

Error Rate Performance of OFDM Transceiver on Software-defined Radio

Sayali Karande¹, P. N. Kota²

Research Scholar, Department of Electronics and Telecommunications, Modern Education Society's College of Engineering, Pune, India¹

Associate Professor, Department of Electronics and Telecommunications, Modern Education Society's College of Engineering, Pune, India²

Abstract: This work is focused on an efficient model of OFDM transceiver with the help of software-defined radio (SDR) system with reconfigurable architecture for wireless communications. SDR possesses revolutionary developments such as algorithm implementation of digital communication and increased computing capabilities. In the paper, OFDM system design and implementation with novel technology development is the primary focus to be studied. Different modulation techniques of OFDM system and other related concepts like PAPR also taken into consideration in order to design the OFDM System. FEC coding is performed by suggesting Convolutional coders. In this paper, bit error rate and symbol error rates performance are analyzed by varying bits per symbol, window size and modulation scheme. Results proved that BER values are improved as modulation scheme size (M) is increased. Moreover, IEEE 802.11a standards are implemented. BPSK modulation with Radix-2 FFT, OFDM transceiver design is implemented using commercial Amitec SDR04 and configured by GNU radio companion in the laboratory environment.

Keywords: software-defined radio (SDR); OFDM; BPSK; GNU Radio Companion, BER.

I. INTRODUCTION

A Software-Defined Radio (SDR) system is a radio where one or more of the physical layer functions are software defined. In this system the components that are typically implemented in hardware is replaced by software. In general, SDR employs SDR04 as hardware setup and GNU radio companion as software. In conventional radio, the system modification is done only by replacing the hardware parts. This helps the new comers, to define and develop their own modules for execution under GNU Radio Companion (GRC). The GRC platform is a rich Graphical User Interface (GUI) which helps in developing SDR modules in a better and effective way, thus enabling less complex ways of development for programmers. The GUI model of program development was devised based on the idea of giving practical exposure to communication theory like signal generation, signal manipulation, multi-rate concepts, multiplexing schemes and analog and digital modulation schemes.

Furthermore, SDR has ability to analyze and visualize the signals that operates and plots the same which can never be done on hardware. The ability of SDR to accept and configure a considerable lot of daughter boards enable it to function on different frequency bands simultaneously with a wide range of operating frequencies. In order to implement SDR, a free and open source software development tool kit known as GNU Radio Companion is available. This toolkit provides a simple drag and drop based signal processing blocks with which many communication experiments can be done. SDR04 consists of a library developed in C++ for signal processing and a python that act as a glue to tie these blocks together for building and developing SDR. Using SDR04, for real-

time implementation of communication experiments, much hardware is available. One among that hardware is Amitec SDR04 [4], a hardware module that provides transmission and reception capabilities over a range of frequencies with two set of front-end antenna are available.

Orthogonal Frequency Division Multiplexing (OFDM) is extensively used in wireless communication systems due to its resistance to Inter Symbol Interference (ISI) caused by dispersive channel. OFDM has been considered strong candidate for the next generation high data rate wireless communication systems because of its good Bit Error Rate (BER) performance and high spectrum efficiency [1]. In a multicarrier technology such as OFDM, it is crucial to maintain orthogonality among all the subcarriers to prevent the Inter Carrier Interference (ICI) which if not done leads to significant performance degradation. In a high mobility environment, such as aerial vehicle communication, muticarrier transmission leads to severe ICI due to Doppler shift. In [2] this paper, the prototype of a GNU Radio based OFDM receiver is designed. Receiver comprises all layers up to parsing the MAC header and extracting the payload of IEEE 802.11 a/g/p networks.

The paper is organized as follows in section 2 we briefly review Software Defined Radio, GNU radio, and GNU radio companion. In Section 3 System Model Overview i.e. OFDM system, FEC encoding technique of the OFDM signal is explained. In Section 4 Transceiver Structure is explained. The performance with simulation result and analysis is discussed in section 5. and finally section 6 explained the conclusions drawn.

II. SOFTWARE DEFINED RADIO

Joe Mitola named a class of radios that is a single hardware that can perform different functions at different times as Software Defined Radio. This radio is said to be able to handle a wide spectrum, traffic, air interfaces and applications. Thus SDR04 is a radio platform that performs signal processing on the digitized version of a complex input (input that is varied in traffic, frequency, application and experiences different air interfaces). SDR04 comprises of RF section, IF section and baseband processing section. RF and IF sections are incorporated in SDR04 and baseband processing is performed in a PC using GNU radio companion. [10]

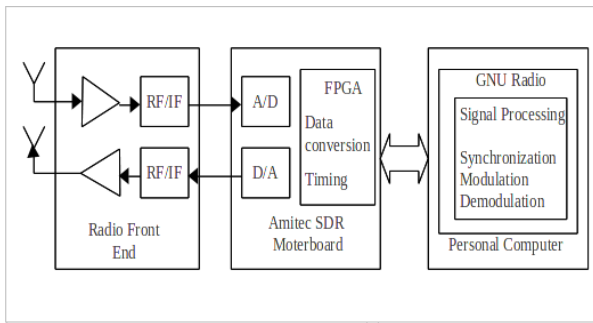


Fig. 1 Block diagram of Software-defined radio with GNU Radio

A. GNU Radio Platform

What is GNU Radio?

GNU Radio project was started by Eric Blossom. It is free software that can be used in Linux based OS (Ubuntu 14.04) for simulating communication systems and for designing Software Defined Radios. It is hardware independent environment.

GNU Radio is a free and open-source software development toolkit that caters signal processing blocks to implement Software Defined Radios (SDR). Different functionalities such as modulation, demodulation, filtering, encoding, decoding, source coding, channel coding etc. are provided as software codes [3]. In GNU Radio, all signal processing is acted through flow graphs, which consists of blocks. A block does transforming, decoding, filtering, adding signals, hardware access or many others. Data passes between blocks in various formats, complex or real integers, floats or basically any kind of data type user can define. The advantage of implementing functionalities as software modules helps provides reconfigurability property to SDR. Traditionally, for example, if a modulation scheme of a radio has to be changed, the entire hardware circuitry employed for modulation has to be changed.

B. GNU Radio Companion

GNU Radio provides a graphical user interface with GNU Radio companion (GRC). Experiments are done by connecting signal processing blocks written in C++. The flow graph is written and performed using python. In GNU Radio, signal processing blocks are written in C++ and they are connected by using Python. SWIG (Simplified Wrapper and Interface Generator) is used as an interface compiler between C++ and Python language.

Required GNU Radio modules are imported with the from gnuradio import command. A flow graph in a class is derived from gr.top_block.

GNU Radio Companion(GRC) allows user to implement GNU radio signal processing blocks in a manner similar to Simulink and Labview. Blocks are manually integrated. Every flow graph needs at least one sink and source. The OFDM modulator block is configured with various parameters such as FFT length, occupied tones, cyclic prefix length, modulation scheme selection (which is used to generate OFDM symbol), etc. [5]

III. SYSTEM MODEL OVERVIEW

A. OFDM (Orthogonal Frequency Division Multiplexing)

The modern digital communication system is developed with Orthogonal Frequency Division Multiplexing (OFDM). OFDM has increasing with high popularity, as it provides high speed data transmission. Here the OFDM communication system is designed in GNU Radio platform which is cost effective when compared to conventional system. The major advantage of this system is that, it is mostly focused on the software part in case of any functionalities changes. So, it is easy to understand and utilize in an effective manner can easily check your margins to see if your print area fits within the space allowed. OFDM has been proven to be a viable technique to overcome multipath fading in wireless channels. It has been adopted in many wireless standards namely, digital audio/video broadcasting, HIPERLAN/2, IEEE 802.11a and g standards for Local Area Networks (WLANs) and is going to be used in various future broadband wireless communication systems. [9]

B. Channel Coding Technique:

In digital communications, a channel code is the term relating to the forward error correction code and interleaving in communication and storage where the communication media is recognized as a channel. The channel code is utilized to defend data sent over it for storage or recovery even in the orientation of noise (errors). Channel coding is referred to process in both transmitter and receiver of a digital communications framework. Channel coding is made out of three techniques, for example Randomization, FEC (Forward Error Correction) and Interleaving.

IV. TRANSCEIVER STRUCTURE

SDR comprises of RF section, IF section and baseband processing section. RF and IF sections are incorporated in SDR04 and baseband processing is performed in a PC using GNU radio companion (shown in Fig. 1).

The SDR04 allows for high-bandwidth, high-dynamic range processing capability. This includes a Alterra Cyclone FPGA with single cycle access memory, 18X18 multipliers for dedicated DSP & programmable general logic elements, two 40 MS/s ADCs, two 40 MS/s DACs and Gigabyte Ethernet connectivity to flow information to and from host processors. All baseband signal processing (e.g. modulation, amplification, mixing, filtering etc.) is done in GNU Radio. SDR04 can be reconfigured (in runtime also) to desired specifications in host computer by

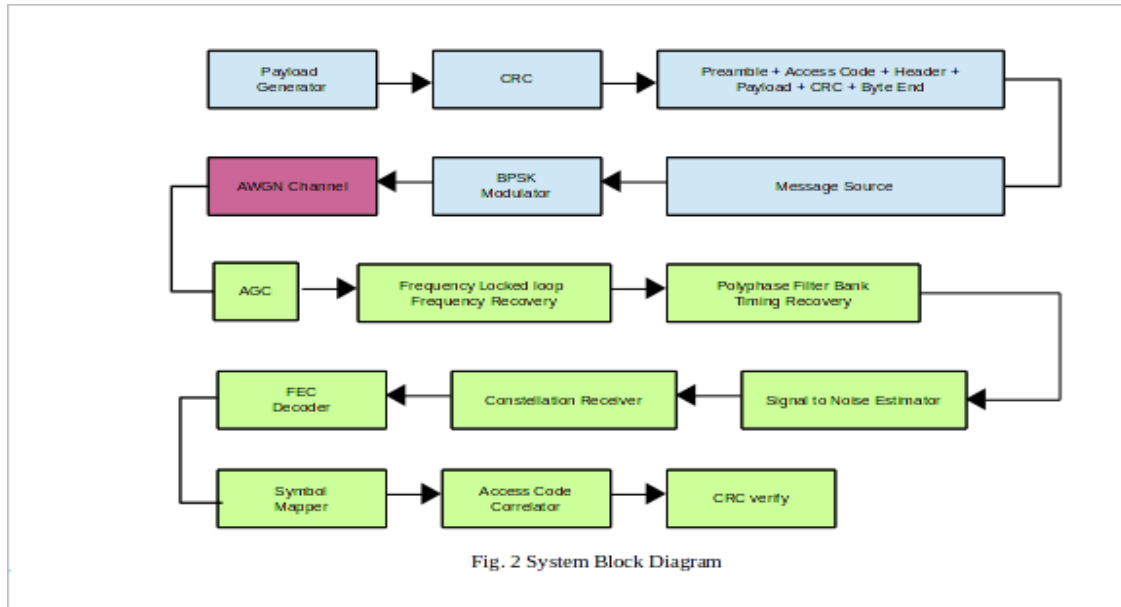


Fig. 2 System Block Diagram

using GNU Radio. GNU Radio Companion is a free software development toolkit that offers the signal processing runtime and readily available more than 100 processing blocks to implement software radios employing low-cost external RF hardware (SDR04) and allows real time SDR applications. In GNU Radio, signal processing blocks are written in Python and those are connected using C++ and both languages are communicated by SWIG (Simplified Wrapper and Interface Generator) interface compiler. Thus, the developer is allowed to accomplish real-time, high-throughput radio systems in a simple to-use, rapid-application development environment. In this paper all GNU Schematics (Signal flow graphs) are drawn for OFDM IEEE 802.11a specifications (FFT size=64) [6].

This project implemented the example frame encoder provided by the IEEE 802.11a standard document. The transmitting path is shown in Fig. 2. This implementation includes a group of standard compliant processing blocks other than the basic ones already existing in GNU Radio. Coding has the usefulness that it allows us to increase the rate at which information may be transmitted over a channel while maintaining a fixed error rate.

Alternatively, coding allows us to reduce the information bit error rate while maintaining a fixed transmission rate [7]. More generally, coding allows to design a communication system in which both information bit rate and error rate are independently and arbitrarily specified but subject to a constraint on bandwidth [8].

Channel coding is used to preclude and to adjust the transmission errors of wireless systems, must have a very good performance in order to preserve high data rates. The 802.11a channel coding chain is framed of three steps: Randomizer, Forward Error Correction (FEC) and Interleaving.

A. Randomizer

Data randomization is accomplished on each uplink and downlink burst of data. If the quantity of data to transmit does not equip exactly the quantity of data allocated,

TABLE 1 IEEE 802.11a OFDM STANDARDS

Parameters	Value
Total subcarriers N_{ST}	52
Data subcarriers N_{SD}	48
Pilot subcarriers N_{SP}	4
Subcarrier Frequency Spacing	312.5 KHz (20 MHz/64)
Symbol Interval Time T_{SYM}	$4\mu s (T_{GI} + T_{FFT})$
Data Interval Time T_{DATA}	$3.2\mu s (1/F_{SP})$
Guard Interval (GI) Time T_{GI}	$0.8\mu s (T_{FFT}/4)$
IFFT/FFT Period T_{FFT}	$3.2\mu s (1/F_{SP})$
SIGNAL Symbol Time T_{SIGNAL}	$4\mu s (T_{GI} + T_{FFT})$
Preamble $T_{PREAMBLE}$	$16\mu s (T_{SHORT} + T_{LONG})$
Short Training Sequence T_{SHORT}	$8\mu s (10 \times T_{FFT}/4)$
Long Training Sequence T_{LONG}	$8\mu s (T_{GI2} + 2 \times T_{FFT})$
Training symbol GI T_{GI2}	$1.6\mu s (T_{FFT}/2)$

padding of 0xFF is added to the end of the transmission block. The Pseudo-Random Binary Sequence (PRBS) generator is employed for randomization.

B. FEC encodings

1. (Tail-biting) Convolutional Code (CCSDS)- This code is mandatory according to the 802.11a standard

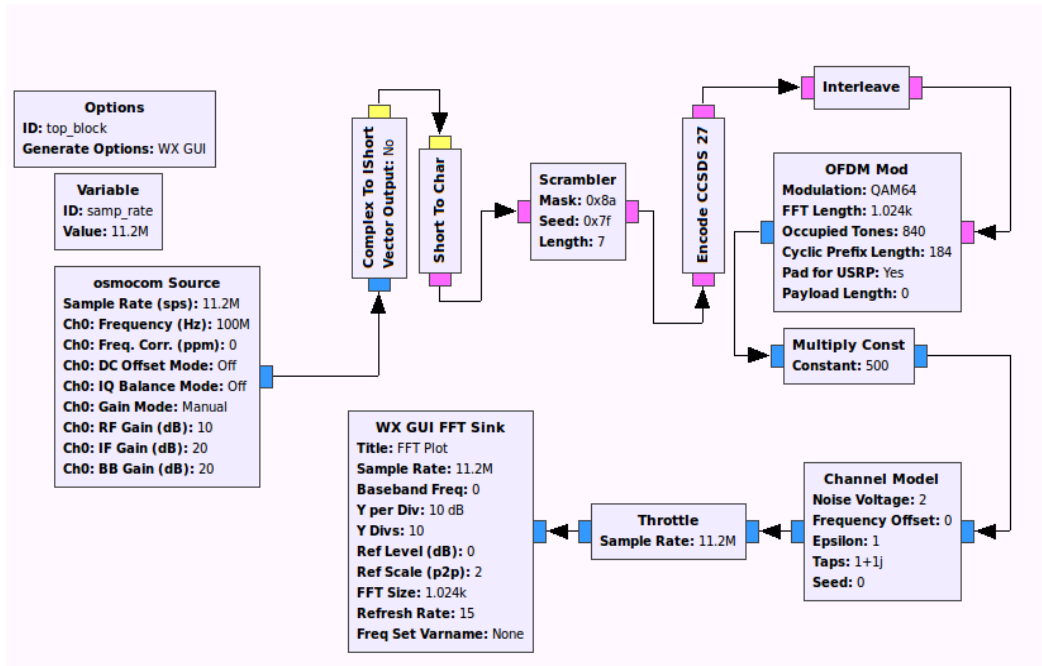


Fig. 3 Block Schematic of OFDM modulation with Channel coder

2. The Zero-Tailing Convolutional Code (ZT CC) is mandatory for 802.11a OFDM.
3. Cyclic Redundancy Coding (CRC) - According to the 802.11a standards.

C. Interleaving

Interleaving is used to defend the transmission against long successions of sequential errors, which are very difficult to adjust. The interleaving is followed in two steps:

1. Separate the coded bits over subcarriers. A first permutation guarantees that adjacent coded bits are mapped on to non-adjacent subcarriers.
2. The second permutation guarantees that adjacent coded bits are mapped alternately on to more or less significant bits of the constellation, thus avoiding long runs of bits of low reliability.

D. Repetition

Repetition is used to raise the signal margin boost over the modulation and FEC coding mechanisms.

V EXPERIMENTAL RESULTS

A. BER Performance of OFDM System:

In error code block, the performances of OFDM system with BPSK and QPSK modulations. The OFDM system with BPSK performs well in the laboratory environment without obstruction. The BER performance achieves below 10^{-3} at the SNR of 19.5 dB. The BER performance of the OFDM system can be obtained via our hardware within the SNR range 8.5 - 21 dB, and the BER performance reaches the floor at SNR of 21 dB. The error floor issue arises due to the power saturation problem at the receiver. The performance of the OFDM system with a QPSK constellation is worse than that with a BPSK constellation.

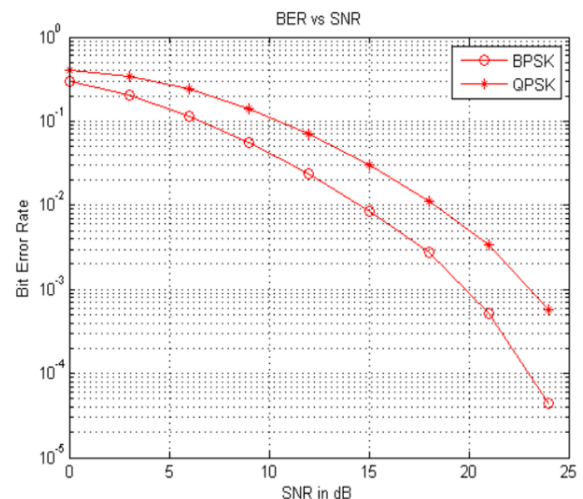


Fig. 4 Comparison of BER vs SNR between BPSK and QPSK

The BER performance can only be obtained for SNR higher than 13 dB. The error floor starts to be seen at SNR of 22 dB.

Fig. 3 also shows that the performance of OFDM systems with BPSK in the same environment is better than that with QPSK by approximate 3 - 4 dB. As per the Fig. 3 the BPSK performance of OFDM systems is better than that with QPSK modulation and it is by 3.5 dB.

B. Channel Coding of OFDM Modulation:

In channel coding block, bits per symbol value are given according to the modulation scheme. For example, on considering the modulation scheme such as BPSK, QPSK. For different values of SNR the corresponding BER values are achieved with different modulation scheme in real time.

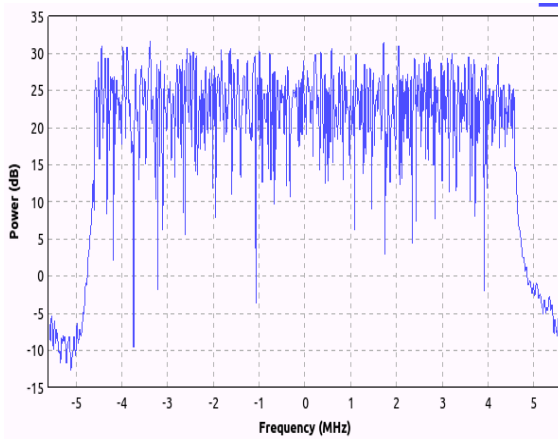


Fig.5 OFDM modulation without Channel coder

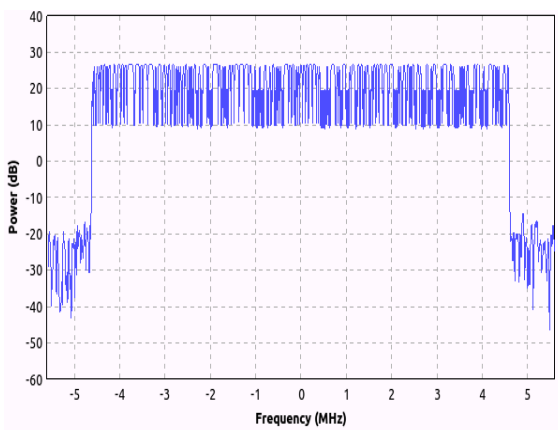


Fig.6 OFDM modulation with Channel coder

The OFDM modulator converts the given data to packets. The Multiply Const block behaves as an amplifier, which can be controlled by the multi_const variable. The Channel Model block simulates a wireless channel. It can be configured by various parameters such as noise voltage, frequency offset, etc. The Throttle block is added to reduce the burden on CPU. The FFT sink is a graphical sink that plots the output signal in frequency domain.

VI. CONCLUSION

This paper shows the development of software defined radio (SDR) utilizing SDR04 testbed and GNU Radio companion toolkit. IEEE802.11a OFDM is executed with channel coding by considering FEC encoder. IEEE 802.11a OFDM transceiver determinations at different (i) modulation schemes (ii) Channel noise levels and (iii) phase variation. This result into the comparative study, it was found that: when channel conditions are poor, energy efficient schemes such as BPSK was used and as the channel quality improves, QPSK was used. It adjusts the modulation method almost directly for optimum data transfer, thus making a most efficient use of the bandwidth and increasing the overall system capacity. Thus

REFERENCES

[1] Abirami M, Gandhiraj R, Soman K P, "Performance Analysis of Real Time OFDM Based Communication System Using GNU Radio and USRP," International Journal of Advanced Research in

Computer Science and Software Engineering, Vol. 3, Issue 6, June 2013.
 [2] Bastian Bloessl, Michele Segata, Christoph Sommer, Falko Dressler, "An IEEE 802.11a/g/p OFDM Receiver for GNU Radio," ACM 978-1-4503-2181-5/13/08, 2013.
 [3] Shraavan Sriram, Gunturi Srivatsa , "Plug-ins for GNU Radio Companion," International Journal of Computer Applications (0975 – 8887) Volume 52– No.16, August 2012.
 [4] SDR, "Software Defined Radio Lab SDR04," available from World Wide Web: <http://amitec.co/home/mobile-wireless-communication/0-4-4-ghz-software-defined-radio-lab-sdr04> .
 [5] GNU Radio Companion Build a Project using Out of Tree module, available on World Wide Web: www.gnuradio.org/redmine/projects/gnuradio/wiki/GNURadioCompanion .
 [6] Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification, IEEE Std. 802.11aa, 1999.
 [7] Hen Geul Yeh, Paul Ingerson, "Software defined radio for OFDM transceivers," IEEE, 2010.
 [8] H. D. Schulze and C. Lueders, Theory and applications of OFDM and CDMA: wideband wireless communications, John Wiley & Son, Chichester, 2005.
 [9] S. Haykin and M. Moher, Modern Wireless Communication, Prentice Hall, 2005.
 [10] Alexander M. Wyglinski, Digital Communication Systems Engineering with Software-Defined Radio, Mobile communications series, U. S. Library, 2013.