

TECHNICAL UNIVERSITY OF CRETE

GRADUATION THESIS

---

# Implementation of Smart Radios with OFDM in SDR platform USRP N200 & Application in a Cognitive Radio Protocol

---

*Author:*

Athanasios GKIOLIAS

*Supervisor:*

Prof. Athanasios P. LIAVAS

***Thesis committee***

Professor Athanasios P. Liavas (ECE)

Professor Michael Paterakis (ECE)

Associate Professor George N. Karystinos (ECE)

*A thesis submitted in fulfilment of the requirements  
for the degree of BSc of Engineering*

*in the*

Department Of  
Electronic and Computer Engineering

November 2015



# Declaration of Authorship

I, Athanasios GKIOLIAS, declare that this thesis titled, 'Implementation of Smart Radios with OFDM in SDR platform USRP N200 & Application in a Cognitive Radio Protocol' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a graduation degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.

Signed:

---

Date:

---

TECHNICAL UNIVERSITY OF CRETE

## *Abstract*

Electronic and Computer Engineering

BSc of Engineering

### **Implementation of Smart Radios with OFDM in SDR platform USRP N200 & Application in a Cognitive Radio Protocol**

by Athanasios GKIOLIAS

*Orthogonal Frequency Division Multiplexing* or *OFDM* is a modulation format that is being used for many of the latest wireless and telecommunications standards. OFDM has been adopted in the Wi-Fi arena at standards like 802.11a, 802.11n, 802.11ac and more. It has also been chosen for the cellular telecommunications standard LTE / LTE-A and, in addition, it has been adopted by other standards such as WiMAX and many more. The combination of high data capacity, high spectral efficiency, and its resilience to intersymbol interference as a result of multi-path effects means that it is ideal for the high data applications that have become a major factor in today's communications scene. Since OFDM has gained a significant presence in the wireless market place, it is very interesting to examine it thoroughly and acquire a great amount of knowledge in this area of telecommunications.

Firstly, the purpose of this thesis is to provide an experimental study of an OFDM implementation by utilizing the Ettus UHD drivers in C++, USRPs N200 and several math libraries. The initial prototyping takes place in Matlab and after thorough testing the code is converted into the corresponding C++ code that will be able to handle the incoming/outcoming OFDM packets' processing in realtime. Afterwards, the transmitter-receiver set will be pushed to its limits in order to examine the existence and estimation of a frequency selective channel instead of a flat fading channel.

Secondly, we have the creation and study of a time slotted protocol with a master transmitter(Primary User - PU) and a slave transceiver(Secondary User - SU) that acts as a cognitive radio. The SU after spectrum sensing transmits packets when the PU is idle, in order to keep all time-slots occupied. Next, this application is modified so that the users can transmit a binary image and a receiver processes the incoming parts. The processing of all the packets is done in realtime and provides us with the received image of each user in the end of his transmission.

## *Acknowledgements*

First, I would like to express my sincere gratitude to my supervisor, Professor Athanasios Liavas, for his guidance, support and advice during my graduation project. This thesis would not have been possible without his help. I owe a great deal of thanks to my family, for their constant support and encouragement throughout the duration of my study. I have to thank Nikos Karamolegkos separately, for our productive collaboration during the last semester. Finally, I would like to thank my friends and colleagues for their help and for the enjoyable time I have experienced during my study at the Technical University of Crete.

# Contents

<b>Declaration of Authorship</b>	<b>ii</b>
<b>Abstract</b>	<b>iii</b>
<b>Acknowledgements</b>	<b>iv</b>
<b>Contents</b>	<b>v</b>
<b>List of Figures</b>	<b>vii</b>
<b>Abbreviations</b>	<b>ix</b>
<b>1 Background Knowledge</b>	<b>1</b>
1.1 Orthogonal frequency-division multiplexing ( <i>OFDM</i> )	1
1.2 Software Defined Radio (SDR)	3
1.2.1 Universal Software Radio Peripheral(USRP)	4
1.2.2 USRP N200, a quick glimpse	4
1.2.3 Daughterboards of our implementation	5
1.3 Cognitive radio	7
1.4 Software needed in the implementation	7
1.4.1 Matlab	7
1.4.2 Usrc Hardware Drivers(UHD)	7
1.4.3 Armadillo library	8
1.4.4 Matlab Engine	8
1.4.5 Gnu Radio	8
1.5 Thesis outline	10
<b>2 Approach of an OFDM based system</b>	<b>11</b>
2.1 Structure and analysis of the OFDM scheme	11
2.2 An analog analysis of the subcarriers	14
2.3 From analog signal to digital samples	16
2.4 Synchronization in OFDM	16
2.4.1 Coarse time synchronization	17
2.4.2 Estimation and correction of Carrier Frequency Offset(CFO)	20
2.4.3 Fine time synchronization	21

2.5	Channel estimation . . . . .	26
2.6	Correction of channel's effect and symbols' estimation . . . . .	28
2.7	Implementation details . . . . .	29
2.7.1	Specifications . . . . .	29
2.7.2	Square Root Raised Cosine (SRRC) filter . . . . .	30
2.7.3	Channel types . . . . .	30
2.7.4	Initial detection of packets . . . . .	32
2.8	Chapter summary . . . . .	33
<b>3</b>	<b>A time slotted protocol with a primary transmitter and a secondary cognitive radio</b>	<b>37</b>
3.1	Introduction . . . . .	37
3.2	Protocol details . . . . .	38
3.3	Synchronization with the Ettus MIMO cable . . . . .	39
3.4	Image transmission from the users . . . . .	40
3.4.1	Image reception at the receiver . . . . .	42
3.4.2	Full exploitation of the time-slots with a cyclic buffer . . . . .	44
3.5	Error analysis in image transfer from various distances . . . . .	46
3.6	Chapter summary . . . . .	47
<b>4</b>	<b>Future work</b>	<b>49</b>
4.1	Future work . . . . .	49

# List of Figures

1.1	OFDM carriers . . . . .	2
1.2	USRP N200 . . . . .	5
1.3	USRP N200 interior . . . . .	6
1.4	USRP daughterboards used in the implementation . . . . .	6
1.5	GNU Radio assembly used to transmit Matlab-created data for initial design . . . . .	9
1.6	GNU Radio assembly used to receive data while plotting the corresponding spectrum . . . . .	9
2.1	An OFDM system . . . . .	12
2.2	$N$ parallel flat fading channels . . . . .	14
2.3	Our OFDM system structure . . . . .	17
2.4	Structure of the synchronization symbol or preamble $S_{Sch}$ . . . . .	18
2.5	Synchronization statistic of equation 2.8 ( $P_{Sch}$ ) . . . . .	19
2.6	Synchronization statistic of equation 2.9 ( $M_c$ ) . . . . .	19
2.7	Spectrum of a received signal for a frequency flat channel . . . . .	23
2.8	Statistic $ P_x $ of an OFDM symbol sent through a frequency flat channel . . . . .	23
2.9	Statistic $M_{opt}$ of an OFDM symbol sent through a frequency flat channel . . . . .	24
2.10	Spectrum of a received signal for a frequency selective channel . . . . .	25
2.11	Statistic $ P_x $ of an OFDM symbol sent through a frequency selective channel . . . . .	25
2.12	Statistic $M_{opt}$ of an OFDM symbol sent through a frequency selective channel . . . . .	26
2.13	Scatterplot of a received OFDM symbol . . . . .	29
2.14	Structure of a packet . . . . .	29
2.15	Illustration of the subcarriers used . . . . .	30
2.16	SRRC pulse shape in the time and frequency domains . . . . .	31
2.17	The Double-Sliding-Window set up initially . . . . .	33
2.18	The DSW sliding towards a received packet . . . . .	33
2.19	The signal window(window 2) gets a part of the packet, SNR_alert is raised . . . . .	34
2.20	As the DSW slides to the right, the SNR_alert reaches its limit . . . . .	34
2.21	The DSW gets out of the packet to continue for further detections . . . . .	35
3.1	Diagram of the USRPs' set up . . . . .	39
3.2	Diagram of the protocols' time-slots . . . . .	39
3.3	MIMO cable used for synchronization . . . . .	40
3.4	Image of 400x400 pixels selected for transmission . . . . .	41
3.5	Received images for PU slot occupancy probability equal to 0.6. Left: PU, Right: SU . . . . .	43

3.6	Received images for PU slot occupancy probability equal to 0.7. Left: PU, Right: SU . . . . .	43
3.7	Received images for PU slot occupancy probability equal to 0.8. Left: PU, Right: SU . . . . .	43
3.8	Received images for PU slot occupancy probability equal to 0.9. Left: PU, Right: SU . . . . .	44
3.9	Possible packet out of buffer during detection . . . . .	44
3.10	Selection of samples to transfer . . . . .	45
3.11	Transfer of samples to new buffer and filling with new samples . . . . .	45
3.12	Buffer is ready for packet detection . . . . .	45
3.13	Received images without/with diversity of order 2 from a distance of 2 metres . . . . .	46
3.14	Received images without/with diversity of order 2 from a distance of 4 metres . . . . .	47
3.15	Received images without/with diversity of order 2 from a distance of 6 metres . . . . .	47
3.16	Received images without/with diversity of order 2 from a distance of 8 metres . . . . .	48
3.17	Bit Error Rate for several distances of transmission without/with diversity of order 2 . . . . .	48



# Abbreviations

<b>UHD</b>	<b>U</b> niversal <b>H</b> ardware <b>D</b> river
<b>USRP</b>	<b>U</b> niversal <b>S</b> oftware <b>R</b> adio <b>P</b> eripheral
<b>PU</b>	<b>P</b> rimary <b>U</b> ser
<b>SU</b>	<b>S</b> econdary <b>U</b> ser
<b>SDR</b>	<b>S</b> oftware <b>D</b> efined <b>R</b> adio
<b>CFO</b>	<b>C</b> arrier <b>F</b> requency <b>O</b> ffset
<b>SRRC</b>	<b>S</b> quare <b>R</b> oot <b>R</b> aised <b>C</b> osine
<b>QAM</b>	<b>Q</b> uadrature <b>A</b> mplitude <b>M</b> odulation
<b>ISI</b>	<b>I</b> nter- <b>S</b> ymbol <b>I</b> nterference
<b>RF</b>	<b>R</b> adio <b>F</b> requency
<b>FPGA</b>	<b>F</b> ield <b>P</b> rogrammable <b>G</b> ate <b>A</b> rray
<b>ADC</b>	<b>A</b> nalog-to- <b>D</b> igital <b>C</b> onverter
<b>DAC</b>	<b>D</b> igital-to- <b>A</b> nalog <b>C</b> onverter
<b>DFT</b>	<b>D</b> iscrete <b>F</b> ourier <b>T</b> ransform
<b>IDFT</b>	<b>I</b> nverse <b>D</b> iscrete <b>F</b> ourier <b>T</b> ransform
<b>MSE</b>	<b>M</b> ean <b>S</b> quare <b>E</b> rror
<b>CP</b>	<b>C</b> yclic <b>P</b> refix
<b>LS</b>	<b>L</b> east <b>S</b> quares
<b>DSW</b>	<b>D</b> ouble <b>S</b> liding <b>W</b> indow
<b>PU_SOP</b>	<b>P</b> rimary <b>U</b> ser <b>S</b> lot <b>O</b> ccupancy <b>P</b> robability
<b>SNR</b>	<b>S</b> ignal to <b>N</b> oise <b>R</b> atio



*Dedicated to my soon to-be-born niece..*



# Chapter 1

## Background Knowledge

### 1.1 Orthogonal frequency-division multiplexing (*OFDM*)

Orthogonal frequency-division multiplexing (OFDM) is a method of encoding digital data on multiple carrier frequencies. OFDM has become a popular scheme for wide-band digital communication, used in applications such as digital television and audio broadcasting, DSL Internet access, wireless networks, powerline networks, and 4G mobile communications [1]. OFDM is a frequency-division multiplexing (FDM) scheme used as a digital multi-carrier modulation method. A large number of closely spaced orthogonal sub-carrier signals are used to carry data on several parallel data streams. Each sub-carrier is modulated with a conventional modulation scheme (such as quadrature amplitude modulation or phase-shift keying, 4-QAM in our implementation) at a low symbol rate, maintaining total data rates similar to conventional single-carrier modulation schemes in the same bandwidth.

An OFDM signal consists of a number of closely spaced modulated carriers. When modulation of any form - voice, data, etc. is applied to a carrier, then both sidebands spread out. It is necessary for a receiver to be able to receive the whole signal to be able to successfully demodulate the data. As a result, when signals are transmitted close to one another they must be spaced so that the receiver can separate them using a filter and there must be a guard band between them. This is not the case with OFDM. Although the sidebands from each carrier overlap, they can still be received without the interference that might be expected because they are orthogonal to each another. This is achieved by having the carrier spacing equal to the reciprocal of the symbol period.

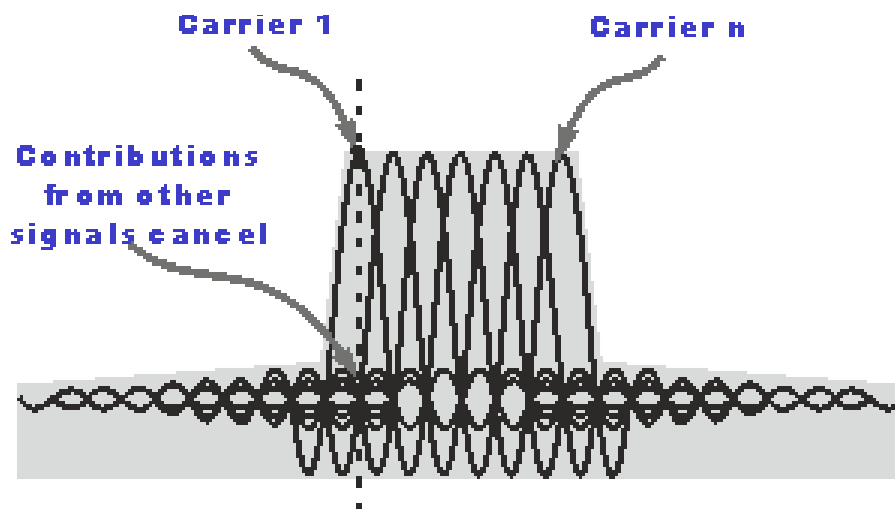


FIGURE 1.1: Carriers of an OFDM signal

To see how OFDM works, it is necessary to look at the receiver. This acts as a bank of demodulators, translating each carrier down to DC. The resulting signal is integrated over the symbol period to regenerate the data from that carrier. The same demodulator also demodulates the other carriers. As the carrier spacing is equal to the reciprocal of the symbol period, a whole number of cycles in the symbol period and their contribution will sum to zero - in other words there is no interference contribution.

To sum up, we present the general advantages and disadvantages of the Orthogonal Frequency-Division Multiplexing:

## OFDM advantages

- **Immunity to selective fading:** One of the main advantages of OFDM is that it is more resistant to frequency selective fading than single carrier systems because it divides the overall channel into multiple narrowband signals that are affected individually as flat fading sub-channels.
- **Resilience to interference:** Interference appearing on a channel may be bandwidth limited and in this way will not affect all the sub-channels. This means that not all the data is lost.
- **Spectrum efficiency:** Using close-spaced overlapping sub-carriers, a significant OFDM advantage is that it makes efficient use of the available spectrum.

- **Resilience to ISI:** Another advantage of OFDM is that it is very resilient to inter-symbol and inter-frame interference. This results from the low data rate on each of the sub-channels.
- **Resilience to narrow-band effects:** Using adequate channel coding and interleaving it is possible to recover symbols lost due to the frequency selectivity of the channel and narrow band interference. Not all the data is lost.
- **Simpler channel equalisation:** One of the issues with CDMA systems was the complexity of the channel equalisation which had to be applied across the whole channel. An advantage of OFDM is that using multiple sub-channels, the channel equalization becomes much simpler.

## OFDM disadvantages

- **High peak to average power ratio:** An OFDM signal has a noise like amplitude variation and has a relatively high large dynamic range, or peak to average power ratio. This impacts the RF amplifier efficiency as the amplifiers need to be linear and accommodate the large amplitude variations and these factors mean the amplifier cannot operate with a high efficiency level.
- **Sensitive to carrier offset and drift:** Another disadvantage of OFDM is that it is sensitive to carrier frequency offset and drift. Single carrier systems are less sensitive.

## 1.2 Software Defined Radio (SDR)

With the exponential growth in the ways and means by which people need to communicate - data communications, voice communications, video communications, broadcast messaging, command and control communications, emergency response communications, etc. – modifying radio devices easily and cost-effectively has become business critical. Software defined radio (SDR) technology brings the flexibility, cost efficiency and power to drive communications forward, with wide-reaching benefits realized by service providers and product developers through to end users.

A number of definitions can be found to describe Software Defined Radio, also known as Software Radio or SDR. The SDR Forum, working in collaboration with the Institute of Electrical and Electronic Engineers (IEEE) P1900.1 group, has worked to establish a definition of SDR that provides consistency and a clear overview of the technology and its associated benefits.

Simply put Software Defined Radio is defined as:

*Radio in which some or all of the physical layer functions are software defined.*

A radio is any kind of device that wirelessly transmits or receives signals in the radio frequency (RF) part of the electromagnetic spectrum to facilitate the transfer of information. In today's world, radios exist in a multitude of items such as cell phones, computers, car door openers, vehicles, and televisions. Traditional hardware based radio devices limit cross-functionality and can only be modified through physical intervention. This results in higher production costs and minimal flexibility in supporting multiple waveform standards. By contrast, software defined radio technology provides an efficient and comparatively inexpensive solution to this problem, allowing multimode, multi-band and/or multi-functional wireless devices that can be enhanced using software upgrades.

### 1.2.1 Universal Software Radio Peripheral(USRP)

Universal Software Radio Peripheral (USRP) is a range of software-defined radios designed and sold by Ettus Research and its parent company, National Instruments. Developed by a team led by Matt Ettus, the USRP product family is intended to be a comparatively inexpensive hardware platform for software radio, and is commonly used by research labs, universities, and hobbyists. The USRP product family includes a variety of models that use a similar architecture. A motherboard provides the following subsystems: clock generation and synchronization, FPGA, ADCs, DACs, host processor interface, and power regulation. These are the basic components that are required for baseband processing of signals. A modular front-end, called a daughterboard, is used for analog operations such as up/down-conversion, filtering, and other signal conditioning. This modularity permits the USRP to serve applications that operate between DC and 6 GHz.

### 1.2.2 USRP N200, a quick glimpse

The USRPs in our configuration are from the Networked Series, the N200. The USRP N200 are high-performance USRP devices that provide higher dynamic range and higher bandwidth than the bus series. An optional GPDSO module can also be used to discipline the USRP N200 reference clock to within 0.01 ppm of the worldwide GPS standard. Using a Gigabit Ethernet interface, the devices in the Networked Series can transfer up to 50 MS/s of complex, baseband samples to/from the host. This series uses a dual, 14-bit, 100 MS/s ADC and dual 16-bit, 400 MS/s DAC. This series also provides a MIMO expansion port which can be used to synchronize two devices from this series.





---

FIGURE 1.2: USRP N200

**The USRP N200 contains the following hardware components:**

- A Xilinx Spartan-3A DSP 1800 FPGA
- Gigabit Ethernet interface
- Dual 100 MS/s, 14-bit, analog-to-digital converter
- Dual 400 MS/s, 16-bit, digital-to-analog converter
- Flexible Clocking and Synchronization
  - External Inputs for 10 MHz and 1 PPS signals (SMA)
  - Optional GPS Disciplined Oscillator
  - Ettus Research MIMO Cable that can be used to synchronize two USRP devices

### 1.2.3 Daughterboards of our implementation

The USRP family features a modular architecture with interchangeable daughterboard modules that serve as the RF front end. Several classes of daughterboard modules exist: Receivers, Transmitters and Transceivers. The daughterboards used in the USRPs of our implementation are two CBXs and a WBX as they are the newest and best products

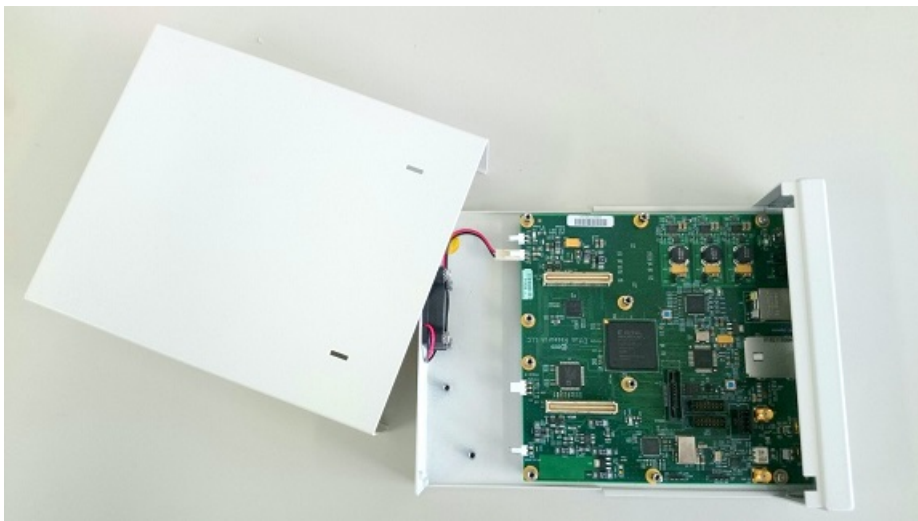


FIGURE 1.3: USRP N200 interior

from Ettus and also available at our lab. The CBX daughterboard is a full-duplex, wideband transceiver that covers a frequency band from 1.2 GHz to 6 GHz with an instantaneous bandwidth of 40 MHz while its transmit and receive gain range from 0 to 31.5 dB and can be set independently. On the other hand, the WBX daughterboard is a full-duplex, wideband transceiver that covers a frequency band from 50 MHz to 2.2 GHz, providing 40 MHz of bandwidth capability while its transmit gain ranges from 0 to 25 dB and its receive gain ranges from 0 to 31.5dB and can be set independently. Both types of daughterboards can serve a wide variety of application areas, including WiFi research, cellular base stations, cognitive radio research and RADAR.

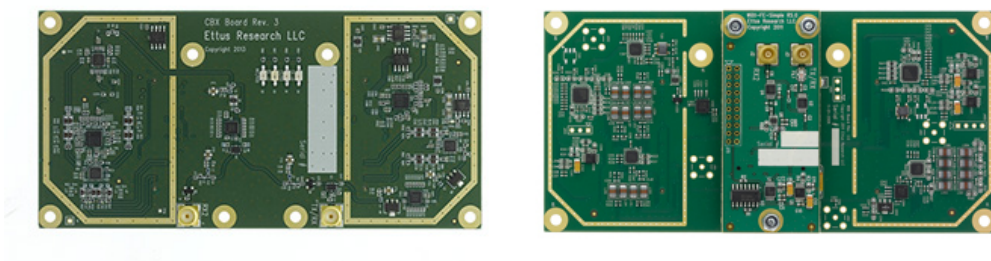


FIGURE 1.4: USRP daughterboards used in the implementation - Left: CBX, Right: WBX

### 1.3 Cognitive radio

A cognitive radio is an intelligent wireless communication device that can be programmed and configured dynamically. Its transceiver exploits side information about its environment to improve spectrum utilization. This side information typically comprises knowledge about the activity, channels, codebooks, and/or messages of other nodes with which the cognitive node shares the spectrum. Based on the nature of the available side information as well as a priori rules about spectrum usage, cognitive radio systems seek to underlay, overlay, or interweave the cognitive radios' signals with the transmissions of noncognitive nodes. Such a radio automatically detects available channels in wireless spectrum, then accordingly changes its transmission or reception parameters to allow more concurrent wireless communications in a given spectrum band at one location. This process is a form of dynamic spectrum management.

### 1.4 Software needed in the implementation

The following subsections contain useful information and details considering the software utilized in this thesis in order to create our OFDM testbed.

#### 1.4.1 Matlab

All the mathematical processing done in this work has been initially written and tested thoroughly in MATLAB in order to find the best functioning algorithms and methods to create, modify and process the data that is sent/received from/by the corresponding USRPs.

MATLAB (matrix laboratory) is a multi-paradigm numerical computing environment and fourth-generation programming language. A proprietary programming language developed by MathWorks, MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces and interfacing with programs written in other languages, including C, C++, Java, Fortran and Python.

#### 1.4.2 Usrc Hardware Drivers(UHD)

The USRP hardware driver (UHD) is the device driver provided by Ettus Research for use with the USRP product family. It supports Linux, MacOS, and Windows platforms. Several frameworks including GNU Radio, LabVIEW, MATLAB and Simulink use UHD. The functionality provided by UHD can also be accessed directly with the UHD API,

which provides native support for C++. Any other language that can import C++ functions can also use UHD.

After deciding the best ways to create, modify and process data for our purpose using Matlab, all the code is re-written in C++ language and additional code regarding the PC-USRP communication is added. The host PC-USRP communication is accomplished by installing and using the UHD drivers that are provided by Ettus, the manufacturer of the USRP platform. The C++ language is preferred in order to achieve realtime data processing and create all the necessary code based on the project's needs while our operating system is the Ubuntu 14.04.

### 1.4.3 Armadillo library

Most mathematical operations in the C++ code have been made with the use of the open-source Armadillo library. Armadillo is a linear algebra software library for the C++ programming language. It aims to provide efficient and streamlined base calculations, while at the same time having a straightforward and easy-to-use interface. Its intended target users are scientists and engineers.

It supports integer, floating point (single and double precision), complex numbers and a subset of trigonometric and statistics functions. Various matrix decompositions are provided through optional integration with Linear Algebra PACKage (LAPACK) and Automatically Tuned Linear Algebra Software (ATLAS) libraries. High-performance LAPACK replacement libraries such as Math Kernel Library (MKL) and AMD Core Math Library (ACML) can also be used.

### 1.4.4 Matlab Engine

In order to create useful figures such as plots of constellations or synchronisation statistics using data stored in C++ matrices and variables, while running C++ code, Matlab was the way to go again. The only way to achieve this pretty easily was by using the Matlab Engine. Matlab Engine allows a stand alone C or C++ program to call Matlab to perform a calculation or execute a command. This allows data to be sent directly to Matlab and use existing Matlab code to analyze data (and even send it back to the C++ program, for closed loop applications).

### 1.4.5 Gnu Radio

A usual SDR application, GNU Radio, was used so as to observe the spectrum while sending packets from the transmitter or store incoming samples to a file and process

them afterwards with Matlab. In this way, we could reach to useful solutions for numerous problems or implementation details needed to be sorted out before proceeding. GNU Radio is a free software development toolkit that provides signal processing blocks to implement software-defined radios and signal processing systems. It can be used with 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.

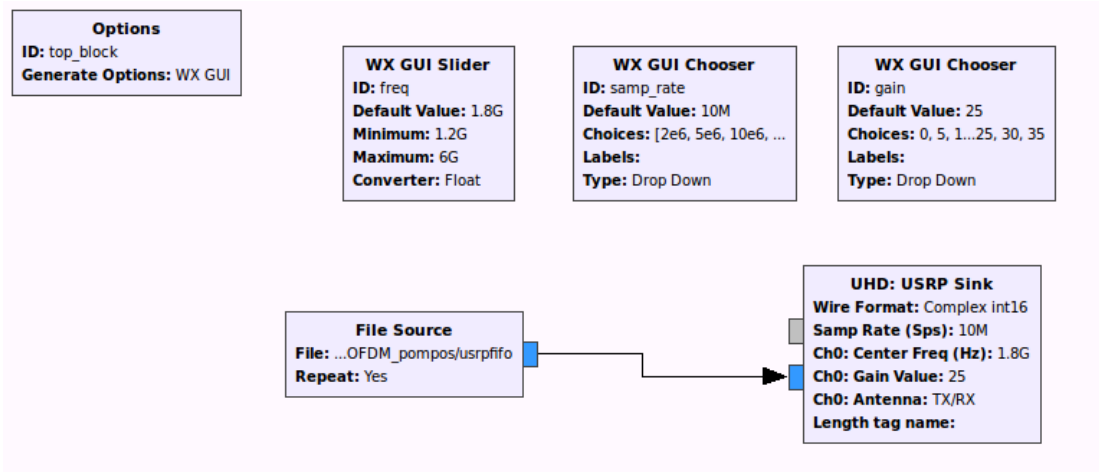


FIGURE 1.5: GNU Radio assembly used to transmit Matlab-created data for initial design

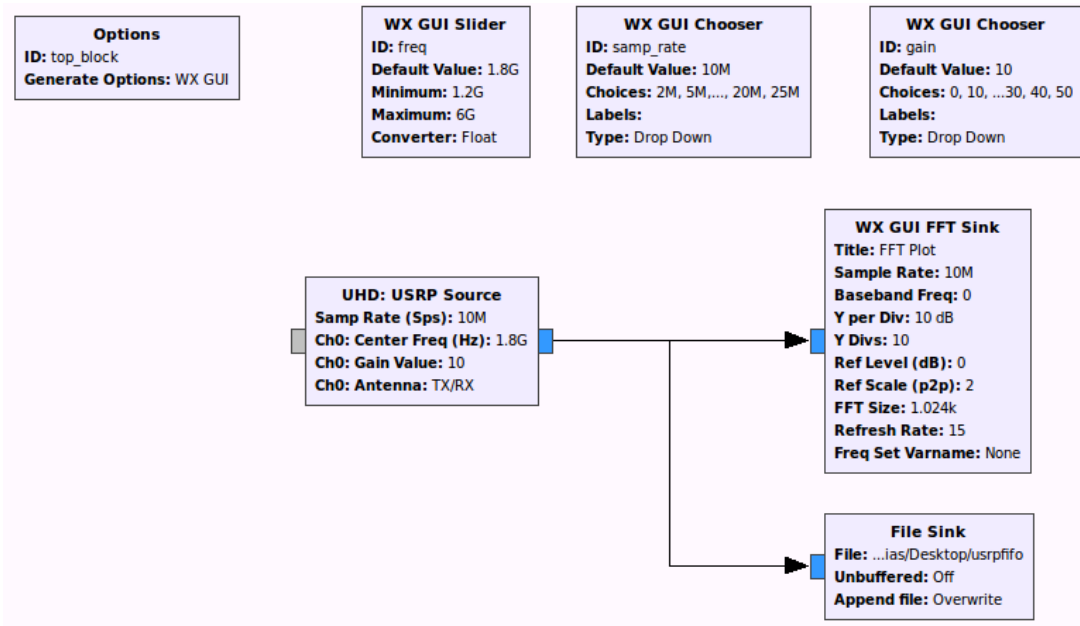


FIGURE 1.6: GNU Radio assembly used to receive data while plotting the corresponding spectrum

## 1.5 Thesis outline

In Chapter 1, we were introduced to the basic ideas that are going to be developed in depth in this thesis. Furthermore, we took a good look at the majority of the software needed for the prototyping and final development of this work.

Chapter 2 presents the detailed work that was done in order to create a fully functional receiver and transmitter that use the OFDM symbols in order to exchange data between them. In the beginning, we are going to see analog and digital OFDM representations, followed by an elaborate description of the time and frequency synchronization methods used. Moreover, a precise channel estimation method is followed so as to find the values of all the channel taps, whether the channel is flat or frequency selective. Details on the technical side of the implementation are given as well, to take a look at the challenges met while developing the C++ code.

Chapter 3 presents a time slotted protocol with a master transmitter (Primary User-PU) and a slave transceiver (Secondary User-SU) that acts as a cognitive radio. All the procedures described in chapter 2 are used as is or firstly modified to better serve this application's needs. The target is to keep all time-slots occupied by the SU, when the PU is not transmitting data, by using spectrum sensing to find the spectral gaps. In the end, the application is further enriched so that both the master and the slave transmit a specific picture's parts during their time-slots. A general receiver gathers all the packets sent by them, assembling each picture and calculating the transmission's errors. A further study is made based on the transmission's errors at various distances between the transmitter and the receiver, while the transmission has or not, time diversity.

Finally, Chapter 4 consists of a brief outline of future ideas that can be used to further evolve the work that was done in this thesis.

## Chapter 2

# Approach of an OFDM based system

### 2.1 Structure and analysis of the OFDM scheme

In this section, we are going to take a look at the basic structure of an OFDM system. Specifically, we will understand how we have to manipulate the data at both sides of the transmitter and the receiver in order to assign it to subcarriers and acquire it from the subcarriers received, respectively. We will not deal with time and frequency synchronization or channel estimation in this section but later in the following sections. Finally, it will be clarified why OFDM communication doesn't require complex equalization at the receiver in order to reverse the channel's effect.

Consider a wideband wireless channel, with a discrete-time impulse response given by  $h_l$ ,  $l = 0, \dots, L - 1$ . We assume that the channel remains constant over the time period that we are interested in [2].

If  $x[m]$  is the input at the time instant  $m$ , then the output  $y[m]$  is given from the following convolution :

$$y[m] = \sum_{l=0}^{L-1} h_l x[m-l] + w[m]$$

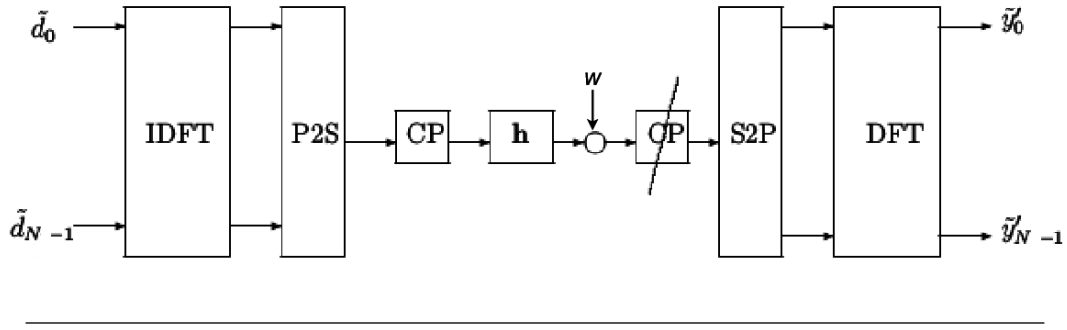


FIGURE 2.1: An OFDM system

Let the data block of length  $N$  be

$$\tilde{\mathbf{d}} = \begin{bmatrix} \tilde{d}[0] \\ \vdots \\ \tilde{d}[N-1] \end{bmatrix}$$

Taking the Inverse Discrete Fourier Transform (IDFT) of  $\tilde{\mathbf{d}}$ , the data block is expressed as

$$\mathbf{d} = IDFT(\tilde{\mathbf{d}}) = [d[0], \dots, d[N-1]]^T.$$

Using  $\mathbf{d}$ , we construct the vector  $\mathbf{x}$ , by inserting a cyclic prefix of length  $L$  (to eliminate intersymbol interference (ISI) arising from the wideband channel),

$$\mathbf{x} = \begin{bmatrix} d[N-L] \\ \vdots \\ d[N-1] \\ d[0] \\ \vdots \\ d[N-1] \end{bmatrix} = \begin{bmatrix} x[0] \\ \vdots \\ \vdots \\ x[N+L-1] \end{bmatrix}.$$

Using  $\mathbf{x}$  as input to the channel, the output (assuming perfect synchronization at the receiver) is written as

$$y[m] = \sum_{l=0}^{L-1} h_l x[m-l] + w[m], \quad m = 0, \dots, N+L-1.$$



At the receiver we ignore the first  $L$  output symbols. Using the  $N$  output symbols  $y[m]$ ,  $m = L, \dots, N + L - 1$ , we construct the vector

$$\mathbf{y}' = \begin{bmatrix} y'[0] \\ \vdots \\ y'[N-1] \end{bmatrix} = \begin{bmatrix} y[L] \\ \vdots \\ y[N+L-1] \end{bmatrix}.$$

It can be proven that

$$\mathbf{y}' = \mathbf{d} \otimes_N \mathbf{h} + \mathbf{w},$$

where  $\mathbf{w} = [w[L], \dots, w[N+L-1]]^T$  and  $\mathbf{a} \otimes_N \mathbf{b}$  is the circular convolution, of length  $N$ , of vectors  $\mathbf{a}$  and  $\mathbf{b}$ . To prove the above assertion, consider

$$y'[0] = y[L] = \sum_{l=0}^{L-1} h_l x[L-l] = h_0 d[0] + \sum_{l=1}^{L-1} h_l d[N-l].$$

Respectively, the first term of the circular convolution is expressed as

$$(\mathbf{d} \otimes_N \mathbf{h})[0] = h_0 d[0] + \sum_{l=1}^{L-1} h_l d[N-l].$$

In a similar way, we can prove the remaining relations and conclude that  $\mathbf{y}' = \mathbf{d} \otimes_N \mathbf{h} + \mathbf{w}$ . Taking the Discrete Fourier Transform (DFT) of both sides, we obtain

$$\begin{aligned} \tilde{\mathbf{y}}' &= DFT(\mathbf{y}') = DFT(\mathbf{d} \otimes_N \mathbf{h} + \mathbf{w}) \\ &= DFT(\mathbf{d} \otimes_N \mathbf{h}) + DFT(\mathbf{w}) \\ &= \sqrt{N} DFT(\mathbf{h}) \odot DFT(\mathbf{d}) + DFT(\mathbf{w}) \end{aligned} \tag{2.1}$$

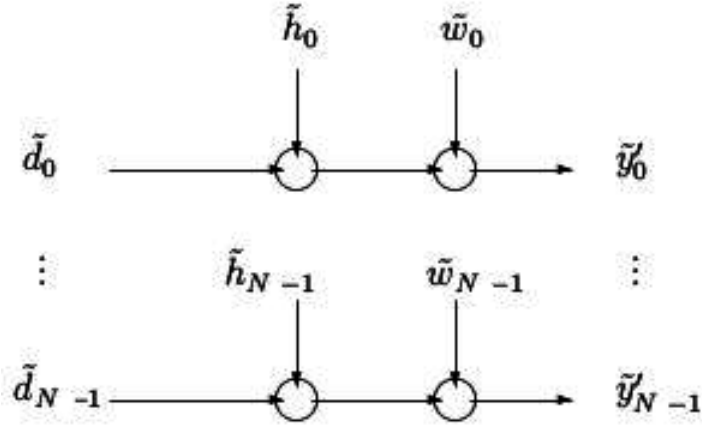
where

$$\begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \odot \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} x_1 y_1 \\ \vdots \\ x_n y_n \end{bmatrix}$$

is the element-wise or Hadamard product. Consequently,

$$\tilde{y}'_n = \tilde{h}_n \tilde{d}_n + \tilde{w}_n, \quad n = 0, \dots, N-1, \tag{2.2}$$

where  $\tilde{h}_n = \sum_{l=0}^{L-1} h_l e^{-j\frac{2\pi l n}{N}}$ ,  $n = 0, \dots, N-1$ . Thus, using OFDM we have converted a wideband channel into a set of  $N$  parallel narrowband channels. As a result, no equalization is required, which has a high computational cost, but a symbol-by-symbol decision for each information symbol.

FIGURE 2.2: A frequency selective channel turns into  $N$  parallel flat fading channels

## 2.2 An analog analysis of the subcarriers

High data-rate is desired in many applications. However, as the symbol duration reduces with the increase of data-rate, the systems using single-carrier modulation suffer from more severe intersymbol interference (ISI) caused by the dispersive fading of wireless channels, thereby needing more complex equalization. OFDM modulation divides the entire frequency selective fading channel into many narrowband flat fading subchannels in which high-bit-rate data are transmitted in parallel and do not undergo ISI due to the long symbol duration [1].

Let  $\{s_k\}_{k=0}^{N-1}$  be the complex symbols to be transmitted by OFDM modulation; the OFDM (modulated) signal can be expressed as

$$s(t) = \sum_{k=0}^{N-1} s_k e^{j2\pi f_k t}, \quad 0 \leq t \leq T_s \quad (2.3)$$

where:

$f_k = f_0 + k\Delta f$  : frequency of the  $k$ -th subcarrier

$f_0$  : Rf carrier frequency

$\Delta f$  : subcarrier spacing of OFDM

$N$  : number of subcarriers

$T_s$  : symbol duration of OFDM

and

$$\phi_k(t) = \begin{cases} e^{j2\pi f_k t}, & \text{if } 0 \leq t \leq T_s, \\ 0, & \text{otherwise.} \end{cases} \quad (2.4)$$

for  $k = 0, 1, \dots, N - 1$  as mentioned above.

In order for the receiver to demodulate an OFDM signal, the symbol duration must be long enough such that  $T_s \Delta_f = 1$ , which is also called orthogonality condition. Because of the orthogonality condition, we have

$$\begin{aligned} & \frac{1}{T_s} \int_0^{T_s} \phi_k(t) \phi_l^*(t) dt \\ &= \frac{1}{T_s} \int_0^{T_s} (e^{j2\pi f_k t}) (e^{j2\pi f_l t})^* dt \\ &= \frac{1}{T_s} \int_0^{T_s} e^{j2\pi (f_k - f_l) t} dt \\ &= \frac{1}{T_s} \int_0^{T_s} e^{j2\pi (k-l) \Delta_f t} dt \\ &= \delta[k - l], \end{aligned} \quad (2.5)$$

where  $\delta[k - l]$  is the delta function defined as

$$\delta[n] = \begin{cases} 1, & \text{if } n=0, \\ 0, & \text{otherwise.} \end{cases}$$

Equation (2.5) shows that  $\{\phi_k(t)\}_{k=0}^{N-1}$  is a set of orthogonal functions. Using this property, the OFDM signal can be demodulated by

$$\begin{aligned} & \frac{1}{T_s} \int_0^{T_s} s(t) e^{-j2\pi f_k t} dt \\ &= \frac{1}{T_s} \int_0^{T_s} \left( \sum_{l=0}^{N-1} s_l \phi_l(t) \right) \phi_k^*(t) dt \\ &= \sum_{l=0}^{N-1} s_l \delta[l - k] \\ &= s_k. \end{aligned} \quad (2.6)$$

### 2.3 From analog signal to digital samples

In order to study the synchronization problems, we present a detailed signal model for OFDM modulation-demodulation [3]. Let the baseband-equivalent OFDM signal that is transmitted through the channel be

$$S(t) = \sum_n s_n g_T(t - nT),$$

where  $g_T(t)$  is the pulse shaping filter. The channel output is

$$Y(t) = c(t) * S(t) + W(t) = c(t) * \left( \sum_n s_n g_T(t - nT) \right) + W(t) = \sum_n s_n h(t - nT) + W(t),$$

where  $h(t) = c(t) * g_T(t)$ . With analog CFO  $\Delta F$  and phase offset  $\phi$ , the received signal is

$$Y(t) = e^{j(2\pi\Delta Ft + \phi)} \sum_n s_n h(t - nT) + W(t).$$

If we sample with period  $T_s = \frac{T}{\text{over}}$ , where *over* is a positive integer, we obtain the *sample-spaced* sequence

$$\begin{aligned} y_k &= Y(kT_s) = e^{j(2\pi\Delta FkT_s + \phi)} \sum_n s_n h(kT_s - nT) + W(kT_s) \\ &= e^{j\phi} e^{j2\pi\Delta f k} \sum_n s_n h_{k,n} + w_k, \end{aligned} \tag{2.7}$$

where  $\Delta f := \Delta FT_s$  and  $h_{k,n} := h(kT_s - nT)$ . Then,  $y_k$  can be expressed as

$$y_k = e^{j\phi} e^{j2\pi\Delta f k} r_k + w_k,$$

where  $r_k := \sum_n s_n h_{k,n}$ .

### 2.4 Synchronization in OFDM

Orthogonal Frequency Division Multiplexing (OFDM) is a popular scheme for fixed and wireless broadband applications due to its robustness against frequency selectivity of the channel. However, OFDM systems are very sensitive to carrier frequency errors and quite sensitive to timing errors. Schmidl [4] proposed using the autocorrelation of a training symbol with two identical parts to estimate timing and fractional frequency offset. An additional training symbol is then used along with the first to determine the integer frequency offset. However, Schmidl's timing metric has an uncertainty plateau and the method can yield timing estimates which are well beyond the ISI-free region, thus leading

to degradation in BER performance. Other autocorrelation methods have been proposed by Park [5], Minn [6] and Shi [7] using different uniquely-designed preamble patterns to obtain sharper timing metrics and improve the timing accuracy. However, they are not robust in fading and strong-ISI channels. When speaking of synchronization, most algorithms are based on two important principles; autocorrelation and crosscorrelation. In the autocorrelation process, the signal is correlated with itself, whereas in the cross-correlation, the signal is correlated with a stored pattern known to the receiver. The algorithm used in our work [8] presents a new method in order to achieve near-ideal accuracy (i.e. timing MSE approaching zero) without significant increase in complexity. This method is multi-stage, wherein a simple autocorrelation technique and a novel restricted cross-correlation technique are combined to achieve enhanced estimation performance.

(**Note:** all the following algorithms assume a symbol spaced sequence.)

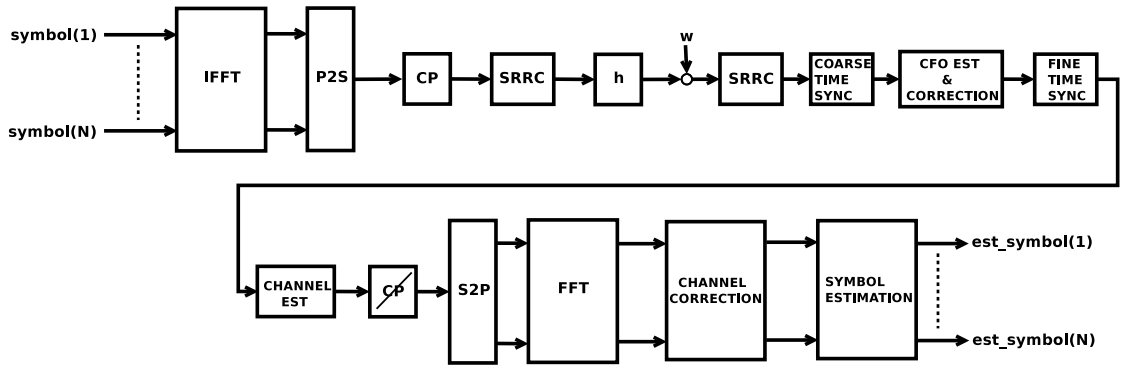


FIGURE 2.3: Our OFDM system structure

### 2.4.1 Coarse time synchronization

A preamble with two identical parts in time-domain (e.g. Schmidl's training symbol  $S_{Sch}$  [4]) is used in order to achieve our synchronization goals. In this preamble, the second half is equal to the first half. This is equivalent to using only every other tone in the OFDM symbol. This synchronization symbol is sent before every OFDM symbol in order to perform perfect sync every time an OFDM symbol is received. The sync symbol's structure is as follows

$$S_{Sch} = [d_{N/2} \ d_{N/2}]$$

This preamble structure also conforms to the WiFi and WiMAX standards, wherein only the 2 long identical patterns are used in the proposed method while the preceding short

patterns are excluded via the autocorrelation stage which can differentiate between the short and long patterns.

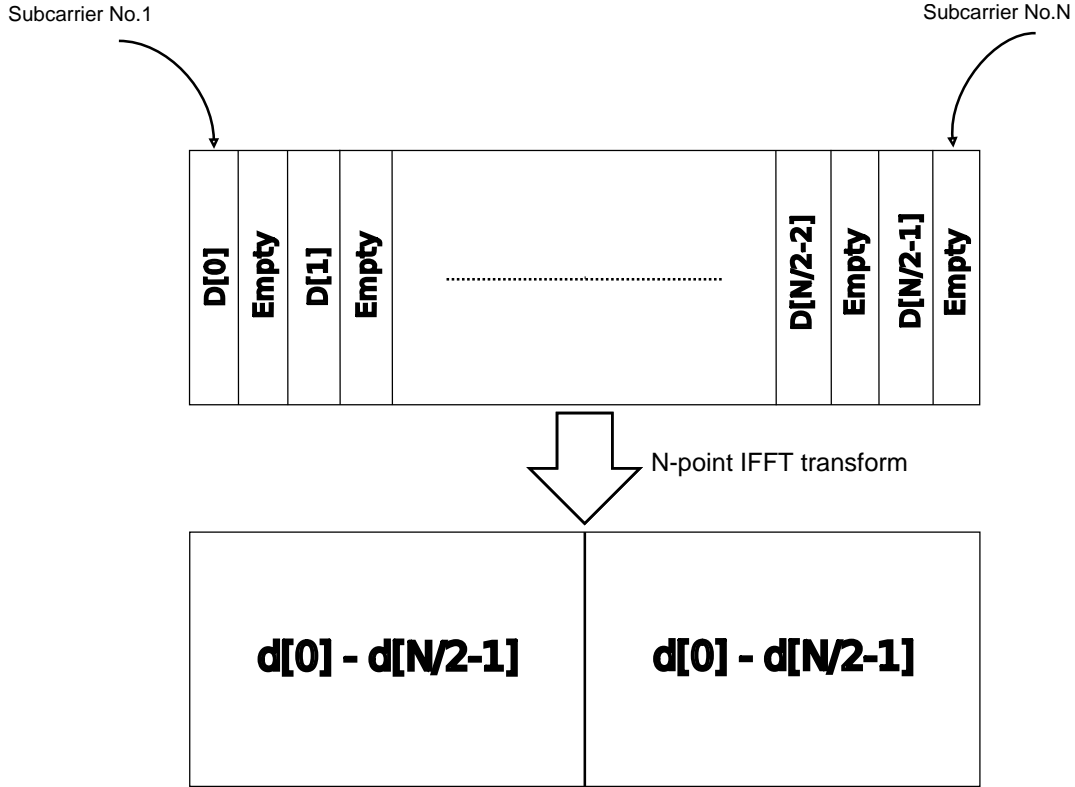


FIGURE 2.4: Structure of the synchronization symbol or preamble  $S_{Sch}$

The coarse timing estimate, where  $y(d)$  is the received sequence, is derived as follows

$$P_{Sch}(d) = \sum_{k=0}^{N/2-1} y^*(d+k) y(d+k+N/2) \quad (2.8)$$

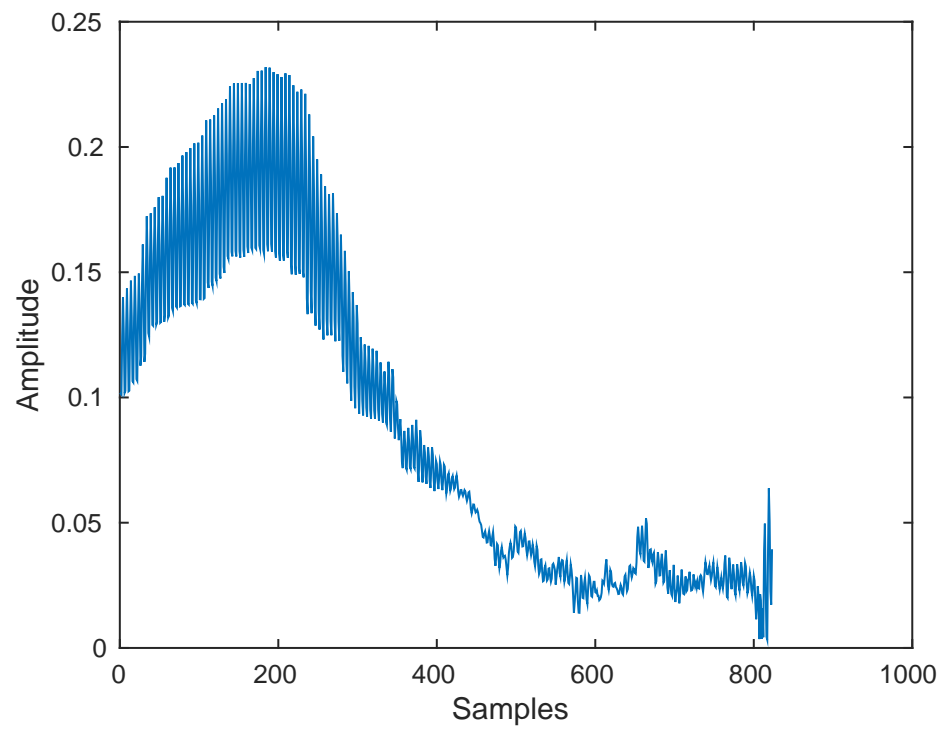
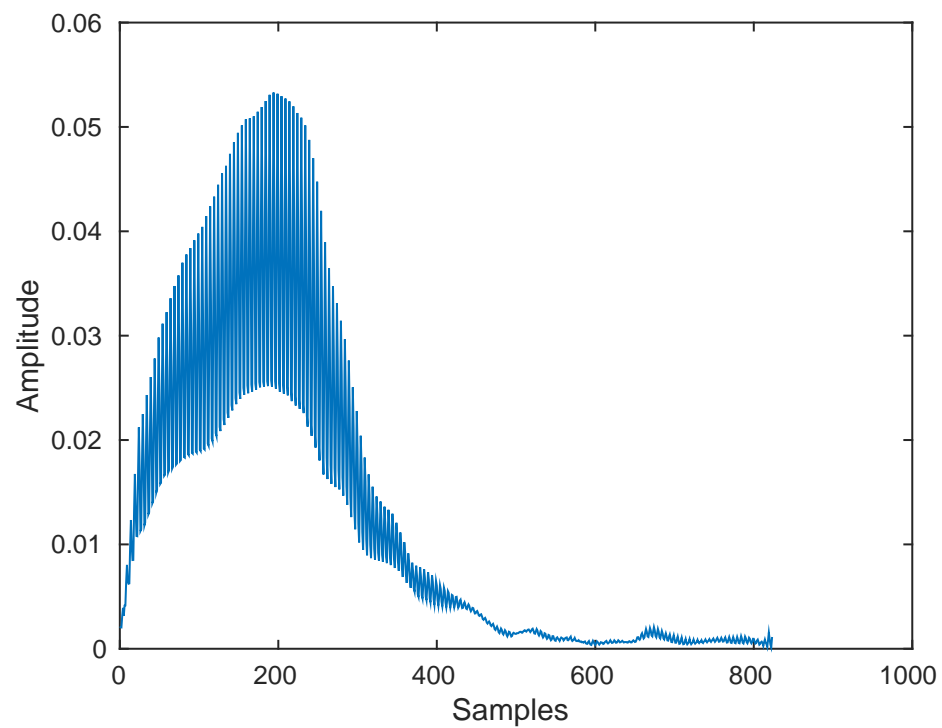
$$M_c(d) = \frac{1}{L+1} \sum_{k=0}^L |P_{Sch}(d-k)|^2 \quad (2.9)$$

$$\hat{d}_c = \underset{d}{\operatorname{argmax}} \{M_c(d)\} \quad (2.10)$$

In case of a sampled-space received sequence, the equations (2.8) to (2.9) change to

$$P_{Sch}(d) = \sum_{k=0}^{N/2-1} y^*(d+k \cdot \text{over}) y(d+(k+N/2) \cdot \text{over}) \quad (2.11)$$

$$M_c(d) = \frac{1}{L+1} \sum_{k=0}^L |P_{Sch}(d-k \cdot \text{over})|^2 \quad (2.12)$$

FIGURE 2.5: Synchronization statistic of equation 2.8 ( $P_{Sch}$ )FIGURE 2.6: Synchronization statistic of equation 2.9 ( $M_c$ )

After the autocorrelation stage of (2.8), we have an integration over the length of the cyclic prefix (2.9) in order to eliminate its uncertainty plateau and achieve a coarse timing metric  $M_c$  whose peak indicates the coarse timing estimate  $\hat{d}_c$  (2.10).

So, our coarse estimate for the length- $N$  received synchronization sequence is

$$y'_0 = y_{\hat{d}_c}, y'_1 = y_{\hat{d}_c+1}, \dots, y'_{N-1} = y_{\hat{d}_c+(N-1)}. \quad (2.13)$$

Assuming that the received sequence is *sample-spaced* and the estimate of (2.10) has resulted from the equations (2.11) and (2.12), then we can express the *symbol-spaced* sequence  $\{y'_l\}_{l=0}^{N-1}$  of (2.13) as

$$\begin{aligned} y'_l &= Y(\hat{d}_c T_s + lT) = Y(t)|_{t=\hat{d}_c T_s + lT} \\ &= e^{j(2\pi\Delta F(\hat{d}_c T_s + lT) + \phi)} \sum_n s_n h(\hat{d}_c T_s + lT - nT) + W(\hat{d}_c T_s + lT) \\ &= e^{j(2\pi\Delta f \hat{d}_c + \phi)} e^{j2\pi(\Delta F \cdot T)l} \sum_n s_n h_{l,n}^{\hat{d}_c} + w'_l, \end{aligned} \quad (2.14)$$

where  $h_{l,n}^{\hat{d}_c} := h(\hat{d}_c T_s + lT - nT)$ .

If we incorporate the constant term  $e^{j(2\pi\Delta f \hat{d}_c + \phi)}$  into the channel and, for simplicity, denote

$$h'_{l,n} = e^{j(2\pi\Delta f \hat{d}_c + \phi)} h_{l,n}^{\hat{d}_c}, \quad (2.15)$$

then, for  $l = 0, \dots, N-1$ ,

$$y'_l = e^{j2\pi(\Delta F \cdot T)l} \sum_n s_n h'_{l,n} + w'_l = e^{j2\pi\Delta f' l} r'_l + w'_l, \quad (2.16)$$

where  $\Delta f' = \Delta F \cdot T$  and  $r'_l = \sum_n s_n h'_{l,n}$ .

### 2.4.2 Estimation and correction of Carrier Frequency Offset(CFO)

The next step is to mitigate the frequency offset in order to retrieve all our symbols successfully. We can easily take the samples that correspond to the received part of the preamble (without its CP) by using the coarse timing estimate  $\hat{d}_c$ .

Assuming that the received samples corresponding to the first half of the training block are given by

$$y'_k = e^{j2\pi\Delta f' k} r'_k + w'_k, \quad k = 1, \dots, \frac{N}{2},$$

---

while (2.10) remains the same



respectively, the samples in the second half take the form

$$y'_{k+\frac{N}{2}} = e^{j2\pi\Delta f'(k+\frac{N}{2})} r'_{k+\frac{N}{2}} + w'_{k+\frac{N}{2}}, \quad k = 1, \dots, \frac{N}{2},$$

where  $r'_k$  and  $r'_{k+\frac{N}{2}}$  are identical. Consequently,

$$\begin{aligned} (y'_k)^* y'_{k+\frac{N}{2}} &= (e^{j2\pi\Delta f'k} r'_k + w'_k)^* (e^{j2\pi\Delta f'(k+\frac{N}{2})} r'_{k+\frac{N}{2}} + w'_{k+\frac{N}{2}}) \\ &= (e^{-j2\pi\Delta f'k} (r'_k)^* + (w'_k)^*) (e^{j2\pi\Delta f'(k+\frac{N}{2})} r'_{k+\frac{N}{2}} + w'_{k+\frac{N}{2}}) \\ &= e^{-j2\pi\Delta f'k} e^{j2\pi\Delta f'(k+\frac{N}{2})} (r'_k)^* r'_{k+\frac{N}{2}} + e^{-j2\pi\Delta f'k} (r'_k)^* w'_{k+\frac{N}{2}} + e^{j2\pi\Delta f'(k+\frac{N}{2})} r'_k (w'_k)^* + (w'_k)^* w'_{k+\frac{N}{2}} \\ &= e^{j2\pi\Delta f'(k+\frac{N}{2}-k)} |r'_k|^2 + \tilde{w}_k \\ &= e^{j\pi\Delta f'N} |r'_k|^2 + \tilde{w}_k \end{aligned} \tag{2.17}$$

where

$$\tilde{w}_k = e^{-j2\pi\Delta f'k} (r'_k)^* w'_{k+\frac{N}{2}} + e^{j2\pi\Delta f'(k+\frac{N}{2})} r'_k (w'_k)^* + (w'_k)^* w'_{k+\frac{N}{2}}.$$

Ignoring the noise part, if we take the argument of  $y_k^* y_{k+\frac{N}{2}}$  and use all the samples of the preamble part, an estimate of CFO,  $\hat{\Delta}f$ , can be derived as

$$\hat{\Delta}f = \frac{1}{\pi N} \arg \left( \sum_{k=0}^{\frac{N}{2}-1} (y'_k)^* y'_{k+\frac{N}{2}} \right),$$

because

$$\arg \left( \sum_{k=0}^{\frac{N}{2}-1} (y'_k)^* y'_{k+\frac{N}{2}} \right) = \arg \left( e^{j\pi\Delta f'N} \sum_{k=0}^{\frac{N}{2}-1} |r'_k|^2 \right) = \pi\Delta f'N.$$

After calculating  $\hat{\Delta}f$ , we can use it to eliminate the effect of CFO by creating the following sequence

$$y_{cor}'_k = e^{-j2\pi\hat{\Delta}fk} y'_k$$

where  $y_{cor}'_k$  is the corrected sequence of all the samples of the packet that are only affected from the channel.

### 2.4.3 Fine time synchronization

Our final step in the synchronization method is to accurately find the exact sample to start demodulating our OFDM symbols. So, we now have the total-frequency corrected signal  $y'_{cor}$  and we apply the cross-correlation stage that is summarized as follows:

$$\begin{aligned}
P_x(d) &= \sum_{k=0}^{N-1} y'_{cor}(d+k) S_{sch}^*(k) \\
d &\in \{\hat{d}_c - N/2, \hat{d}_c + N/2\}
\end{aligned} \tag{2.18}$$

Now, by using (2.9) and (2.18), we calculate the final statistic  $M_{opt}$

$$\begin{aligned}
M_{opt}(d) &= |P_x(d)|^2 M_c(d) \\
d &\in \{\hat{d}_c - N/2, \hat{d}_c + N/2\}
\end{aligned} \tag{2.19}$$

Finally, we acquire the fine timing estimate using (2.19), as

$$\begin{aligned}
\hat{d}_{opt} &= \underset{d}{\operatorname{argmax}} \{M_{opt}(d)\} \\
d &\in \{\hat{d}_c - N/2, \hat{d}_c + N/2\}
\end{aligned} \tag{2.20}$$

It is important to note that in cases of channels with frequency selective fading, we have to use a sliding window with size of  $L$  symbols that calculates the sum of the energies of the samples in it (i.e. 2.21) instead of using (2.20). In that way, we will be able to find the correct start of the OFDM symbol even if the  $M_{opt}$  statistic has multiple peaks due to frequency selectivity. We have to gather the sums of energies from the areas of interest and then we set as the start, the first position of the window with the greatest sum. As a result, we are sure that we have taken all the great peaks of  $M_{opt}$  in mind.

$$\begin{aligned}
fine\_sync\_win[n] &= \sum_{m=0}^{L-1} |M_{opt}[n+m]|^2 \\
n &\in \{\hat{d}_c - N/2, \hat{d}_c + N/2\}
\end{aligned} \tag{2.21}$$

$$\hat{d}_{opt} = \underset{n}{\operatorname{argmax}} \{fine\_sync\_win(n)\}$$

After getting the estimate  $\hat{d}_{opt}$  from any method, we have to go  $L$  symbols backwards in order to also get the CP of the preamble in the packet that we obtain.

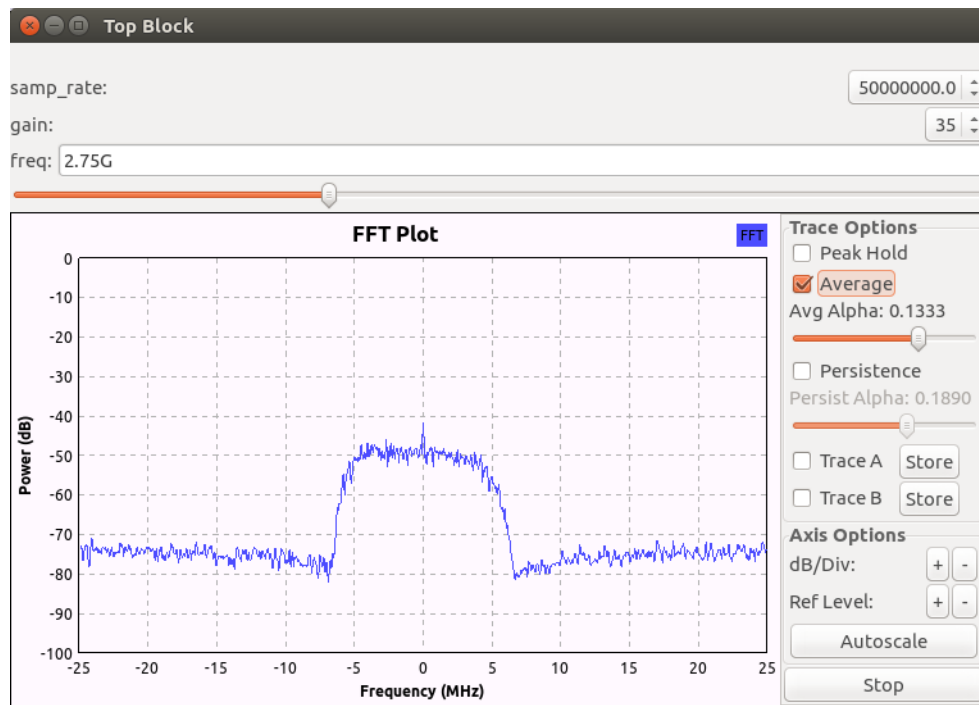
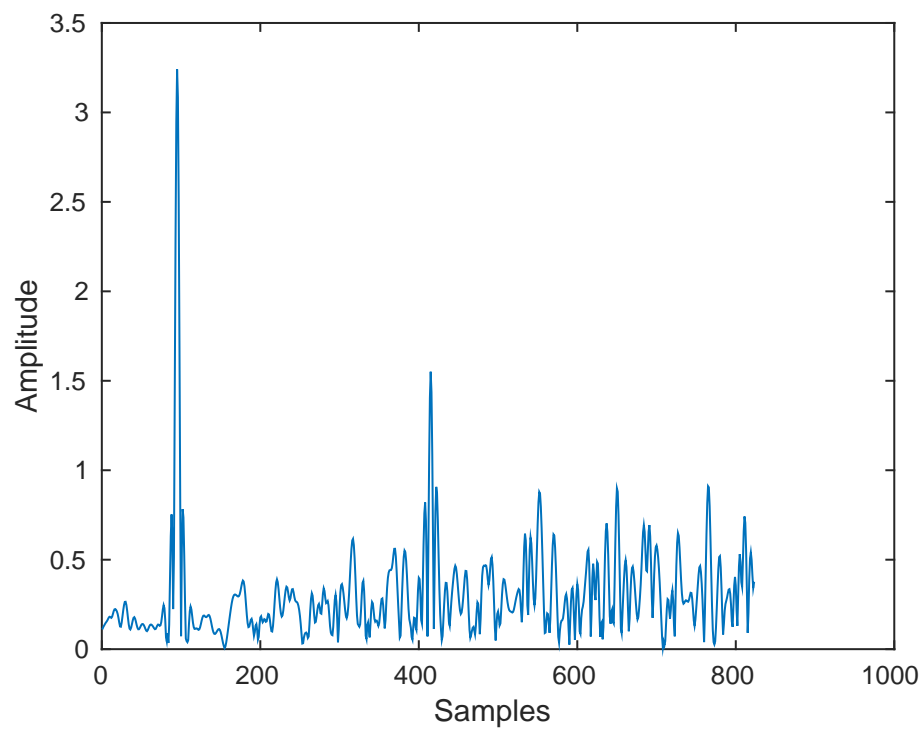


FIGURE 2.7: Spectrum of received signal for a frequency flat channel

FIGURE 2.8: Statistic  $|P_x|$  of an OFDM symbol sent through a frequency flat channel

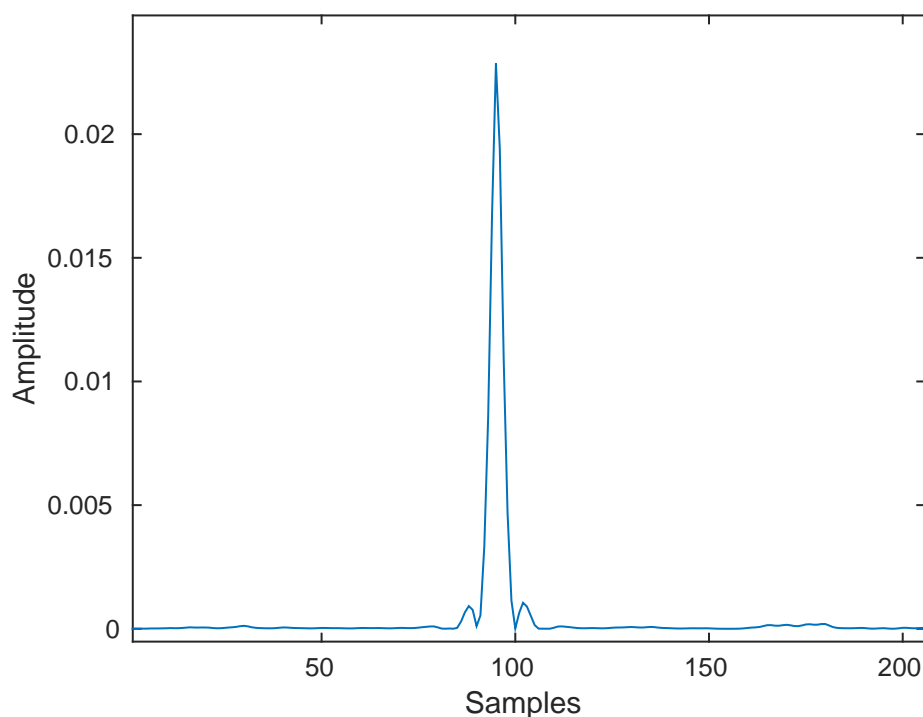


FIGURE 2.9: Statistic  $M_{opt}$  of an OFDM symbol sent through a frequency flat channel

In Figure 2.7, we see the spectrum of an OFDM symbol that is sent with rate equal to 50 Msamples/sec or 10 Msymbols/sec due to our oversampling factor that equals to 5. It is clear that we deal with a flat fading channel (or else channel with short delay spread) as the frequency response is flat, without notches.

In Figure 2.9, we see the synchronization statistic  $M_{opt}$  of an OFDM symbol that is sent through the channel described above. The fact that our channel is flat fading, can be confirmed from the one and only great peak that appears in the figure.

In Figure 2.10, we see the spectrum of an OFDM symbol that is sent with rate equal to 50 Msamples/sec or 10 Msymbols/sec due to our oversampling factor that equals to 5. It is clear that we deal with a frequency selective channel (or else channel with long delay spread) as the frequency response is not flat but presents a notch that degrades the frequencies that correspond to the notch.

In Figure 2.12, we see the synchronization statistic  $M_{opt}$  of an OFDM symbol that is sent through the channel described above. The fact that our channel is frequency selective can be confirmed from the two great peaks that appear in the figure.

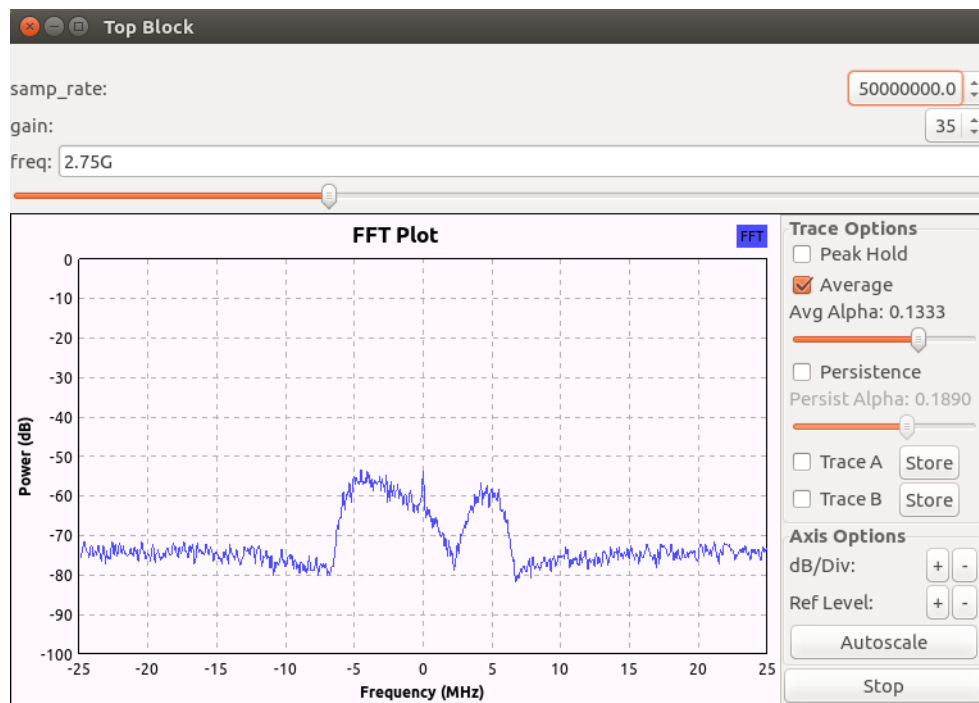
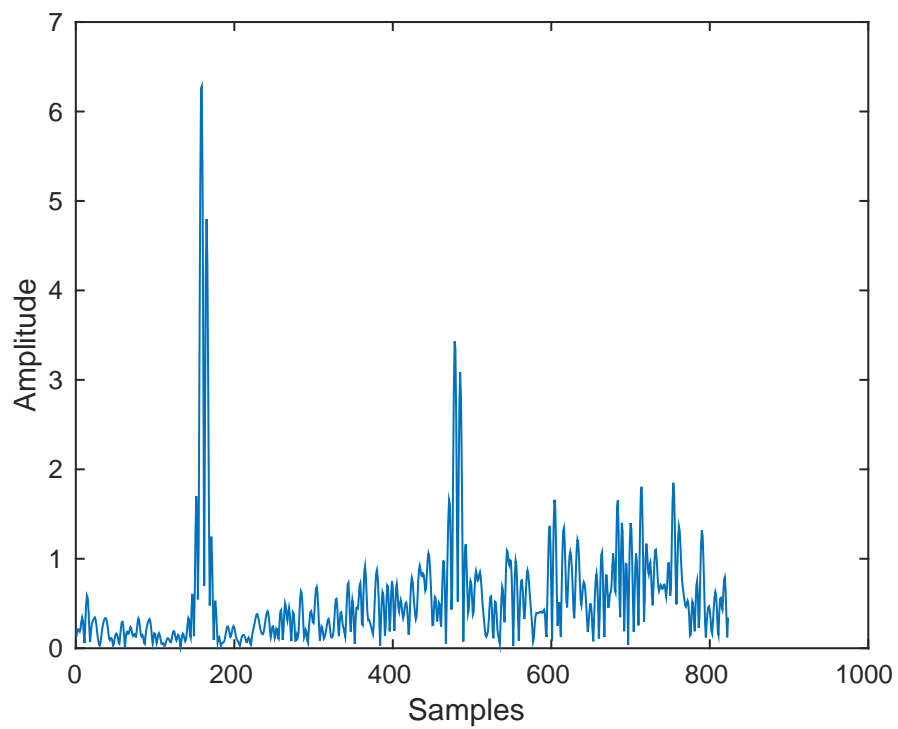


FIGURE 2.10: Spectrum of a received signal for a frequency selective channel

FIGURE 2.11: Statistic  $|P_x|$  of an OFDM symbol sent through a frequency selective channel

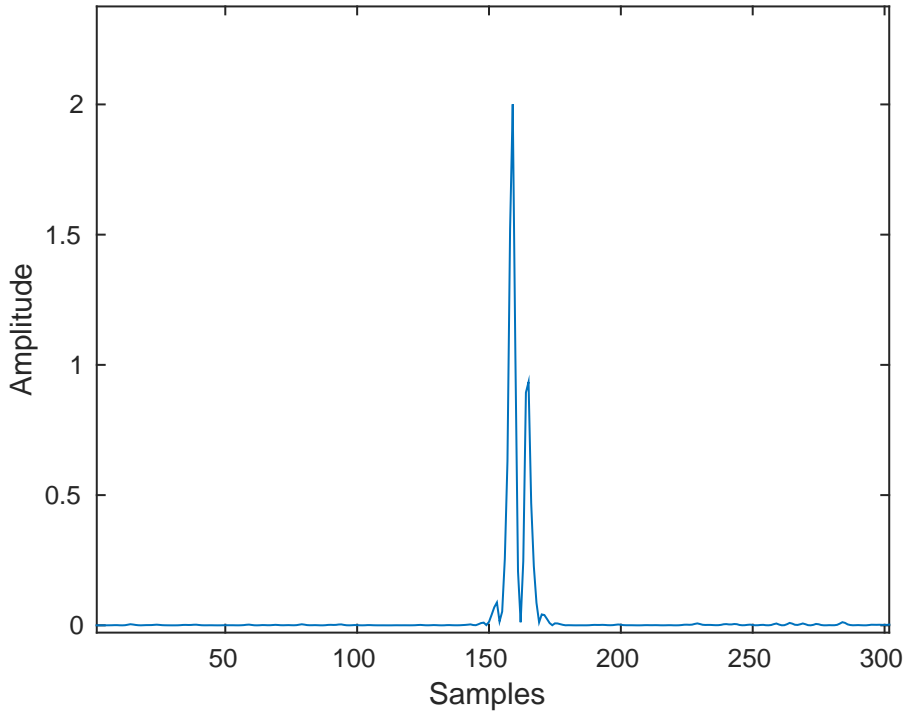


FIGURE 2.12: Statistic  $M_{opt}$  of an OFDM symbol sent through a frequency selective channel

## 2.5 Channel estimation

In our experiments, the channel can be either frequency flat or frequency selective. This depends on the configuration's settings (e.g. transmission rate) or the environment's condition that leads to short or long delay spread of the transmitted signal. Thus, we have to introduce a way to estimate the channel correctly and reliably. In this section, we describe the channel estimation procedure. After time synchronization and CFO correction as mentioned before, the received synchronization symbol (preamble including CP) can be expressed as

$$y'_m = \sum_{l=0}^{L-1} h_l s_{m-l} + w_m, \quad m = 0, \dots, N + L - 1. \quad (2.22)$$

or in vector form

$$\mathbf{y}' = \mathbf{h} * \mathbf{s} + \mathbf{w}. \quad (2.23)$$

Analytically, the above convolution can be written as (assuming the maximum channel's length  $L$  equal to 4)

$$\mathbf{y}' = \begin{bmatrix} h_0 s_0 \\ h_0 s_1 + h_1 s_0 \\ h_0 s_2 + h_1 s_1 + h_2 s_0 \\ h_0 s_3 + h_1 s_2 + h_2 s_1 + h_3 s_0 \\ \vdots \\ h_0 s_{N+L-1} + h_1 s_{N+L-2} + h_2 s_{N+L-3} + h_3 s_{N+L-4} \end{bmatrix} + \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_{N+L-1} \end{bmatrix}$$

or in other form

$$\mathbf{y}' = \begin{bmatrix} s_0 & 0 & 0 & 0 \\ s_1 & s_0 & 0 & 0 \\ s_2 & s_1 & s_0 & 0 \\ s_3 & s_2 & s_1 & s_0 \\ \vdots & \vdots & \vdots & \vdots \\ s_{N+L-1} & s_{N+L-2} & s_{N+L-3} & s_{N+L-4} \end{bmatrix} \begin{bmatrix} h_0 \\ h_1 \\ h_2 \\ h_3 \end{bmatrix} + \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_{N+L-1} \end{bmatrix}$$

Assuming that  $\mathbf{s}$  is known (i.e.  $\mathbf{s}$  is the preamble used previously for synchronization), we have to create a Toeplitz matrix with it, as follows

$$\mathbf{S}_{\text{toep}} = \begin{bmatrix} s_0 & 0 & 0 & 0 \\ s_1 & s_0 & 0 & 0 \\ s_2 & s_1 & s_0 & 0 \\ s_3 & s_2 & s_1 & s_0 \\ \vdots & \vdots & \vdots & \vdots \\ s_{N+L-1} & s_{N+L-2} & s_{N+L-3} & s_{N+L-4} \end{bmatrix}$$

with  $(N + L)$  rows and  $L$  columns. The Toeplitz matrix is used in order to calculate all the channel's values (maximum  $L$  values). In case of a frequency selective channel, some of these values (number equal to the channel's real length) will have comparable magnitude while in case of a frequency flat channel, only one value will be noticeably greater than the other  $L - 1$  values. This method operates also perfectly for flat channels as channels  $h_1, h_2$  and  $h_3$  will have zero or almost zero value, so only the first column of  $\mathbf{y}'$  will have useful information to retrieve the value of  $h_0$  (the other three channel values will return zero or almost zero value as expected from a flat channel).

Then the Least Squares (LS) channel estimate is given by

$$\mathbf{h}_{LS} = (\mathbf{s}_{\text{toep}}^H \mathbf{s}_{\text{toep}})^{-1} \mathbf{s}_{\text{toep}}^H \mathbf{y}', \quad \text{with } \text{size}(\mathbf{h}_{LS}) = L \quad (2.24)$$

where  $\cdot^H$  denotes the Hermitian transpose.

## 2.6 Correction of channel's effect and symbols' estimation

After acquiring our OFDM symbol ( $\mathbf{y}$ ) of length  $N$  from the synchronized and frequency corrected sequence as described previously, we perform a  $N$ -point FFT transform to it. In addition, we perform another  $N$ -point FFT transform to the  $L$  channel values that we calculated in the previous section.

$$\begin{aligned} \mathbf{Y} &= \text{fft}(\mathbf{y}, N) \\ \mathbf{H}_{LS} &= \text{fft}(\mathbf{h}_{LS}, N) \end{aligned}$$

Then, we can easily perform symbol-by-symbol channel correction

$$\mathbf{rx\_symbols} = \frac{\mathbf{H}_{LS}^* \cdot \mathbf{Y}}{|\mathbf{H}_{LS}|.^2}$$

with  $(\cdot^*)$  the element-by-element multiplication and  $(.^2)$  the element-by-element exponentiation to 2 (the division is also element-wise).

The final step is to have the symbol decisions on the **rx.symbols** vector's elements. This task is performed easily by looking at the sign of the real and imaginary part of each symbol using the following rule

- if  $\text{real}(\mathbf{rx\_symbols}[i]) > 0$  ,  $\text{imag}(\mathbf{rx\_symbols}[i]) > 0$  then  $\text{est\_symbol}(i) = 1+j$
- if  $\text{real}(\mathbf{rx\_symbols}[i]) > 0$  ,  $\text{imag}(\mathbf{rx\_symbols}[i]) < 0$  then  $\text{est\_symbol}(i) = 1-j$
- if  $\text{real}(\mathbf{rx\_symbols}[i]) < 0$  ,  $\text{imag}(\mathbf{rx\_symbols}[i]) > 0$  then  $\text{est\_symbol}(i) = -1+j$
- if  $\text{real}(\mathbf{rx\_symbols}[i]) < 0$  ,  $\text{imag}(\mathbf{rx\_symbols}[i]) < 0$  then  $\text{est\_symbol}(i) = -1-j$

We can see a typical scatterplot of a received OFDM symbol that has come from 4-QAM modulated data symbols in Figure [2.13](#).



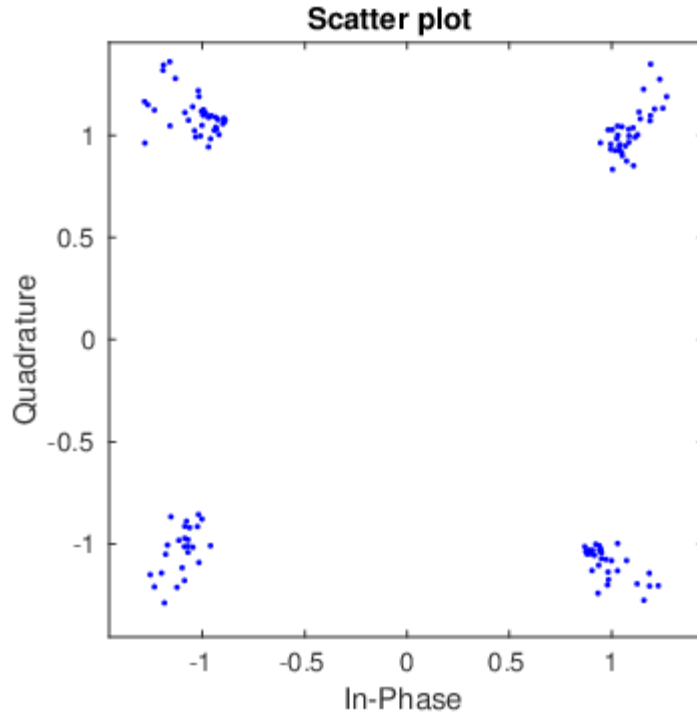


FIGURE 2.13: Scatterplot of a received OFDM symbol using 4-QAM constellation

## 2.7 Implementation details

### 2.7.1 Specifications

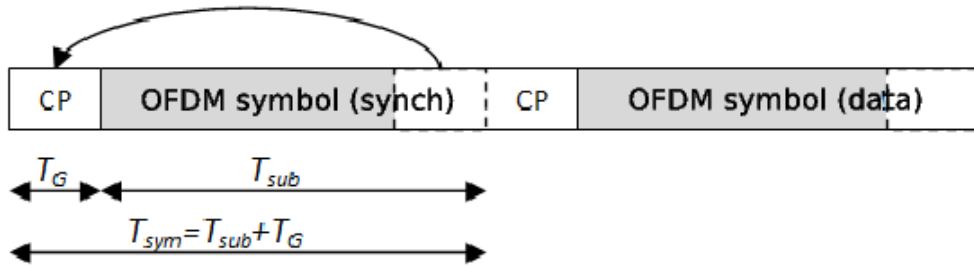


FIGURE 2.14: Structure of a packet

In our experiments, we consider a preamble (synchronization symbol) followed by an OFDM symbol containing useful data, as a packet. In that way, the synchronization algorithm is run every time a single packet arrives, in order to check its efficacy and precision. We consider a gap of zeros between consecutive packets so as to make the detection of each packet feasible. Each OFDM symbol consists of 128 subcarriers ( $N=128$ ), where each of them carries a 4-QAM symbol generated randomly. Only the DC subcarrier

is left empty because hardware non-linearities distort the carried information seriously. Each packet is upsampled with a factor equal to 5 ( $over = 5$ ) and a cyclic prefix of length  $L$  equal to 4 symbols (or  $L * over$  samples) is added to the front, as described in the first section of this chapter. So each packet consists of  $(2N + 2L) * over$  samples. Afterwards, every upsampled packet is convolved with a Squared Root Raised Cosine (SRRC) filter of our preference, in order to obtain some advantages described in subsection 2.7.2, and then the resulting signal is sent from the transmitter. Finally, after detecting the incoming packet using the procedure of subsection 2.7.4, the receiver performs matched filtering with an SRRC pulse, identical to the one that the transmitter has used before transmission and then the synchronization method starts.

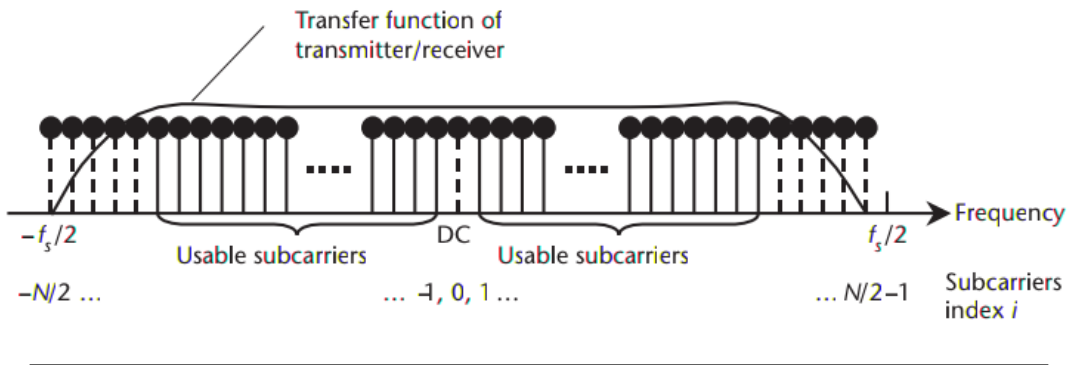


FIGURE 2.15: Illustration of the subcarriers used

### 2.7.2 Square Root Raised Cosine (SRRC) filter

In signal processing, a Root Raised Cosine filter (RRC), sometimes known as Square Root Raised Cosine filter (SRRC), is frequently used as the transmit and receive filter in a digital communication system to perform matched filtering. The combined response of two such filters, as the pulse shaping filter is split between the transmitter and the receiver, is that of the raised-cosine filter, which does not suffer from intersymbol interference (ISI).

As we see in Figure 2.16, the size of ripples and the filter's passband depend on the parameter  $\beta$ , while in our implementation  $\beta$  equals to 0.3.

### 2.7.3 Channel types

At the previous section, we had to modify the propagation environment suitably in order to change its the delay spread. In telecommunications, the delay spread is a measure of the multipath richness of a communications channel. In general, it can be interpreted as the difference between the time of arrival of the earliest significant multipath component

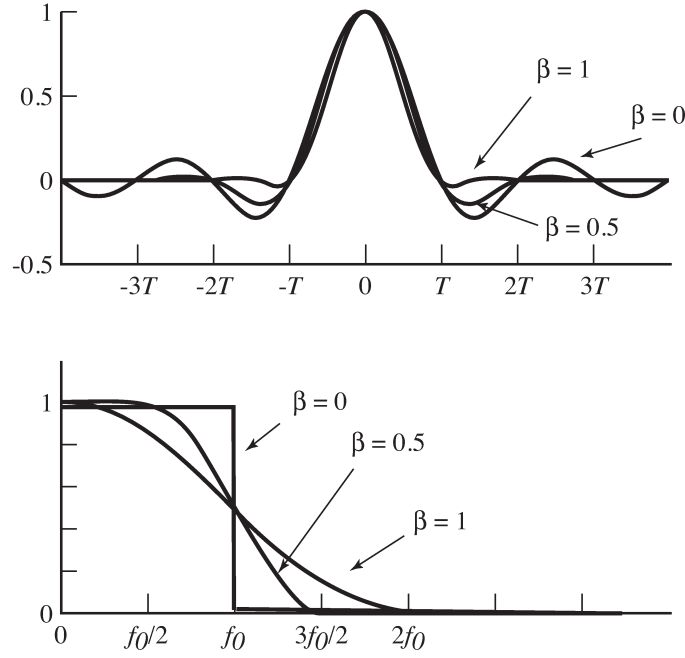


FIGURE 2.16: SRRC pulse shape in the time and frequency domains

(typically the line-of-sight component) and the time of arrival of the latest multipath components. The delay spread is mostly used in the characterization of wireless channels, but it also applies to any other multipath channel (e.g. multipath in optical fibers).

The wireless channels are distinguished in those of short delay spread and those of long delay spread. More specifically, if the delay spread  $T_m(t)$  satisfies the relation

$$T_m(t) \ll \frac{1}{B} \simeq T,$$

then the channel has short delay spread and the measure of the frequency response of the discrete channel is flat ( $H(F) = \mathcal{F}\{h\} = h$ ). This type of fading, which affects only the amplitude and phase of the input symbols at the output, without introducing intersymbol interference, is called **frequency flat fading**.

On the other hand, if the delay spread  $T_m(t)$  satisfies the relation

$$T_m(t) \geq \frac{1}{B} \simeq T,$$

then the channel has long delay spread and the frequency response of the discrete channel is  $H(F) = \mathcal{F}\{\dots, h_{-1}, h_0, h_1, \dots\}$ , the magnitude of which is not flat. This type of fading, that introduces intersymbol interference, is called **frequency selective fading**.

At the experiments that we were in need of a frequency flat channel, the space between the transmitter and the receiver was empty in order to accomplish a short delay spread

channel. On the other side, when we wanted a frequency selective channel, the space between the transmitter and the receiver was full of obstacles in order to accomplish a long delay spread. The great transmission rate also helps us to achieve the appearance of a frequency selective channel as bigger rate leads to lower symbol period. Last but not least, we had to use 8-bit samples in our implementation in cases of transmission rate greater than 25 Msamples/sec, so as to stay within the hardware's potential. Lower transmission rates could be feasible with the more ordinary 16-bit samples.

#### 2.7.4 Initial detection of packets

The first and ceaseless job of a receiver is to be able to detect any incoming transmissions. This procedure, also known as *packet synchronization*, has to overlook any possible peaks of noise and compute a rough estimate for the beginning of any incoming packet. In order to accomplish that, we use a *Double Sliding Window* (DSW) algorithm for packet detection. The DSW algorithm uses two consecutive sliding windows to calculate the sum of the energies of the samples in each window. At each window movement, the ratio of these two sums ( $SNR_{est}$ ) is calculated and if it exceeds a designated threshold ( $SNR_{thresh}$ ), a counter is raised by one ( $SNR_{alert}++$ ). If this counter reaches a specific limit, then we know that we have found a packet and after estimating a rough start, its synchronization begins as described previously. On the other hand, each time the ratio of the two sums is lower than the threshold, the counter becomes zero ( $SNR_{alert}=0$ ).

The length of each window (signal\_win is the right window and noise\_win is the left window) is equal to  $\frac{length(packet)}{8}$ , the threshold  $SNR_{thresh}$  is equal to 7 and the limit over which we stop raising the SNR\_alert counter is equal to  $\frac{length(window)}{2}$ . All the above values have been calculated experimentally and fulfill their purpose reliably.

$$\begin{aligned}
 signal\_win[n] &= \sum_{m=0}^{W_{len}-1} |r[n + W_{len} + m]|^2, \\
 noise\_win[n] &= \sum_{m=0}^{W_{len}-1} |r[n + m]|^2, \\
 SNR_{est} &= \frac{signal\_win[n]}{noise\_win[n]}.
 \end{aligned}$$

The ratio SNR\_est becomes large when the signal\_win's input is part of an incoming packet. The above process is illustrated in the following pictures for further understanding.

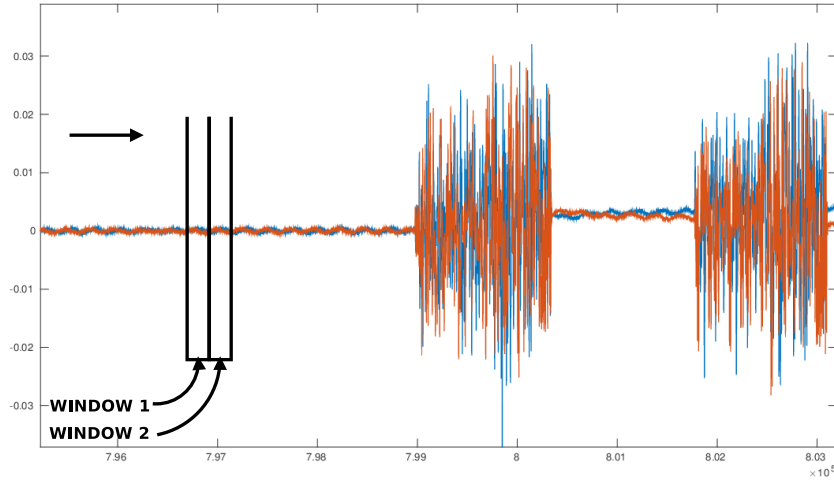


FIGURE 2.17: The Double-Sliding-Window set up initially

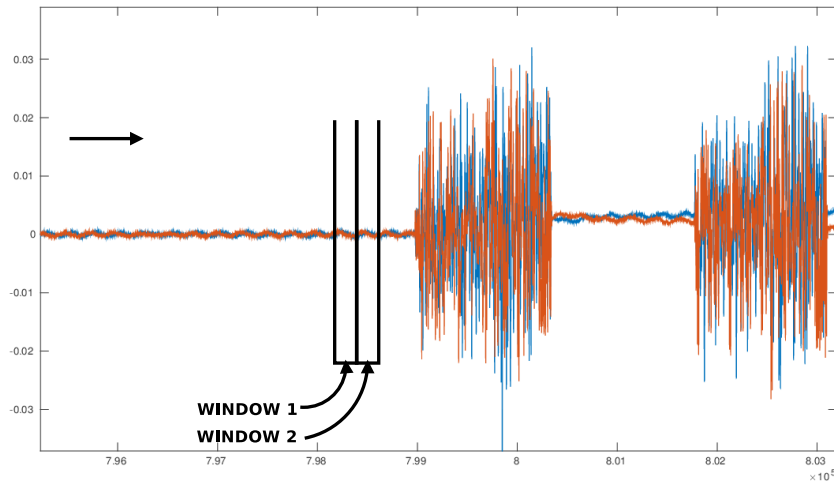


FIGURE 2.18: The DSW sliding towards a received packet

## 2.8 Chapter summary

Several system models and details were introduced in this chapter to become familiar with OFDM and understand the primary reasons of its popularity in modern telecommunications. After intensive and careful testing, a series of algorithms for time and frequency synchronization, channel estimation and correction and finally the symbols' retrieval were chosen for use in our transmitter-receiver design. The implementation's precision showed the efficiency and accuracy of the algorithms, dealing with frequency

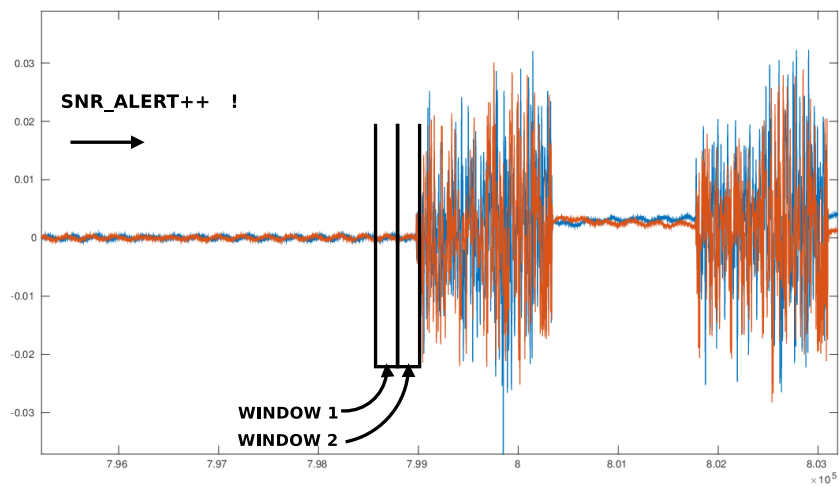


FIGURE 2.19: The signal window(window 2) gets a part of the packet, SNR\_alert is raised

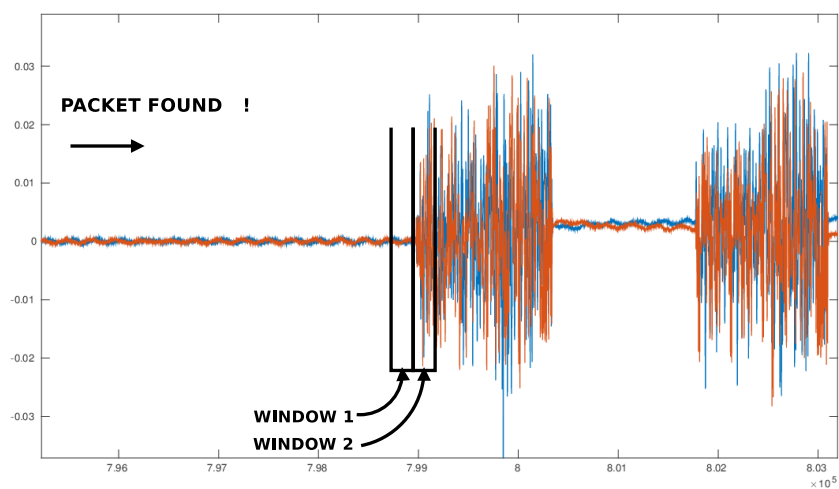


FIGURE 2.20: As the DSW slides to the right, the SNR\_alert reaches its limit. Packet found!

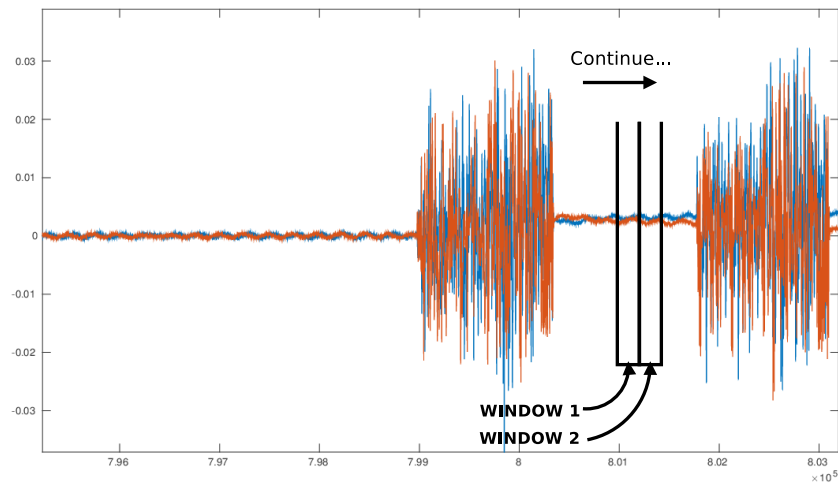


FIGURE 2.21: The DSW gets out of the packet to continue for further detections

flat and frequency selective channels, delivering the transmitted symbols from the transmitter to the receiver correctly. In the following chapter, this implementation is going to be modified in order to serve another purpose; the development of a time-slotted protocol used by two users that want to transmit their data using this OFDM structure.





## Chapter 3

# A time slotted protocol with a primary transmitter and a secondary cognitive radio

### 3.1 Introduction

Cognitive radio is an emerging technology promoting the concept of dynamic spectrum access, contrary to the existing spectrum allocation model of fixed spectrum access. Under the conventional spectrum access model, license holders receive exclusive rights to spectrum usage, while no other parties may transmit on these spectra under any circumstances. However, most technologies are not active 100% of the time, thus spectral opportunity is wasted if the license holders do not transmit on the licensed spectrum. This outdated model is often considered to be the major problem of spectrum scarcity, where new wireless technologies that can provide better service and better spectral utilization cannot be implemented as they do not have access to sufficient spectrum.

The concept of dynamic spectrum access and cognitive radio technology aims to address this issue by permitting secondary access to spectrum licensed to primary users. Based on the proposed concept, non-licensed, secondary users (SU) can access spectral bands owned by licensed, primary users (PU) as long as interference to PU activity is maintained minimal. Therefore, the SU must possess cognitive capabilities to observe the surrounding environment, learn from its observations and adapt its operation characteristics to restrict interference to the PU and maintain the efficiency of its own network. Cognitive radio can provide multiple benefits to license holders and non-licensed users alike. For license holders, the opportunity to 'sub-lease' the spectrum provides financial benefits to cover the cost associated with spectrum auction and obtaining the initial

license. On the other hand, non-licensed users have the opportunity to establish a new wireless service and reap its associated profits, while improving the overall spectral utilization efficiency [9].

The introduction of a new technological concept also raises countless questions and problems. Cognitive radio is envisioned to operate on different PU frequencies, therefore existing radio hardware designed for specific spectral bands cannot be used. Innovative software algorithms have to be proposed to quickly and efficiently support all cognitive capabilities; the computation load will be far more complex compared to conventional wireless technologies. The communication protocols within a cognitive radio network is also a significant challenge, as SU may not have access to dedicated control channels handling the basic operations of a communication network. Spectrum allocation and management plays a crucial role in enabling the coexistence of multiple SU networks, promoting fair use and security of individual networks. Overall, cognitive radio has raised numerous opportunities for research, innovation and breakthrough, leading to a birth of a new industry.

## 3.2 Protocol details

The protocol defines time slots of 100 msec each, with the start of each slot known to the users. In order to share common time knowledge, the users have to be time synchronized somehow. Their synchronization is accomplished by using a special cable that will be described in the next section. The users consist of one Primary User (PU) and one Secondary User (SU) while a receiver is collecting the packets sent from both of them. In order to make each user's packets distinguishable to the receiver, 2 out of 128 OFDM subcarriers of each user no longer carry useful data, but a tag known to the receiver.

The PU's structure is not the same as the transmitter's from Chapter 2 but slightly modified, in order to send packets at the beginning of each time-slot with a desired probability. On the other hand, the structure of the SU has been deeply altered as he now has to perform spectrum sensing at the beginning of each time slot in order to detect any PU's transmission. The spectrum sensing is performed by an energy detector whose operation is identical with the package detection process described thoroughly in 2.7.4. Both of these detectors operate energy-wise and while their final targets are different, the energy detection stage is common, allowing us to re-use this module.

The spectrum sensing lasts for 20 msec at most. If the SU detects the PU transmitting in this slot, he doesn't transmit and waits for the arrival of the next slot. On the

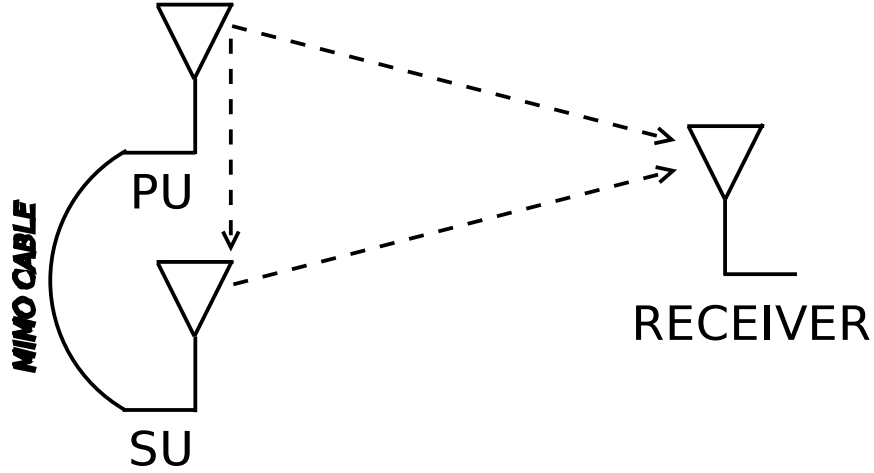


FIGURE 3.1: Diagram of the USRPs' set up

contrary, if the PU is not detected after 20 msecs, the SU transmits his packets and the slot gets occupied successfully. Last but not least, the last 5 msecs of each slot remain always empty in order to avoid interference between the transmissions of different users.

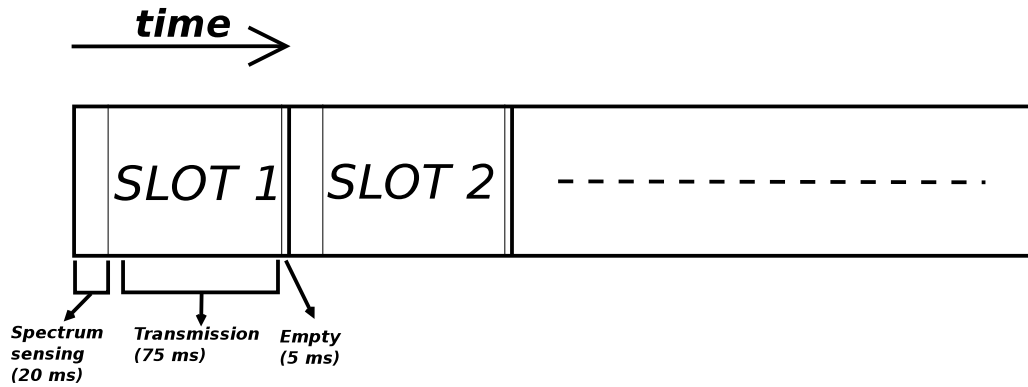


FIGURE 3.2: Diagram of the protocols' time-slots

### 3.3 Synchronization with the Ettus MIMO cable

Some applications like our protocol, require synchronization across multiple USRP devices. Ettus Research provides several convenient solutions for synchronization. For example, two USRP N200s can be synchronized using a MIMO cable. It is also possible to synchronize more than two units using the Ettus Research OctoClock. Optional GPS-disciplined oscillators provide the capability to synchronize devices to the GPS standard over a large geographic area. The first option was chosen for our application as it is an adequate and easy-to-use solution for linking two USRP devices.

The MIMO expansion cable is used to link a pair of USRP N2xx systems together. The sample clocks are then synchronized and aligned while we also get a common time reference that each user can use to know the start of each time-slot.



---

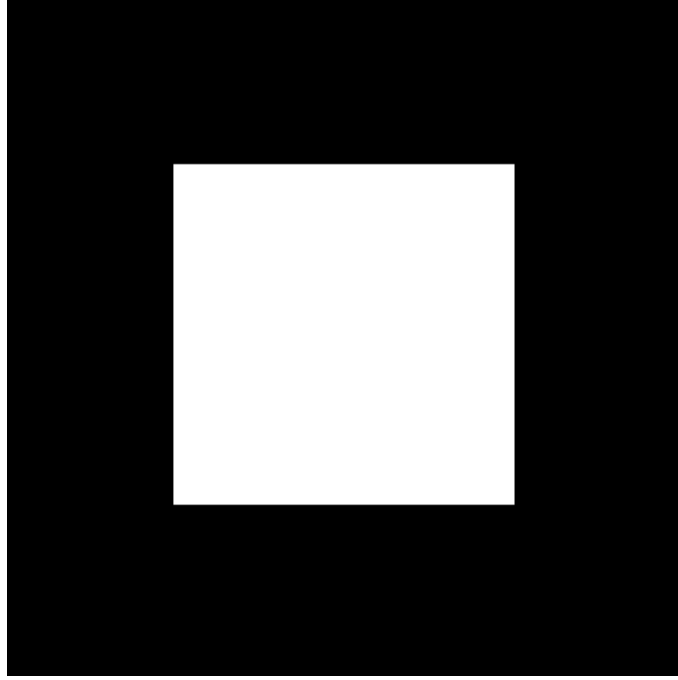
FIGURE 3.3: MIMO cable used for synchronization

### 3.4 Image transmission from the users

While transmitting randomly generated data to test the implementation of the time-slotted protocol is adequate, it would be more practical to transmit specific data that could be printed from the receiver, such as an image. So we assign the same image to both the PU and the SU for transmission, in order to have the same size. The image chosen for transmission is a bitmap image (400x400 pixels) with binary values that can be easily assigned to 4-QAM modulation symbols. The packet's structure remain the same as described before.

**The following settings have been chosen for this experiment:**

- Each packet has a length of 2760 samples (including the following gap between packets)
- The transmission rate is 1 Msamples/sec
- The size of the receive buffer is 100000 samples
- The first packet of each user's transmission is discarded because of data distortion due to USRP's imperfections




---

FIGURE 3.4: Image of 400x400 pixels selected for transmission

- Each user sends 5 packets per time-slot

Each time-slot has 75 msec of useful transmission time or  $(75 \text{ msec}) * (1 \text{ Msamples/sec}) = 75000 \text{ samples/slot}$ . This equals to

$$\text{max\_packets} = \lfloor \frac{75000}{2760} \rfloor = 27 \text{ packets per slot}$$

The receive buffer is a complex vector that we use to store incoming samples at the receiver when we desire to perform reception. Until now, if a packet got cut at the end of the buffer after reception, it was discarded. This is not acceptable now, because dropping packets with useful data would lead to partial image loss. So, each user sends only 5 packets per slot, in order to get all the incoming packets in our receive buffer successfully. Another method is presented later in section 3.4.2, in order to take full advantage of the time-slot and not leave it partially empty, that is undesired.

### 3.4.1 Image reception at the receiver

We have run four experiments based on the settings described in 3.4. At these experiments, the PU slot occupancy probability ranges from 0.9 to 0.6. In the beginning of both experiments, we have the resetting of the common time reference to 0.0 by the PU. After one second (1.0), the transmission of the images starts from both the transmitters in their respective time-slots.

The duration of transmission (in seconds) for the images sent from the PU and the SU are shown at table 3.1.

USER	PU_SOP=0.9	PU_SOP=0.8	PU_SOP=0.7	PU_SOP=0.6
PU	35.5	41.4	44.1	52.4
SU	63.9	63.9	63.9	63.9

TABLE 3.1: Duration of transmission without diversity and 5 packets per slot

where PU\_SOP is the PU Slot Occupancy Probability.

As we can see, the SU's transmissions end at the same time after the PU, as the PU has greater priority. The increase in the PU slot occupancy probability allows the PU to end his transmission faster while letting less gaps for the SU to fill and vice versa.

Afterwards, the same experiments were conducted again but with time diversity of order 2. Specifically, a new packet is transmitted every two transmitted packets instead of one. In that way, the receiver keeps the estimated symbols that come from each packet, and after a pair has arrived, the symbols are added and the sign-wise symbol retrieval is conducted with considerably increased precision. This decrease in bit error rate will be examined later in section 3.5. On the other hand, the rate of the data transmission is reduced by half as a tradeoff to the increased reliability. As a result, the durations corresponding to the ones at the table 3.1, are doubled in this case for both users.

In Figures 3.5 to 3.8, we can see the screenshots of the received images from both users right after the PU completion. As expected, the increase in the PU slot occupancy probability leads to less SU transmissions while the PU is still active.

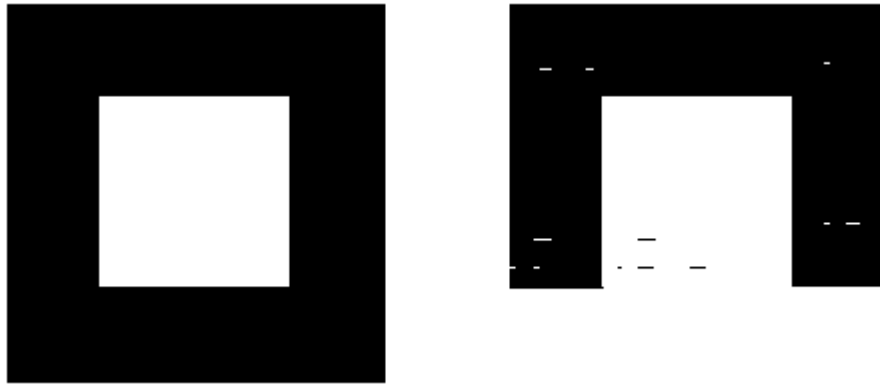


FIGURE 3.5: Received images for PU slot occupancy probability equal to 0.6.  
Left: PU, Right: SU

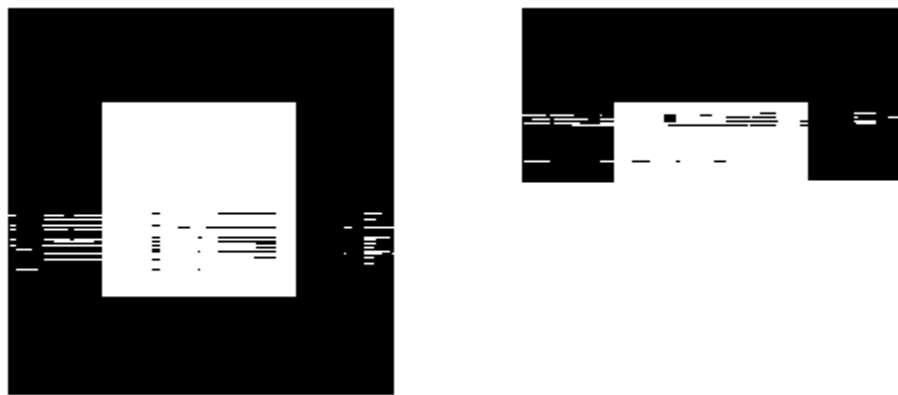


FIGURE 3.6: Received images for PU slot occupancy probability equal to 0.7.  
Left: PU, Right: SU



FIGURE 3.7: Received images for PU slot occupancy probability equal to 0.8.  
Left: PU, Right: SU



FIGURE 3.8: Received images for PU slot occupancy probability equal to 0.9.  
Left: PU, Right: SU

### 3.4.2 Full exploitation of the time-slots with a cyclic buffer

The receive buffer is a complex vector that we use to store incoming samples at the receiver when we desire to perform reception. As previously mentioned, if a packet got cut at the end of this buffer after reception, then we discarded it. One solution to this issue would be to create a cyclic type of buffer. While performing the packet detection, when we identify the existence of a possible packet whose end is beyond the end of the buffer, we transfer the corresponding samples to a new buffer and then fill the rest of it with the following samples. Before this transfer, the first positions of the new buffer (equal to the size of the two windows-DSW) are filled with pseudonoise so that we can apply the two windows used for packet detection as described in 2.7.4. Then, the packet detection process continues in order to detect new packets and repeat the above process when needed. We can observe a detailed illustration of this process in figures 3.9 to 3.12.

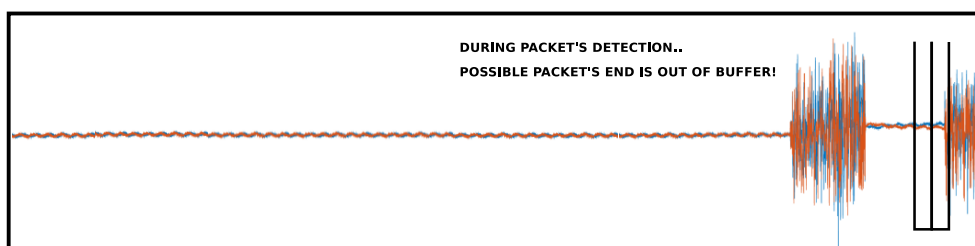


FIGURE 3.9: Possible packet out of buffer during detection



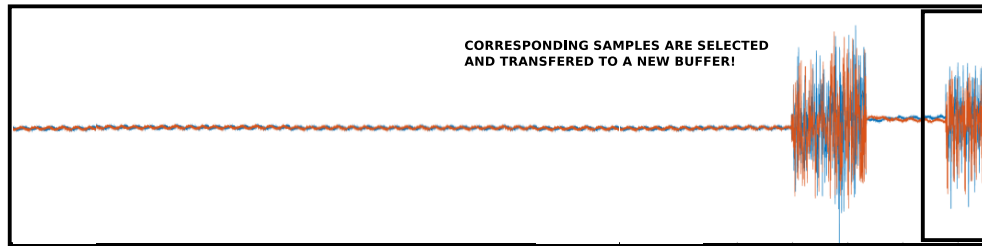


FIGURE 3.10: Selection of samples to transfer

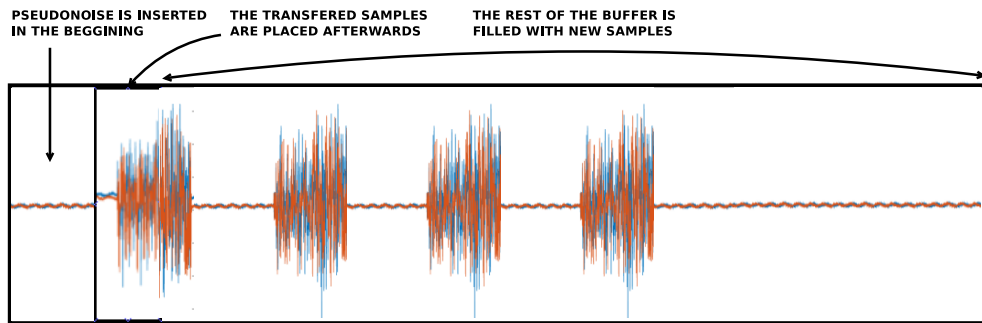


FIGURE 3.11: Transfer of samples to new buffer and filling with new samples

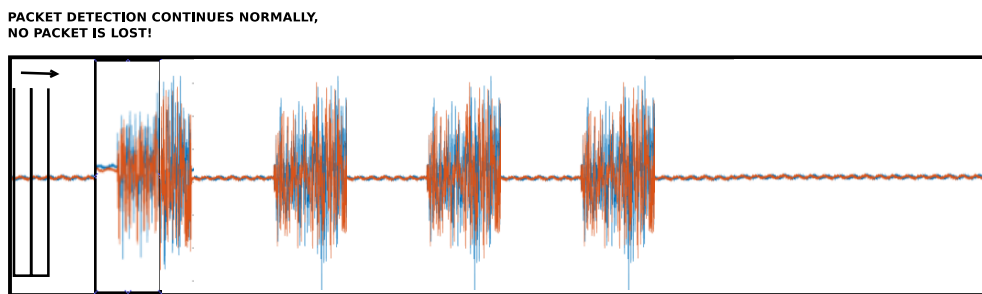


FIGURE 3.12: Buffer is ready for packet detection

USER	PU_SOP=0.9	PU_SOP=0.8	PU_SOP=0.7	PU_SOP=0.6
PU	10.4	11.0	13.3	13.4
SU	18.3	18.3	18.3	18.3

TABLE 3.2: Duration of transmission without diversity and 15 packets per slot

At the tables 3.2 and 3.3, we can see the time needed to receive the same picture that was sent in 3.4.1. Obviously, as we fill the time slot with more packets, the slot utilization is raised and the total transmission duration becomes less for both users.

USER	PU_SOP=0.9	PU_SOP=0.8	PU_SOP=0.7	PU_SOP=0.6
PU	6.0	7.4	8.1	9.6
SU	10.7	10.7	10.7	10.7

TABLE 3.3: Duration of transmission without diversity and 25 packets per slot

### 3.5 Error analysis in image transfer from various distances

In this section, we will have a bit error rate analysis of a user of the protocol described in this chapter. Moreover, many illustrations of the received images from different distances will make their comparison interesting. The figures 3.13 to 3.16 contain the received images that were reconstructed at the receiver. At the left of each figure, we have the image that resulted from a transmission with no diversity while at the right we have the image that resulted from a transmission with diversity of order 2 (due to retransmission of the same packets) in order to increase the reliability of the result.

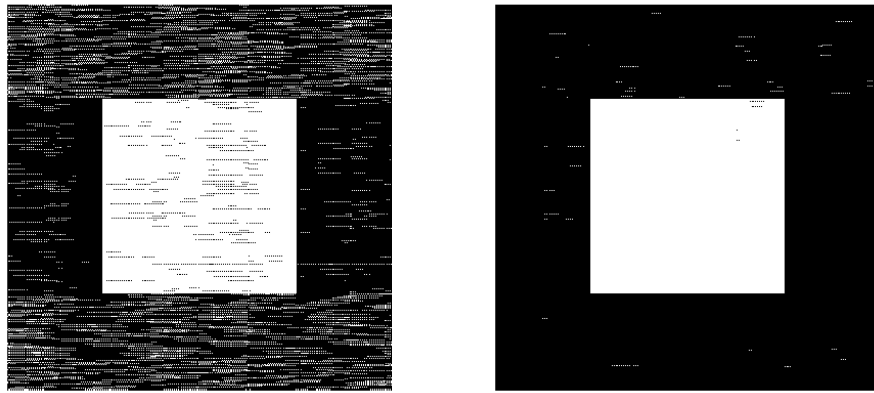


FIGURE 3.13: Received images without/with diversity of order 2 from a distance of 2 metres

DIVERSITY	2 METRES	4 METRES	6 METRES	8 METRES
NO	1801	15754	38584	48569
YES	0	244	28246	42760

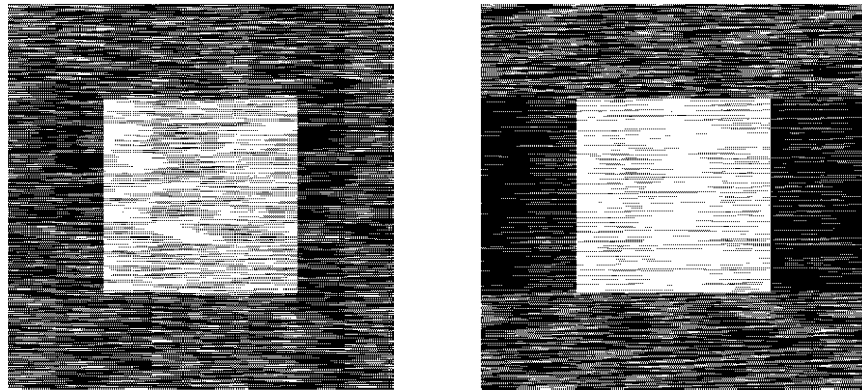
TABLE 3.4: Number of wrong received bits out of  $400 \times 400 = 160000$  image bits

As we can see from Figures 3.13 to 3.17 and Table 3.4, there is a significant increase in the reliability of the image transfer in case of diversity of order 2. Specifically, we observe that the improvement in short distance transmissions is far greater than the improvement from a certain distance and further.




---

FIGURE 3.14: Received images without/with diversity of order 2 from a distance of 4 metres




---

FIGURE 3.15: Received images without/with diversity of order 2 from a distance of 6 metres

### 3.6 Chapter summary

Our time-slotted application of the modified transmitter-receiver system that was developed in Chapter 2, provided us with the results that were expected intuitively. The transmission of a specific image from both users, gave us the opportunity to have several graphic representations of the received data. Finally, the observation of these representations showed the importance of the existence of cognitive radios in modern telecommunication systems. A great amount of underused bandwidth can be assigned to users in a more sophisticated way in order to serve the great spectral needs of the increasing number of network devices nowadays.

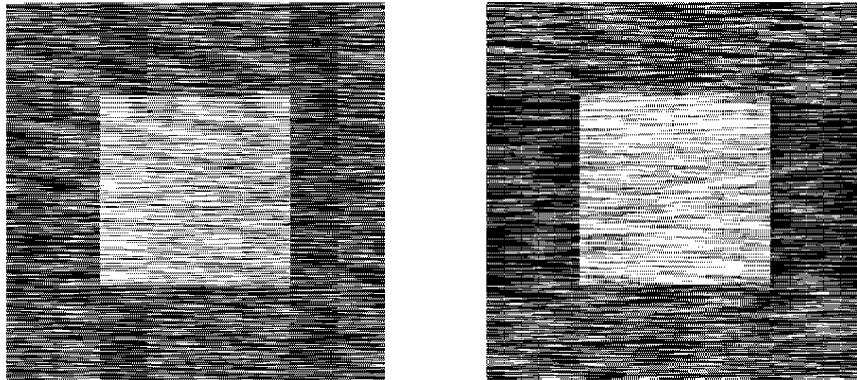


FIGURE 3.16: Received images without/with diversity of order 2 from a distance of 8 metres

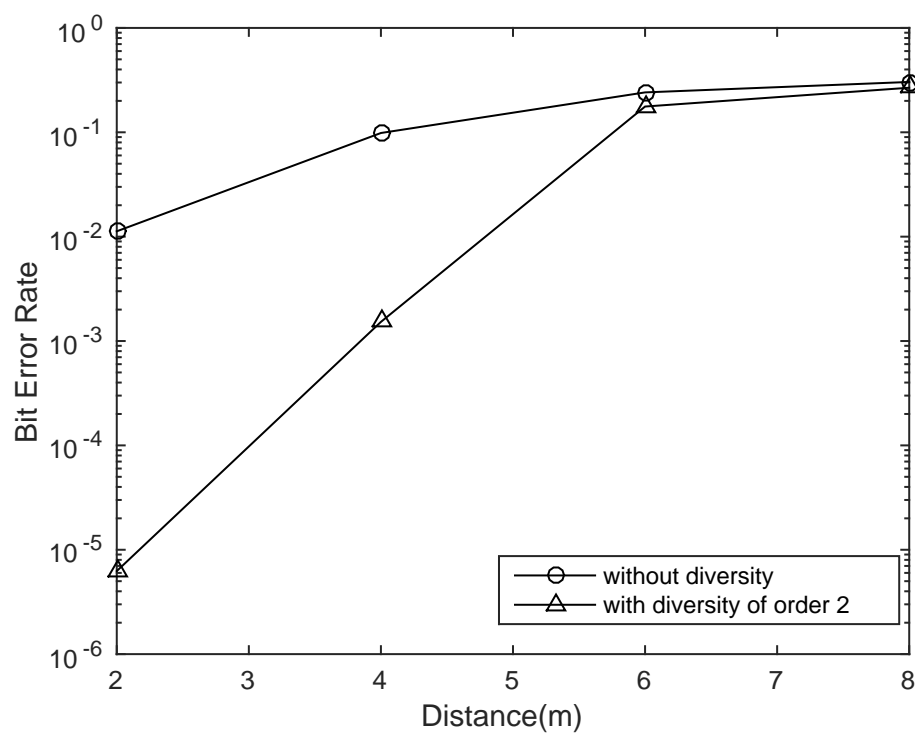


FIGURE 3.17: Bit Error Rate for several distances of transmission with/without diversity of order 2

## Chapter 4

# Future work

### 4.1 Future work

The future work considering the work that was done in this thesis, consists of several improvements and further evolution. For example, a cyclic redundancy check, which is an error-detecting code, could be added in the design in order to detect transmission errors and correct it. In that way, each packet would be checked and the reliability of the link would be enhanced. Another way of achieving the diversity we achieved would be by means of frequency diversity. In that case, we would have to increase the transmission rate and have a more complex environment in order to have a frequency selective channel (i.e. with long delay spread). Then, we would use subcarriers that are affected by frequency selective fading to carry repeated symbols to increase reliability.

In the area of cognitive radios, several ways of spectral sensing could be examined thoroughly so as to compare them and reach to useful conclusions regarding their efficiency and accuracy. Finally, a way of having the USRPs synchronized to a common time reference with precision, without the use of MIMO cables, would be very useful in time-slotted applications with many users.



# Bibliography

- [1] Ye Li and Gordon L. Stuber. Orthogonal frequency division multiplexing for wireless communications. pages 113–183, 2006.
- [2] Athanasios P. Liavas. Lecture notes on orthogonal frequency division multiplexing. *Technical University Of Crete*, pages 1–7, November 2013.
- [3] I. Kardaras. Software-defined radio implementation of an ofdm link. *Master of Science, Technical University Of Crete*, pages 1–64, August 2010.
- [4] T. Schmidl and D. C. Cox. Robust frequency and timing synchronization for ofdm. *IEEE Trans. Commun.*, 45(12):1613–1621, December 1997.
- [5] B. Park, H. Cheon, C. Kang, and D. Hong. A novel timing estimation method for ofdm systems. *IEEE Commun. Lett.*, 7(5):239–241, May 2003.
- [6] H. Minn, V. K. Bhargava, and K. B. Letaief. A robust timing and frequency synchronization for ofdm systems. *IEEE Trans. Wireless Commun.*, 2(4):822–839, July 2003.
- [7] K. Shi and E. Serpedin. Coarse frame and carrier synchronization of ofdm systems: a new metric and comparison. *IEEE Trans. Wireless Commun.*, 3(4):1271–1284, July 2004.
- [8] Adegbenga B. Awoseyila, Christos Kasparis, and Barry G. Evans. Improved preamble-aided timing estimation for ofdm systems. *IEEE Commun. Lett.*, 12(11): 825–827, November 2008.
- [9] Kevin Chang. Spectrum sensing, detection and optimisation in cognitive radio for non-stationary primary user signals. *PhD Thesis*, pages 1–12, February 2012.