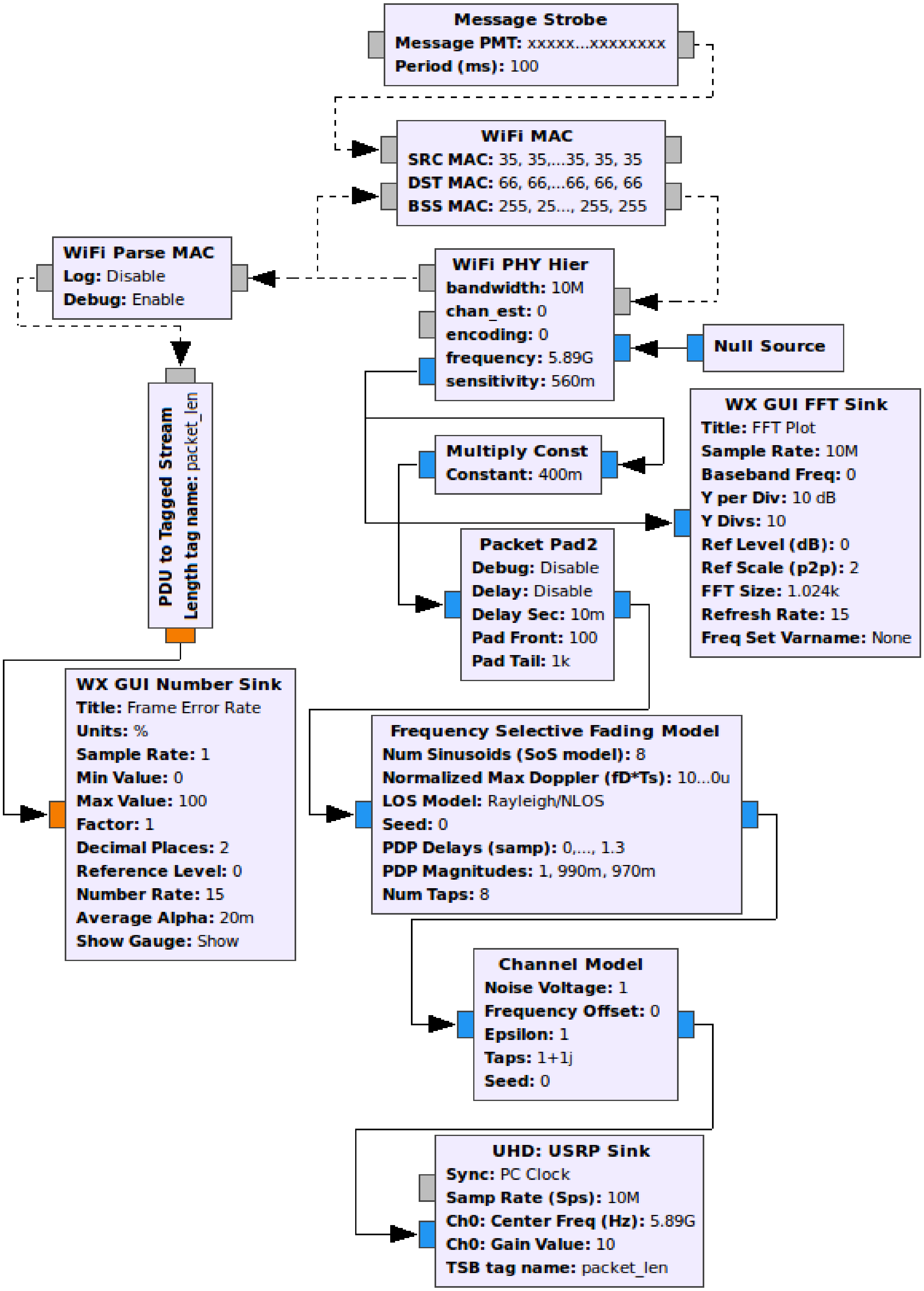

Frequency Domain N Data Symbol /PMT Message
$x(k)$, 0≤k≤N-1
WiFi MAC
Put data massage into the MAC frame
WiFi Mapper
Scrambling, interleaving and convolution coding
Packet Header Generator
And apply modulation technique
OFDM Carrier Allocation, Pilot and CP insertion
N-FFT operation $x(n)$
Transmitting Symbol over the frequency selective multipath fading channels
$y_G(n) = x_G(n) * h(n,\tau) + w(n)$
WiFi Sync Short (Frequency offset correction and starting of frame detection)
WiFi Sync Long
Symbol alignment using the cross-correlation
Removal of CP, FFT operation
WiFi Frame Equalizer
Applied channel estimation, phase offset correction
Set the start value of filter channel coefficients $a_n(0)$
Compute comber output from received signal and channel coefficient
$y(n) = a_n^T x(n)$
Error Estimation
$e(n) = d(n) - y(n)$
Compute Mean Square Error (MSE), $E[|e(n)|^2]$, apply the minimization technique equation in terms of tap weight a_n and obtain the optimum MSE cost function
Is the estimation is zero?
N
Y
Update channels tap coefficient
$a_L(n+1) = a_L(n) - \alpha e^*(n) y_L(n)$
Save the channel coefficient
& end the adaptive algorithm
WiFi Decode MAC
De-scrambling, De- interleaving and demodulation
Stop

Fig. 6.4. Implementation algorithm of IEEE 802.11 transceiver using LMS CE



Message Strobe
Message PMT: xxxxx...xxxxxxxx
Period (ms): 100

WiFi MAC
SRC MAC: 35, 35,...35, 35, 35
DST MAC: 66, 66,...66, 66, 66
BSS MAC: 255, 25..., 255, 255

WiFi Parse MAC
Log: Disable
Debug: Enable

WiFi PHY Hier
bandwidth: 10M
chan_est: 0
encoding: 0
frequency: 5.89G
sensitivity: 560m

Null Source

PDU to Tagged Stream
Length tag name: packet_len

WX GUI FFT Sink
Title: FFT Plot
Sample Rate: 10M
Baseband Freq: 0
Y per Div: 10 dB
Y Divs: 10
Ref Level (dB): 0
Ref Scale (p2p): 2
FFT Size: 1.024k
Refresh Rate: 15
Freq Set Varname: None

Multiply Const
Constant: 400m

Packet Pad2
Debug: Disable
Delay: Disable
Delay Sec: 10m
Pad Front: 100
Pad Tail: 1k

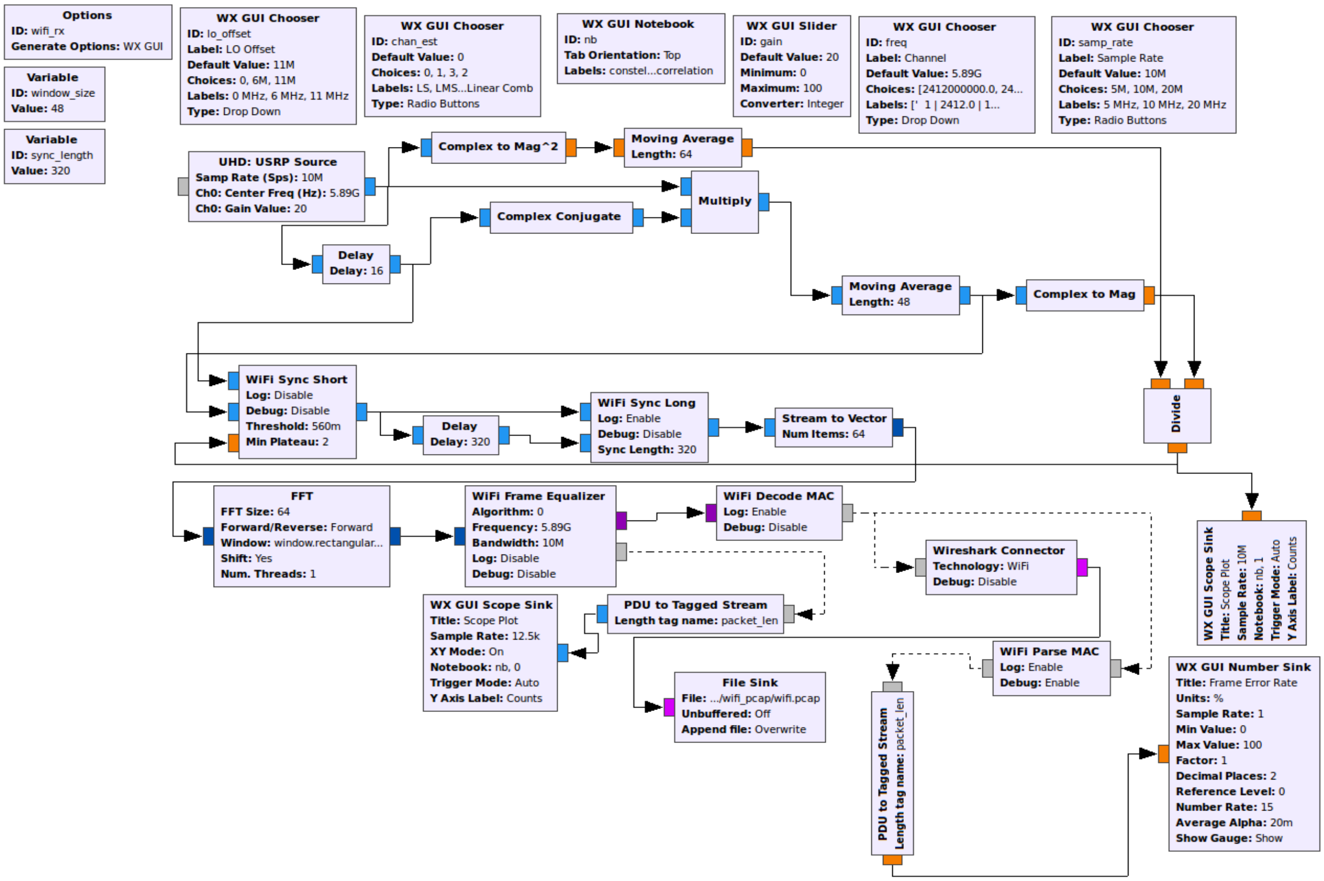
WX GUI Number Sink
Title: Frame Error Rate
Units: %
Sample Rate: 1
Min Value: 0
Max Value: 100
Factor: 1
Decimal Places: 2
Reference Level: 0
Number Rate: 15
Average Alpha: 20m
Show Gauge: Show

Frequency Selective Fading Model
Num Sinusoids (SoS model): 8
Normalized Max Doppler (fD*Ts): 10...0u
LOS Model: Rayleigh/NLOS
Seed: 0
PDP Delays (samp): 0,..., 1.3
PDP Magnitudes: 1, 990m, 970m
Num Taps: 8

Channel Model
Noise Voltage: 1
Frequency Offset: 0
Epsilon: 1
Taps: 1+1j
Seed: 0

UHD: USRP Sink
Sync: PC Clock
Samp Rate (Sps): 10M
Ch0: Center Freq (Hz): 5.89G
Ch0: Gain Value: 10
TSB tag name: packet_len

Fig. 6.5 IEEE 802.11p Transmitter in GRC



Options
ID: wifi_rx
Generate Options: WX GUI

Variable
ID: window_size
Value: 48

Variable
ID: sync_length
Value: 320

WX GUI Chooser
ID: lo_offset
Label: LO Offset
Default Value: 11M
Choices: 0, 6M, 11M
Labels: 0 MHz, 6 MHz, 11 MHz
Type: Drop Down

WX GUI Chooser
ID: chan_est
Default Value: 0
Choices: 0, 1, 3, 2
Labels: LS, LMS...Linear Comb
Type: Radio Buttons

WX GUI Notebook
ID: nb
Tab Orientation: Top
Labels: constel...correlation

WX GUI Slider
ID: gain
Default Value: 20
Minimum: 0
Maximum: 100
Converter: Integer

WX GUI Chooser
ID: freq
Label: Channel
Default Value: 5.89G
Choices: [2412000000.0, 24...
Labels: [' 1 | 2412.0 | 1...
Type: Drop Down

WX GUI Chooser
ID: samp_rate
Label: Sample Rate
Default Value: 10M
Choices: 5M, 10M, 20M
Labels: 5 MHz, 10 MHz, 20 MHz
Type: Radio Buttons

UHD: USRP Source
Samp Rate (Sps): 10M
Ch0: Center Freq (Hz): 5.89G
Ch0: Gain Value: 20

Complex to Mag^2

Moving Average
Length: 64

Multiply

Complex Conjugate

Delay
Delay: 16

Moving Average
Length: 48

Complex to Mag

Divide

WiFi Sync Short
Log: Disable
Debug: Disable
Threshold: 560m
Min Plateau: 2

Delay
Delay: 320

WiFi Sync Long
Log: Enable
Debug: Disable
Sync Length: 320

Stream to Vector
Num Items: 64

FFT
FFT Size: 64
Forward/Reverse: Forward
Window: window.rectangular...
Shift: Yes
Num. Threads: 1

WiFi Frame Equalizer
Algorithm: 0
Frequency: 5.89G
Bandwidth: 10M
Log: Disable
Debug: Disable

WiFi Decode MAC
Log: Enable
Debug: Disable

Wireshark Connector
Technology: WiFi
Debug: Disable

WX GUI Scope Sink
Title: Scope Plot
Sample Rate: 10M
Notebook: nb, 1
Trigger Mode: Auto
Y Axis Label: Counts

WX GUI Scope Sink
Title: Scope Plot
Sample Rate: 12.5k
XY Mode: On
Notebook: nb, 0
Trigger Mode: Auto
Y Axis Label: Counts

PDU to Tagged Stream
Length tag name: packet_len

File Sink
File: .../wifi_pcap/wifi.pcap
Unbuffered: Off
Append file: Overwrite

WiFi Parse MAC
Log: Enable
Debug: Enable

PDU to Tagged Stream
Length tag name: packet_len

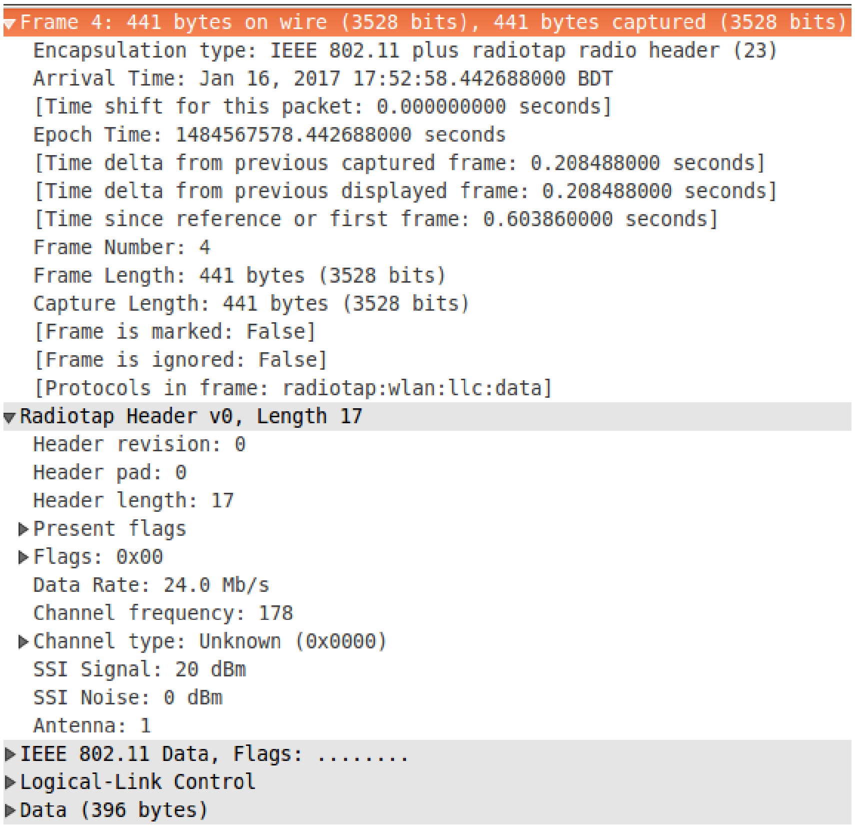
WX GUI Number Sink
Title: Frame Error Rate
Units: %
Sample Rate: 1
Min Value: 0
Max Value: 100
Factor: 1
Decimal Places: 2
Reference Level: 0
Number Rate: 15
Average Alpha: 20m
Show Gauge: Show

Fig. 6.6 IEEE 802.11p Receiver in GRC

The Figure 6.5 illustrates the hierarchy of an IEEE 802.11p Transmitter stack in GNU Radio Companion. The “Message Strobe” blocks creates the input messages that are to be transmitted in a container called as Polymorphic Types (PMT). PMT are opaque data types that are designed as generic containers of data that can be safely passed around between blocks and threads in GNU Radio. They are heavily used in the stream tags and message passing interfaces. The size of the message and the repetition rate can also be varied to test the system at different packet rates. The WiFi MAC block puts the data message into the MAC frame according to the frame structure defined in the standard. These MAC frames are then passed to the hierarchical block called as the “WiFi PHY Hier” which contains the PHY layer blocks of the standard. Then the time domain samples are transmitted over the doubly selective multi-path fading channels model blocks and fed to the “UHD: USRP sink” block that transmitted the samples to the USRP hardware through the Giga Bit Ethernet connection.

The mac_out port of “WiFi PHY Hier” block is fed to the “WiFi Parse MAC” block that filters out the incoming MAC frames and extracts only the data frames to calculate the frame error rate of the flow-graph. The frame error rate is finally displayed in “WX GUI Number Sink” as a bar graph. The other port carrier, containing the complex symbol streams is used to display the data spectrum in frequency domain using the sink plot “WX GUI FFT Sink” block.

The Figure 6.6 describes the IEEE 802.11p receiver section which detects the starting of frames based on the correlation plateau mechanism between the short preamble and accordingly decodes the frames. The “USRP source block” is the UHD block that interfaces to communicate with the receiver USRP Radio Frequency (RF) port (RX2) in order to get the transmitted OFDM time domain signal that has transmitted over 5 GHz channel by the transmitter USRP. Rest of the signal processing task is discussed elaborately at Chapter 3 under section 3.9.



```
▼Frame 4: 441 bytes on wire (3528 bits), 441 bytes captured (3528 bits)
   Encapsulation type: IEEE 802.11 plus radiotap radio header (23)
   Arrival Time: Jan 16, 2017 17:52:58.442688000 BDT
   [Time shift for this packet: 0.000000000 seconds]
   Epoch Time: 1484567578.442688000 seconds
   [Time delta from previous captured frame: 0.208488000 seconds]
   [Time delta from previous displayed frame: 0.208488000 seconds]
   [Time since reference or first frame: 0.603860000 seconds]
   Frame Number: 4
   Frame Length: 441 bytes (3528 bits)
   Capture Length: 441 bytes (3528 bits)
   [Frame is marked: False]
   [Frame is ignored: False]
   [Protocols in frame: radiotap:wlan:llc:data]
▼Radiotap Header v0, Length 17
   Header revision: 0
   Header pad: 0
   Header length: 17
  ▶Present flags
  ▶Flags: 0x00
   Data Rate: 24.0 Mb/s
   Channel frequency: 178
  ▶Channel type: Unknown (0x0000)
   SSI Signal: 20 dBm
   SSI Noise: 0 dBm
   Antenna: 1
▶IEEE 802.11 Data, Flags: ........
▶Logical-Link Control
▶Data (396 bytes)
```

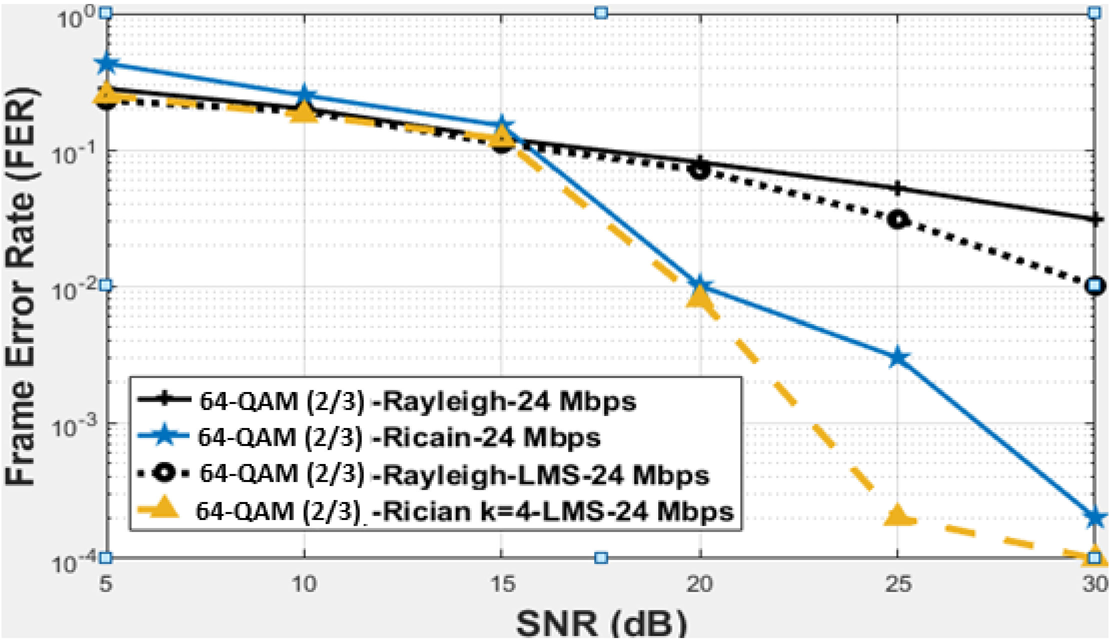
Fig. 6.7 IEEE 802.11p Frames captured by Wireshark Packet Analyzer

The Figure 6.7 gives the standard IEEE 802.11 frame structure captured by using Linux “tcpdump” tool from the Ethernet interface of the Linux host machine during data symbol processing over the USRP device from GNU Radio software.

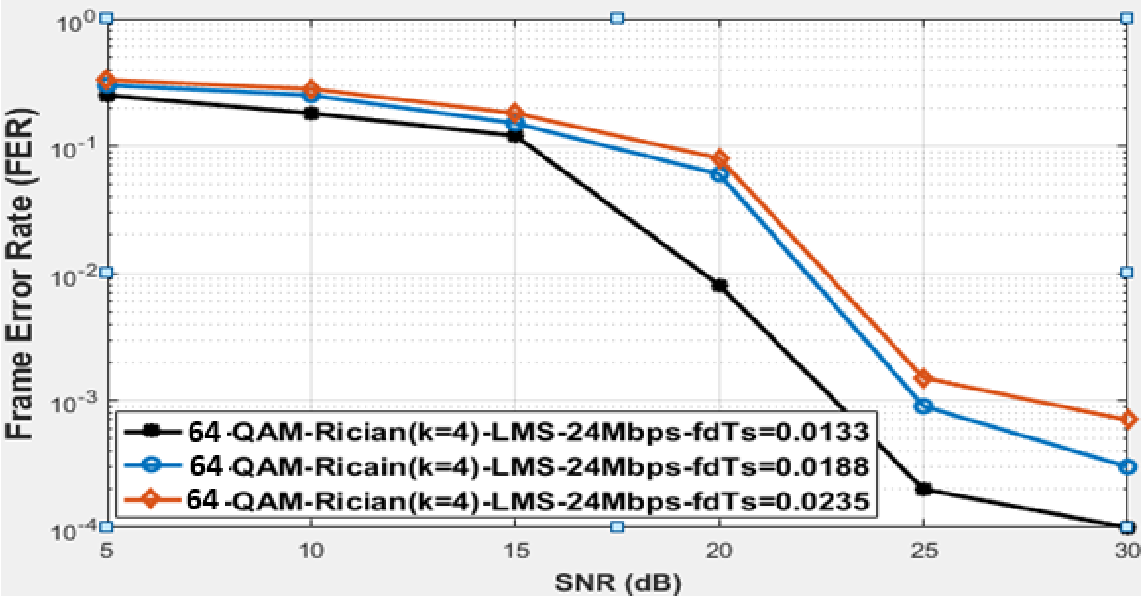
6.5 The LMS CE Performance Evaluation for Mobile Wireless Fading Channel Model

The Figure 6.8 (a) illustrates the FER performance after applying LMS CE technique during data symbol transmission over Rayleigh (NLOS) and Rician (LOS) distributed fading channels. It has been noticed that the adaptive LMS CE technique reduced FER at the receiver that are represented in dotted line in the Figure 6.8 (a) for both the channel models with compared to the not equalized FER which are represented in solid line in the Figure 6.8 (a). Here the receiver has given significantly less FER when the SNR approaches at 25 dB.

The Figure 6.8 (b) illustrates the LMS CE estimator performance for different normalized Doppler frequency values in Rician (LOS) fading channel. It has been observed that with increasing the relative speed $v \in \{84, 120, 150\}\, meter\ (m)/second\ (s)$ and corresponding normalized Doppler frequency $f_d T_s \in \{0.0133, 0.0188, 0.0235\}$, the FER has significantly lower values while SNR approaches at 25 dB.



(a) FER vs. SNR (dB) curve for 64-QAM (2/3), $f_d T_s$=0.0133



(b) FER vs. SNR (dB) curve for 64-QAM (2/3), $f_d T_s \in \{ 0.0133, 0.0188, 0.0235\}$

Fig. 6.8 LMS CE estimator performance comparisons for 64-QAM (2/3), 24 Mbps data rate, Rayleigh and Rician channel model; **(a)** FER vs. SNR for $f_d T_s$=0.0133; **(b)** FER vs. SNR with different normalized Doppler shift $f_d T_s =$ 0.0133, 0.0188 and 0.0235 respectively and corresponding relative speed $v \in \{84, 120,150\}\ m/s$.

CHAPTER 7

CONCLUSION

With the advent of all major car makers pursuing an effective self-driving system, it is imperative that a vehicular communication system, which is functional even in the harshest of environments. Channel estimation and channel equalization are one of the most important system blocks in a communication system to deal with the high mobility environment of the vehicular communication system. The various innovations proposed in this thesis provide adequate solutions to handle the dynamic environments. The novel ideas that presented in this thesis has been published as research paper in KSII Transactions on Internet and Information Systems is online scholarly journal indexed in SCIE (Thomson Reuters) and SCOPUS (Elsevier) and published by Korean Society of Internet Information (KSII) and supported by Korean Electronics Technology Institute (KETI) at Feb, 2018.

7.1 Contributions

This research work is started with the design of physical layer implementation of IEEE 802.11 standard transceiver prototype in cognitive frame work model over 5 GHz wireless channel considering the fading channel characteristics of the transmission medium. Using GNU Radio researcher benefited by an active community and a large open source radio echo-system that provides the graphical editor to configure the transceiver model. It also provides the graphical outputs of the signal that help to visualize the live signal.

In this thesis, LS and training symbol based adaptive iterative LMS CE techniques have been applied for IEEE 802.11 receiver in GNU Radio and USRP N200 platform to minimize the FER of designed transceiver model while data symbol were passed through the wireless fading channel models developed in GRC considering the high mobility environments. The proposed scheme not only improves the performance of existing multi-carrier systems but also provides greater flexibility in the design of telecommunication systems for the future. Although the proposed technique is implemented for IEEE 802.11 a/g/p standard, it is suitable for any OFDM based multi-carrier systems that has to cope with high mobility or work reliably at higher frequencies.

More precise approximation of SNR is done by modifying the SNR calculation process of YANS and NIST error rate model as mentioned in the equation 5.5 of chapter 5 to get accurate BER and FER of IEEE 802.11 a/g/p standard transceiver. The wireless fading channels can be expressed as the FIR filter with delay taps and complex tap coefficients that actually gives impulse response by adding certain distortion to the amplitude and phase of the transmitted symbol. Here it has been observed that for fading channels LMS CE techniques give better performance as it adjusts with the channel condition through the training based adaptive minimization process of error cost function. According to the FER vs. SNR curve analysis, for the both of fading channel models known as Rayleigh and Rician, assigned Power Delay Profile (PDP), normalized Doppler shift 0.0133, chosen modulation scheme 16 QAM and data rate 24 Mbps, the receiver has given suggestively less FER while SNR approaches at 25dB as discussed in section 6.5 of chapter 6. Therefore, according to the analysis, it has been stated that applied channel estimation and equalization technique enhances the receiver sensitivity and adaptability of the channel condition.

7.2 Future Scope of Work

Due to the decentralized and dynamic nature of vehicular communication it is very problematic to design the transceiver model of IEEE 802.11 standard. The main aim for this thesis is to give the cognitive frame work to enable adequate channel estimation and equalization in all mobility conditions. This goal has been achieved and proved through simulations as well as the test-bed of the designed model in GRC is applied practically using one peer of USRP N200 device in the laboratory environment.

This cognitive frame work of IEEE 802.11 standard can be extended and designed in Multiple-Input Multiple-Output (MIMO) state platform with MIMO expansion of USRP N200 to achieve higher data rate by using parallel channels in spatial domain over the same time and frequency without consuming extra bandwidth. Many other channel estimation techniques can be applied to this model to enhance the performance especially the powerful Kalman filter based iterative channel estimation can deploy in GNU Radio environment to cope with fast time varying channel. MIMO in OFDM wireless standard can be achieved by using spatial multiplexing technique and beam forming is required to get the fast time varying channel model behavior.

REFERENCES

[1] Muhammad Morshed Alam, Mohammad Rakibul Islam, Muhammad Yeasir Arafat and Feroz Ahmed, "FER Performance Evaluation and Enhancement of IEEE 802.11 a/g/p WLAN over Multipath Fading Channels in GNU Radio and USRP N200 Environment," *KSII Transactions on Internet and Information Systems*, vol. 12, no. 1, pp. 178-203, 2018. DOI: 10.3837/tiis.2018.01.009.

[2] Bastian Bloessl, Michele Segata, Christoph Sommer and Falko Dressler, "Towards an Open Source IEEE 802.11p Stack: A Full SDR-based Transceiver in GNU Radio", *Proceedings of 5th IEEE Vehicular Networking Conference (VNC 2013),* Boston, MA, Dec. 2013, pp. 143-149.

[3] Bastian Bloessl, Michele Segata, Christoph Sommer and Falko Dressler, "Decoding IEEE 802.11a/g/p OFDM in Software using GNU Radio", *Proceedings of 19th ACM International Conference on Mobile Computing and Networking (MobiCom)*, Miami, FL, Oct. 2013, pp. 159-161.

[4] Yan Yang, Dan Fei and Shuping Dang, "Inter-Vehicle Cooperation Channel Estimation for IEEE 802.11p V2I Communications", *Journal of Communications and Networks*, Vol.19, No.3, June 2017.

[5] Lihua Yang, Longxiang Yang and Yan Liang, "Iterative Channel Estimation for MIMO-OFDM system in Fast Time Varying Channels", *KSII Transactions of Internet and Information Systems*, Vol. 10, No. 9, Sept. 2016.

[6] Razvan Andrei Stoica, Stefano Severi and Giuseppe Thadeu Freitas de Abreu, "On Prototyping IEEE 802.11p Channel Estimators in Real World Environments Using GNU Radio*", IEEE Intelligent Vehicles Symposium (IV)*, Gothenburg, Sweden, June, 2016.

[7] Aida Zaier and Ridha Bouallegue, "A Full Performance Analysis of Channel Estimation Methods for Time Varying OFDM Systems", *International Journal of Mobile Network Communication & Telematics*, Vol.1, No.2, Dec. 2011.

[8] Jusnaini Muslimin, A.L. Asnawi, A.F. Ismail, H.A. Mohd Ramli, A.Z. Jusoh, S. oorjannah and Nor F. M. Azmin, "Basic Study of OFDM with Multipath Propagation Model in GNU platform", *IEEE Conference on Wireless Sensors*, 2015.

[9] ElSayed A. Ahmed and Mohamed M. Khairy, "Semi-adaptive Channel Estimation Technique for LTE Systems", *IEEE Symposium on Computers and Communications (ISCC)*, 2011.

[10] Meng-Han Hsieh and Che-Ho Wei, "Channel Estimation for OFDM Systems Based on Comp-type Pilot Arrangement in Frequency Selective Fading Channels", *IEEE Transactions on Consumer Electronics*, Vol.44, No.1, February 1998.

[11] P. Fuxjager, A. Cotantini, D. Valerio, P. Castiglione, G. Zacheo, T. Zemen and F. Ricciato, "IEEE 802.11p Transmission Using GNU Radio", *6th Karlsruhe Workshop on Software Radios*, March 2010, Karlsruhe, Germany from http://thomaszemen.org/papers/Fuxjaeger10-WSR-paper.pdf

[12] Ha Cheol Lee, "FER Performance in the IEEE 802.11 a/g/n Wireless LAN over Fading Channel", *Scientific Research on Communications and Network*, 2013, 5, 10-15.

[13] M.Eng. Christopher, B.Eng. Arthur Witt and Prof. Dr. Roland Muenzner, "A new ns-3 WLAN error rate model- Definition, validation of the ns-3 implementation and comparison to physical layer measurements with AWGN channel", *Workshop on ns-3 (WNS3)*, Barcelona, Spain, 2015 from https://www.nsnam.org/wp-content/uploads/2015/04/WNS3_2015_submission_34.pdf

[14] Md. Masud Rana, Jisang Kim and Won-Kyung Cho, "An Adaptive LMS Channel Estimation Method for LTE SC-FDMA Systems", *International Journal of Engineering & Technology*, Vol. 10, No. 5, December 2010.

[15] Olivier Goubet, Gwilherm Baudic, Frederic Gabry and Tobias J. Oechtering, "Low Complexity Scalable Iterative Algorithms for IEEE 802.11p Receivers", *IEEE Transactions on Vehicular Technology*, Vol. 64, 2014.

[16] Md. Masud Rana and Md. Kamal Hosain, "Adaptive Channel Estimation Techniques for MIMO OFDM Systems, *International Journal of Advanced Computer Science and Applications*", Vol.1, No.6, December 2010.

[17] Zijun Zhao, Xiang Cheng and Miaowen, Bingli Jiao, Cheng-Xiang Wang, "Channel Estimation Schemes for IEEE 802.11p Standard*", IEEE Intelligent Transportation Systems Magazine*, 2013.

[18] Guangyu Pei and Thomas R. Henderson, "*Validation of OFDM NIST and YANS error rate model in ns-3, Boeing Research & Technology*", Technical Report from https://www.nsnam.org/~pei/80211ofdm.pdf

[19] Bahattin Karakaya, Huseyin Arslan, Hakan and Ali Cirpan, "Channel Estimation for LTE Uplink in High Doppler Spread", *IEEE* Wireless Communications and Networking Conference, 2008.

[20] GNU Radio Manual and C++ API Reference (3.7.10.1) [Online]. Available: https://www.gnuradio.org/doc/doxygen/index.html

[21] Sinem Coleri, Mustafa Ergen, Anuj Puri and Ahmad Bahai, "A study of channel estimation in OFDM systems", *IEEE 56th* Vehicular Technology Conference, 2002.

[22] Bastian Bloessl, IEEE 802.11 a/g/p Transceiver [Online]. Available: https://github.com/bastibl/gr-ieee802-11.git

[23] Samaneh Shooshtary, "Development of a MATLAB Simulation Environment for Vehicle-to-Vehicle and Infrastructure Communication Based on IEEE 802.11p", *M.S. thesis, Department of Technology and Build Environment, University of Gavle*, Dec., 2008 from http://www.divaportal.org/smash/get/diva2:133311/FULLTEXT01.pdf

[24] Rao, K. Deergha, "Channel Coding Technique for Wireless Communication", *Performance of digital communication over fading channels*, Chapter 2, Springer, 2015 from http://www.springer.com/gp/book/9788132222910

[25] Theodore S.Rappaport, "*Wireless Communication Principles and Practice*", 2nd Edition, Pearson Education, 2002, ch. 5-7, pp. 177-415.

[26] GNU Radio, the free and open source radio ecosystem, [Online]. Available: http://gnuradio.org/redmine

[27] GNU Radio Out of Tree Module, [Online]. Available: https://wiki.gnuradio.org/index.php/OutOfTreeModules

APPENDIX A

PROGRAMMING INTRICACIES

A.1 Build GNU Radio and UHD packages from source in Linux

The build-gnuradio is an install script for recent Fedora and Ubuntu systems provided by Marcus Leech. This script has a number of options that we can use to install different versions of GNU Radio in Linux platform especially in Debian or Ubuntu Linux Desktop. Just running build-gnuradio with no additional options, it builds the latest released version from the 3.7 series and also build UHD packages that gives the C/C++/Python API to connect with the hardware USRP [26].

```
# sudo apt-get update

# sudo apt-get upgrade

# mkdir gnuradio

# cd gnuradio

# wget http://www.sbrac.org/files/build-gnuradio && chmod a+x build-
gnuradio && ./build-gnuradio

# gnuradio-companion (launching the GNU Radio)

# ifconfig eth0 192.168.10.1 (setting the IPv4 address in LAN)

# ldconfig

# ifconfig

# ping 192.168.10.2 (Default IPv4 address at USRP N200)

# uhd_find_devices (to detect the USRP hardware using the UHD module)
```

A.2 Build IEEE 802.11 a/g/p Transceiver Module in GNU Radio

Swig is required to create the python bindings.

```
# sudo apt-get install swig

# sudo port install swig
```

For getting new logging feature of GNU Radio, it requires log4cpp module. It can be installed using below commands [22]:

```
# sudo apt-get install liblog4cpp5-dev
# sudo port install log4cpp
```

In order to connect the wire-shark to the block interface it will use to trace and analyze the packet flow or even the frame structure of IEEE 802.11, we require a new module named as gr-foo. It can be installed using below commands:

```
# git clone https://github.com/bastibl/gr-foo.git
# cd gr-foo
# mkdir build
# cd build
# cmake ..
# make
# sudo make install
# sudo ldconfig
```

The commands that are essential to install and build IEEE 802.11 a/g/p blocks are given below:

```
# git clone git://github.com/bastibl/gr-ieee802-11.git
# cd gr-ieee802-11
# mkdir build
# cd build
# cmake ..
# make
# sudo make install
# sudo ldconfig
```

As the transmitted is using the Tagged Stream Blocks it has to store a complete frame in the buffer before processing it. The default maximum shared memory might not be enough on most Linux system. The allocated memory can be increased using the command given below:

```
# sudo sysctl -w kernel.shmmax=2147483648
```

To capture the data packets over the wireless LAN card, the command given below is used:

```
# tcpdump –w ieee802.11_frame.pcap –i eth0
```

A.3 Creating a Custom Block in GNU Radio Companion (GRC)

To develop a module, there are lot of complex tasks are required to do such as boilerplate code, makefile editing, etc. The default script "gr_modtool" may help to all these tasks by automatically editing makefiles, using templates, and doing as much work as possible for the developer such that you can jump straight into the DSP coding. The script "gr_modtool" is now available in the GNU Radio source tree and it is installed by default [27].

```
# gr_modtool newmod gr-IEEE-802.11-LMS

# cd gr-IEEE-802.11-LMS

# ls

apps   cmake   CMakeLists.txt   docs   examples   grc   include   lib
python  swig
```

To create empty files for the block and edit the CMakeLists.txt files, need to run below command:

```
#  gr-IEEE-802.11-LMS  %  gr_modtool  add  -t  general  -l  cpp
[C/C++/Python Program Name]
```

It consists of several sub-directories. The programs that will be written in C++ (or C, or any language that is not Python) is put into the directory "lib/". For C++ files, we usually have headers which are put into "include/" or even in "lib/".

The python scripts are placed into the directory "python/". The GNU Radio blocks are available in Python even if they were written in C++. This is done by the help of SWIG, the simplified wrapper and interface generator, which automatically creates glue code to make this possible. SWIG needs some instructions on how to do this, which are put into the "swig/" subdirectory.

If someone want the blocks to be available in the GRC, the graphical UI for GNU Radio, then needed to add XML descriptions of the blocks and put them into "grc/".

For documentation, "docs/" contains some instructions on how to extract documentation from the C++ files and Python files (we use Doxygen and Sphinx for this) and also make sure they're available as docstrings in Python. Even someone can add custom documentation here as well.

The “apps/” subdir contains any complete applications (both for GRC and standalone executables) which are installed to the system alongside with the blocks.

The directory, “examples/” can be used to save all the examples, which are a great addendum to documentation because other developers can simply look straight at the code to see how your blocks are used.

The build system brings some baggage along, as well: the “CMakeLists.txt” file (one of which is present in every subdirectory) and the “cmake/” folder. You can ignore the latter for now, as it brings along mainly instructions for CMake on how to find GNU Radio libraries etc. The CMakeLists.txt files need to be edited a lot in order to make sure the custom module builds correctly.

To build the custom blocks for GRC following commands should have to use [27]:

```
# mkdir build

# cd build/

# cmake ../

# make
```

A.4 The LS Estimator in C++ Code

The theorical algorithm of Least Square based channel estimator has been briefly discussed in chapter 6 under section 6.1 to 6.2. The entire algorithm is implemented in C++ code into the file “ls.cc” and build in GNU Radio to make it available in GRC block to use it in the receiving section of IEEE 802.11 fame processing. The details C++ code is given below:

```
#include "ls.h"

#include <cstring>

#include <iostream>

using namespace gr::ieee802_11::equalizer;

void ls::equalize(gr_complex *in, int n, gr_complex *symbols, uint8_t *bits,
boost::shared_ptr<gr::digital::constellation> mod) {
    if(n == 0) {
        std::memcpy(d_H, in, 64 * sizeof(gr_complex));
    } else if(n == 1) {
        double signal = 0;
        double noise = 0;
        for(int i = 0; i < 64; i++) {
            if((i == 32) || (i < 6) || ( i > 58)) {
                continue;
            }
            noise += std::pow(std::abs(d_H[i] - in[i]), 2);
            signal += std::pow(std::abs(d_H[i] + in[i]), 2);
            d_H[i] += in[i];
            d_H[i] /= LONG[i] * gr_complex(2, 0);
        }
        d_snr = 10 * std::log10(signal / noise / 2);
    } else {
        int c = 0;
        for(int i = 0; i < 64; i++) {
            if( (i == 11) || (i == 25) || (i == 32) || (i == 39) || (i == 53) || (i < 6) || ( i >
58)) {
                continue;
            } else {
                symbols[c] = in[i] / d_H[i];
                bits[c] = mod->decision_maker(&symbols[c]);
```

```
                                c++;
                        }
                }
        }
}
double ls::get_snr() {
        return d_snr;
}
```

A.5 The Adaptive LMS Filter Implementation in C++ Code

The theorical algorithm of adaptive LMS based channel estimator has been briefly discussed in chapter 6 under section 6.3 to 6.6. The entire algorithm is implemented in C++ code into the file “lms.cc” and build in GNU Radio to make it available in GRC block to use it in the receiving section of IEEE 802.11 fame processing. The details C++ code is given below:

```
#include "lms.h"
#include <cstring>
#include <iostream>
using namespace gr::ieee802_11::equalizer;
void lms::equalize(gr_complex *in, int n, gr_complex *symbols, uint8_t *bits,
boost::shared_ptr<gr::digital::constellation> mod) {
        if(n == 0) {
                std::memcpy(d_H, in, 64 * sizeof(gr_complex));
        } else if(n == 1) {
                double signal = 0;
                double noise = 0;
                for(int i = 0; i < 64; i++) {
                    if((i == 32) || (i < 6) || ( i > 58)) {
                        continue;
                    }
                    noise += std::pow(std::abs(d_H[i] - in[i]), 2);
                    signal += std::pow(std::abs(d_H[i] + in[i]), 2);
                    d_H[i] += in[i];
                    d_H[i] /= LONG[i] * gr_complex(2, 0);
                }
                d_snr = 10 * std::log10(signal / noise / 2);
        } else {
                int c = 0;
                for(int i = 0; i < 64; i++) {
                    if( (i == 11) || (i == 25) || (i == 32) || (i == 39) || (i == 53) || (i < 6) || ( i >
58)) {
                        continue;
                    } else {
                        symbols[c] = in[i] / d_H[i];
                        bits[c] = mod->decision_maker(&symbols[c]);
                        gr_complex point;
                        mod->map_to_points(bits[c], &point);
                        d_H[i] = gr_complex(1-alpha,0) * d_H[i] + gr_complex(alpha,0) *
in[i] / point;
                        c++;
```

```
                        }
                }
        }
}
double
lms::get_snr() {
        return d_snr;
}
```

APPENDIX B

B.1 Publication List

[1] Muhammad Morshed Alam, Mohammad Rakibul Islam, Muhammad Yeasir Arafat and Feroz Ahmed, "FER Performance Evaluation and Enhancement of IEEE 802.11 a/g/p WLAN over Multipath Fading Channels in GNU Radio and USRP N200 Environment," *KSII Transactions on Internet and Information Systems*, vol. 12, no. 1, pp. 178-203, Feb-2018. DOI: 10.3837/tiis.2018.01.009.