
5G Techniques

- Proof-of-concept Testbed -

Master Thesis
Manuel Sainz

Aalborg University
Department of Electronic Systems
Fredrik Bajers Vej 7B
DK-9220 Aalborg

Copyright © Aalborg University 2015

The following thesis work has been written using L^AT_EX. On the one hand, the design diagrams presented in the figures were designed in LabVIEW, which is the software defined radio adopted for our implementation. On the other hand, some of the results were calculated and plotted in MATLAB.



Department of Electronic Systems

Fredrik Bajers Vej 7

DK-9220 Aalborg Ø

<http://es.aau.dk>

AALBORG UNIVERSITY

STUDENT REPORT

Title:

5G Techniques

Proof-of-concept Testbed

Theme:

Wireless Communications Systems

Project Period:

Fall Semester 2014

Project Group:

Group 1201

WCS 10

Participant(s):

Manuel Sainz

Supervisor(s):

Troels Bundgaard Sørensen

Joan M. Gené Bernaus

Copies: 2

Page Numbers: 174

Date of Completion:

April 23, 2015

Abstract:

The exponential increase of the data traffic in the next years and the limitations of the wireless spectrum shared by multi-users have challenged the design of the coming 5G radio access technology. In the last decade, multiple-input multiple-output (MIMO) systems have become one of the key technologies offering a higher spectral efficiency. Nevertheless, the advantages provided by MIMO systems come to the detriment of an increase in the cost to deploy multiple antennas and also in the receiver complexity. Thus, a pre-equalization stage called precoding is applied at the transmitter in order to improve the coverage and range of the network in fading environments. This thesis investigates performance aspects of such MIMO channel matrix precoding techniques. On the one hand, a digital communication system testbed based on USRP radio hardware is implemented by means of the software defined radio (SDR) LabVIEW. On the other hand, the effective potential of precoding techniques in boosting the network throughput is addressed in a real world scenario testbed. Our results reveal that the spatial diversity provided by multiple antennas is exploited by precoding and thus achieving higher channel capacity.

The content of this report is freely available, but publication (with reference) may only be pursued due to agreement with the author.

Contents

Preface	ix
Acronyms	xi
1 Introduction	1
1.1 Motivation	1
1.2 Background	2
1.3 Objectives	7
1.4 Guidelines	8
2 Digital Communications Systems	11
2.1 Elements of a Digital Communication System	11
2.1.1 Transmitter	14
2.1.2 Channel	17
2.1.3 Receiver	18
2.2 System Performance Measures	19
2.2.1 Signal-to-Noise Ratio (SNR)	19
2.2.2 Probability of Error	20
2.2.3 Bit Error Rate (BER)	22
2.2.4 Probability Distribution Function (PDF) and Cumulative Distribution Function (CDF)	22
2.3 Source and Channel Coding	24
2.4 Block and Convolutional Codes	25
2.4.1 Linear Block Codes	25
2.4.2 Convolutional Codes	26
2.5 Modulation Schemes	27
2.5.1 Quaternary Phase-shift keying (QPSK)	27
2.5.2 Quadrature amplitude modulation (QAM)	30
2.6 Multi-Carrier System Fundamentals	32
2.6.1 OFDM	32
2.6.2 IFFT and FFT	34
2.6.3 Cyclic Extension/Prefix (CP)	36
2.7 Wireless Channel Characteristics	37
2.7.1 Fading Channels	38
2.8 Channel Estimation and Equalization over Inter-symbol Interference Channels	39
2.9 Diversity Techniques	41
2.9.1 Spatial Diversity and MIMO	42
2.9.2 Maximal Ratio Combining (MRC)	44

2.9.3	MIMO Channel Precoding Systems	46
2.10	4G-LTE and Beyond-5G	50
3	LabVIEW Design	59
3.1	SISO System	59
3.1.1	Model Design	59
3.1.2	Performance Analyzers Design	81
3.1.3	Graphical User Interface (GUI)	91
3.2	SIMO System	93
3.2.1	SIMO Model Design	93
3.3	USRP Design	95
3.3.1	SISO	95
3.3.2	2x2 MIMO	97
4	SDR and USRP Real-World Scenario Testbed	103
4.1	Scenario Description	103
4.2	Performance Analysis	106
5	Results	111
5.1	BER Performance Analysis	111
5.1.1	SISO	111
5.1.2	SIMO	114
5.2	Real World Testbed Performance Analysis	115
5.2.1	2x2 MIMO Scenario	115
5.2.2	MIMO Channel Precoding	118
6	Discussion	119
7	Conclusion	123
	Bibliography	127
A	120 Years of Radio Communication Engineering	129
B	Software Defined Radio and USRP Radio Hardware	135
C	LabVIEW Additional Implementations	139
C.1	SISO System	139
C.1.1	Global Parameters	139
C.1.2	Transmitter	140
C.1.3	Channel	146
C.1.4	Receiver	148
C.1.5	Graphical User Interface (GUI)	154
C.2	SIMO System	161
C.2.1	SIMO Channel	161
C.3	USRP Design	162
C.3.1	2x2 MIMO Transmitter	162
C.3.2	2x2 MIMO Receiver	162

D Additional Simulation Features	165
D.1 Save Results to File	165
D.1.1 Diagram Design	166
D.1.2 GUI Design	170
D.2 MATLAB Scripts	171
D.2.1 Get Throughput and Obtain the CDF	171
D.2.2 Generate Precoding Matrix	173
D.2.3 Calculate SINR	173
D.2.4 Average SINR of All the Subcarriers	174
D.2.5 Calculate Throughput	174

Preface

It has been already nine years since I initiated my university education on telecommunication engineering. From the defined five years degree, I decided to make a detour to universities abroad with the purpose of enriching my cultural and academic background. During these adventures, I acquired wide knowledge in the communications, electronics and computer science fields. Nevertheless, this experience has revealed to me that the more I studied different subjects the less I retained everything I learned.

On account of that, I resolved to culminate my degree with this 8 months thesis work in pursuance of reviewing the mobile communication systems that drew my attention at the begging of this long journey.

The idea for the development of this project came due to the intensive research being done to designing the 5G mobile broadband generation. As the simulations and tests require preparation time, it was determined to employ a new software defined radio (SDR) known as LabVIEW (LV) which was proven to offer design flexibility, fast development and a friendly graphical user interface. Therefore, the implementation of a real-world scenario testbed by means of the SDR based on the USRP radio hardwares gave us the opportunity to test some of the techniques that enhanced the channel capacity.

This thesis is written and presented in a sequential structure according to the development process. First, we describe the software implementation of a digital transmission system, starting from the transmitter, then the channel and finally the receiver. All the designs are implemented exclusively with LV. Similarly, the performance results are computed within the SDR. The LV diagrams illustrated in our Figures at the main body of this project are designed by the author. On the other hand, additional modules and scripts that are secondary or obtained from external sources are presented in the appendixes. Regarding the testbed scenario, we provide in this work with the configuration basics of the USRPs in order to set up a 2x2 MIMO system. We depict our indoor scenario and we show our nodes location for our different realization scenarios. The results of our simulations and real-world testbed are presented and discussed at the end.

We conclude our thesis work by confirming that the techniques employed in our testbed boost the network throughput. Furthermore, we agree that the SDR LV grants with development tools to simulating most of the blocks present in our mobile communications. We note that only the precoding is analyzed in the testbed while there are more available techniques that achieve also capacity gains and higher performances.

First of all, I would like to express my gratitude to my thesis supervisor **Troels Bundgaard Sørensen, Dereje A. Wassie** and **Gilberto Berardinelli** for their professional guidance through all the design processes of a wireless communication system. Furthermore, their unconditional patience due to my academic background memory gaps was truly appreciated.

A very special thanks goes to **Pablo Fuentes Pazos** who helped me with the endless LabVIEW obstacles. Without his resolute findings I could have not progressed in my implementations and hence I would have spent weeks or even months trying to find the design errors.

I would like to express my sincere gratitude to **Troels Bastholm**, who made my daily work at the office agreeable. Moreover, his support and guidance in cultural matters such as my adaptation into a new country made possible the conclusion of this project.

From my home university, I would also take this chance to thank **Montse Targarona** for her assistance and counseling in international affairs during my education abroad. Moreover, I owe my gratitude to my supervisor in Barcelona Professor **Joan M. Gené Bernaus** for his helpful disposition.

Finally, on a more personal note, my deepest gratitude is expressed to my mother and closest friends whose encouragement has given me strength to keep working with an optimistic attitude.

Aalborg University, April 23, 2015

Manuel Sainz
<msainz14@student.aau.dk>

Acronyms

5G	5-Generation
A/D	Analog to Digital
AM	Amplitude Modulation
AMU	Antenna Matching Unit
AP	Access Point
ARQ	Automatic Repeat Request
AWGN	Additive White Gaussian Noise
BCH	Broadcast Control Channel
BER	Bit Error Rate
BLER	Block Error Rate
BS	Base Station
BW	Bandwidth
CAZAC	Constant Amplitude Zero Autocorrelation
CDF	Cumulative Distribution Function
CDMA	Code Division Multiple Access
CP	Cyclic Prefix
CSI	Channel State Information
D/A	Digital to Analog
DL	Downlink
EGC	Equal Gain Combining
eNB	Evolved Node-B
FDM	Frequency Division Multiplexing
FFT	Fast Fourier Transform
FM	Frequency Modulation
GUI	Graphic User Interface
ICI	Inter Carrier Interference
ICIC	Intercell Interference Coordination
IF	Intermediate Frequency
IFFT	Inverse Fast Fourier Transform
IMTS	Improved Mobile Telephone Service
IRC	Interference Rejection Combining
ISI	Inter Symbol Interference
LAN	Local Area Networks
LDPC	Low Density Parity Check
LFSR	Linear Feedback Shift Registers
LMMSE	Linear Minimum Mean Square Error
LOS	Line of Sight

LPC	Linear Predicting Code
LSB	Less Significant Bit
LTE	Long Term Evolution
LV	LabVIEW
M2M	Machine to Machine
MAN	Metropolitan Area Networks
MIMO	Multiple Input Multiple Output
MRC	Maximal Ratio Combining
NI	National Instruments
NLOS	Non Line of Sight
OFDM	Orthogonal Frequency Division Multiplexing
P/S	Parallel to Serial
PCM	Pulse Code Modulation
PDF	Probability Density Function
PDP	Power Delay Profile
QAM	Quadrature Amplitude Modulation
QPSK	Quadrature Phase Shift Keying
RF	Radio Frequency
RTT	Round Trip Time
S/P	Serial to Parallel
SDR	Software Defined Radio
SIC	Successive Interference Cancellation
SIMO	Single Input Multiple Output
SINR	Signal to Interference plus Noise Ratio
SISO	Single Input Single Output
SNR	Signal to Noise Ratio
SVD	Single Value Decomposition
TDD	Time Division Duplex
TDM	Time Division Multiplexing
TDMA	Time Division Multiple Access
UE	User Equipment
UL	Uplink
USRP	Universal Software Radio Peripheral
VI	Virtual Instrument
WAN	Wide Area Networks
WCDMA	Wideband Code Division Multiple Access
ZC	Zadoff-Chu
ZF	Zero Forcing

Chapter 1

Introduction

1.1 Motivation

From the first communication device to the brand new smartphones we have increased exponentially the complexity of the communications techniques and particularly the amount of information sent and received. Furthermore, the number of mobile users has grown deriving to bandwidth limitations, among others. Therefore, in order to cope with these issues we aim to increase mainly the efficiency and capacity through the successive generations of our cellular communications systems.

The dramatic increase of users data traffic in the next decade [1][2] requires the design and implementation of advanced techniques for the coming cellular generation 5G which is going to cope with these needs after 2020. But before the transition to the fifth generation it is of significant importance to test the performance of these novel techniques. For example, the latency, the power consumption and spectral efficiency are some of the key factors of the next cellular generation [2]. Apart from the simulated tests, mobile services providers strongly rely on real world testbed scenarios, where relevant features can be obtained.

The difficulty in simulating the real world network topology makes experimental trials a fundamental source of information for the performance analysis. For instance, as the propagation model limits the simulation assumptions by acquiring experimental data from practical scenarios we are able to validate our hypothesis [3]. It is for this reason that testbed networks are essential to proof the effectiveness of the novel techniques that will boost the network throughput.

Currently, a software platform called ASGARD is being used for experimentation at Aalborg University. It provides implementation of cognitive radio communication systems over Software Defined Radio (SDR) equipment. Moreover, it eases the configuration of multi-layered communication systems and its platform architecture is component based. However, the use of a new software called LabVIEW (LV) is foreseen for future testbed scenarios. By way of its component libraries and graphical development environment, LV provides comprehensive tools to configure our workbench in easy steps. Furthermore, it supports a wide reconfigurable hardware compatibility that can be used to build any measurement or control application in less time.

LabVIEW has been developed by National Instruments (NI) who also offers hardware solutions for rapid prototyping of high-performance wireless communication systems. In point of fact, Universal Software Radio Peripherals (USRP) are used for the ASGARD testbeds. In this project, the NI USRP N200 (see Appendix B) are used with the LV communication system design for the testbed on a real-world scenario.

1.2 Background

Mobile communications systems have evolved in the last decades to cope with the growth of data traffic. While years ago users owned one mobile device, nowadays communication devices such as smartphones, tablets and laptops are concentrated in small areas and demand a high wireless data traffic. Therefore, a significant effort is being put forth on the definition of the 5th Generation of cellular systems to cope with this demand.

According to the Cisco global mobile data traffic forecast for 2018 [4], the evolution of the data traffic within the next five years reveals a dramatic impact on the cellular networks. For example, the estimated data traffic growth is presented on the Figure 1.1. As we can see, nearly an 11-fold increase over 2013 is predicted and the mobile data traffic will reach the amount of 15.9 Exabytes by 2018.

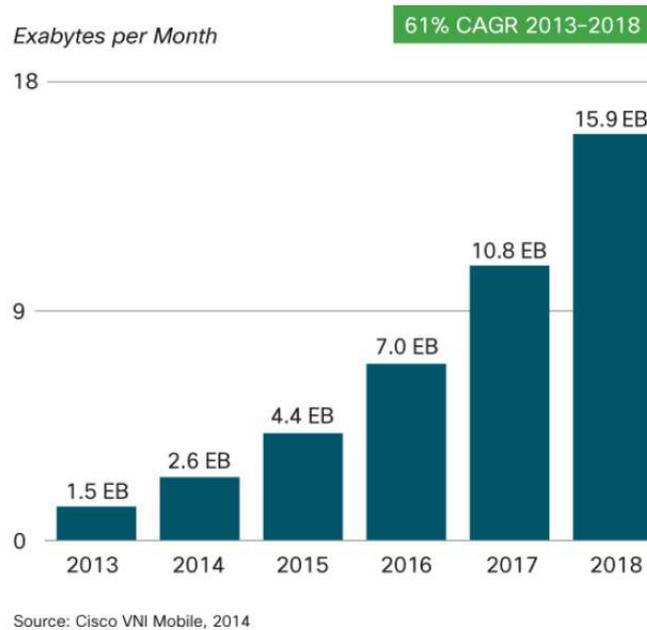


Figure 1.1: Cisco Forecats 15.9 Exabytes per Month of Mobile Data Traffic by 2018.

Although the 4G-LTE network is being used by more devices every day increasing the data traffic capacity, it is clear that we need to develop the next cellular generation with advanced techniques able to support the predicted exponential data increase.

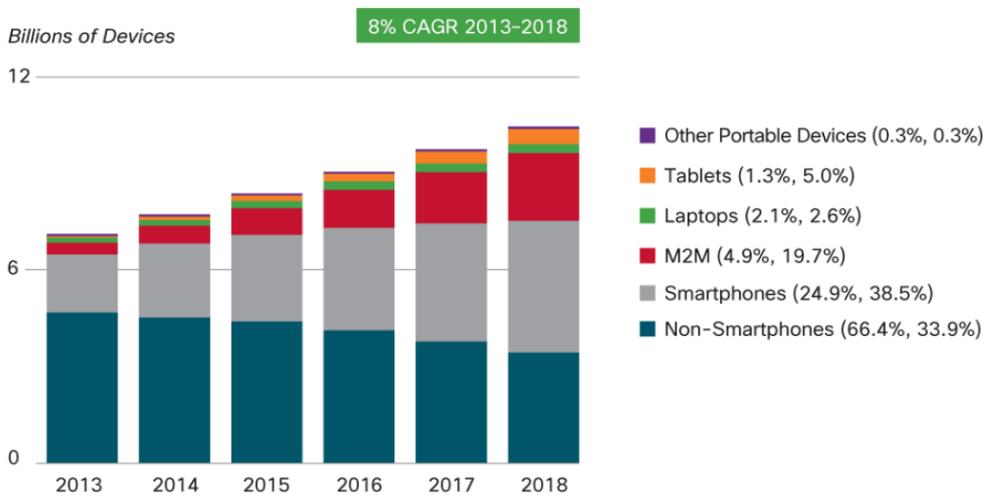
We outline some more relevant facts that are foreseen in the next years:

- The number of mobile-connected devices will exceed the world's population by 2014.
- Smartphones will reach 66 percent of mobile data traffic, while tables 15 percent.
- 4G traffic usage will be over the half of the total mobile traffic.
- The mobile data traffic will surpass 15 exabytes by 2018, six times higher than the traffic by 2014.
- The mobile network connection speeds will increase 2-fold by 2018.

The major trends that contribute to the growth of mobile data traffic are outlined as follows:

- Smarter Mobile Devices leading the market.
- Emerging of Wearable Devices.
- The Dominance of Video Applications.
- Bandwidth Consumption by Device.
- Assessing Mobile Traffic/Offload by Access Type (2G, 3G and 4G).
- The Adoption of IPv6 – Emerging Protocol.
- Definition of Mobile “Prime Time” – Peak vs. Average Usage.

These expectations among other predictions suppose a challenge for the upcoming mobile generations. Therefore, a big effort is being done in research to cope with this traffic demand and design more efficient communication techniques. As we see in the Figure 1.2, the number of wireless devices is increasing and it is the main cause to the global mobile traffic growth.



Figures in parentheses refer to device or connections share in 2013, 2018.
Source: Cisco VNI Mobile, 2014

Figure 1.2: Global Mobile Devices and Connections Growth.

Tablets are the most remarkable growth, followed by machine-to-machine connections (M2M), both increasing nearly six fold from 2013 to 2018. Every year the market is being updated with new devices with more capabilities and intelligence. These smart equipment have a wide field of application (e.g. automotive industry, medical applications, manufacturing sectors, et al). Although these trends have global influence, some regions are more sensitive to these changes than others. As we show in the Table 1.3, the leading countries to the transition to smart devices are North America and Western Europe with over a 90 percent and 80 percent of expansion respectively by 2018.

Region	2013	2018
North America	65%	93%
Western Europe	45%	83%
Central and Eastern Europe	15%	61%
Latin America	14%	55%
Asia Pacific	17%	47%
Middle East and Africa	10%	36%

Source: Cisco VNI Mobile, 2014

Figure 1.3: Regional Share of Smart Devices and Connections (Percent of the Regional Total).

If we compare the growth of traffic with the transition to the smart devices in the following years, we identify that on average smart devices generate higher traffic than a non-smart device. Moreover, we observe that by 2018 smart traffic is going to raise from 88 percent of the total global mobile traffic to 96 percent. Another important factor on these contributions is the evolution from lower-generation networks connectivity (2G) to higher-generation networks connectivity (3G, 3.5G, and 4G-LTE). In other words, faster and more intelligent networks lead to advanced multimedia applications that increase dramatically the data traffic rate.

As mentioned before, 4G-LTE will be slowly adopted by more smart devices in the following years. We illustrate this generation transition in the Figure 1.4. All in all, 4G will be an important factor for the improvement of the network performance, due to its higher bandwidth, lower latency and increased security.

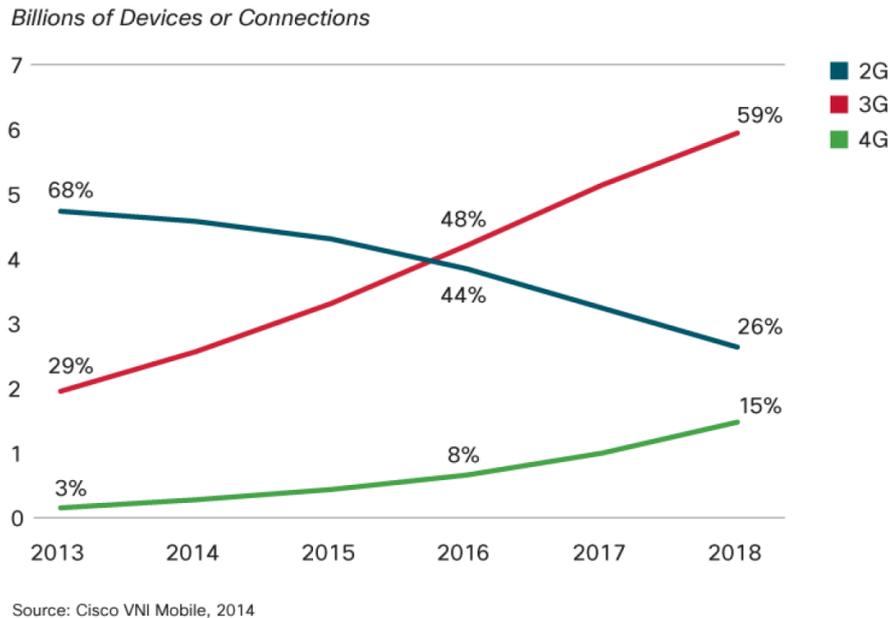


Figure 1.4: Global Mobile Devices and Connections by 2G, 3G, and 4G.

In spite of that, we need to think what will happen beyond 4G. It is expected that the fifth mobile generation will be introduced by 2020 and cope with the traffic demands until 2030. Therefore, it is necessary to design and develop advanced and novel techniques that will surpass the capabilities of the 4G and at the same time, compete in equally conditions with the continuous evolution of the devices technology for ten years.

Currently, there are two main approaches to cope with this necessity. One is focused in the improvement of the cellular base station (BS) capacity. This is achieved with massive multiple input multiple output (MIMO) arrays of antennas. The other approach is targeted on local area networks, where the user equipment (UE) connection is coordinated by local indoor access points (AP). These AP through time division and interference mitigation techniques grant wireless connectivity to a higher number of users avoiding coexistence issues in the wireless network. On the same way, this also helps to reduce the capacity demands on the BS.

Right now, novel 5G techniques for local networks are being tested on simulations and real-world scenarios. We summarize from the related work the major enhancements of these techniques on the network performance in the following paragraphs. First of all, we review the most significant key performance indicators of 5G. Finally, we present some of the advanced designs and techniques that improve the network's performance.

As described in more detail in [2], mobile broadband systems will have to become more capable of meeting the demands mentioned before. For achieving this goal, the relevant key requirements and their significance are illustrated in the Figure 1.5.

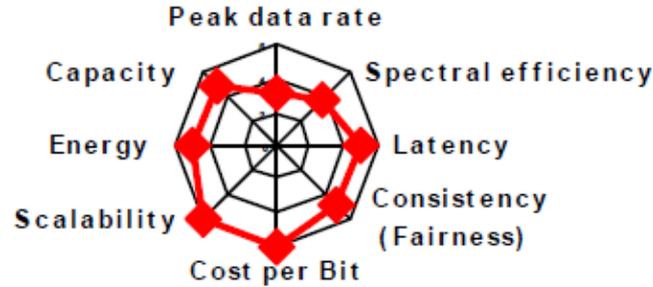


Figure 1.5: Relevant key requirements for B4G.

As shown, the most important design targets are the cost per bit and the system scalability to heterogeneous deployment scenarios. A target factor is also the energy efficiency that allow us to increase the battery life of the users' equipment. This factor has an influence in the operational expenses and consequently to the cost per bit. In order to meet some of these requirements, there are several key enablers which are beneficial for local areas. For example, a low latency frame structure allows minimum round trip times (RTT) from the current LTE-Advanced system of 10ms to a desired 0.1 - 1ms for 5G designs. Thus, the frame structure is the critical element that affects the latency which has a direct influence with the buffers size. Therefore, a new frame structure design leads to cost savings and low latency [5].

Moreover, we boost the throughput reducing the overhead by decreasing the cyclic prefix (CP) length, for instance. However, optimizations for latency and efficiency are conflicting and thus it is necessary to provide more versatility to adapt the system to different deployment scenarios [6]. Another enabler technique is the interference mitigation. Inter-cell interference in dense cells scenarios can be managed with frequency reuse techniques [7] and Interference Rejection Combining (IRC) advanced receivers which elevate the data rate in interference limited conditions [8]. Further interference coordination solutions are found in related work on real-world deployment conditions [9].

As LTE/LTE-Advanced is optimized for wide area environments, 5G is designed to optimize local areas. Hence, it is feasible to maximize the similarities between uplink (UL) and downlink (DL) and as a result similar multiple access schemes could be used in both directions, simplifying the design and hardware implementation of the system. Because of that, Time Division Duplex (TDD) mode gains relevance for 5G air interface designs providing more capacity and efficiency locally.

To sum up, the most significant 5G design key indicators for small cells [6] are the following:

- Peak data rate in the order of 10 Gbps.
- Round Trip Time (RTT) around 0.1 – 1ms.
- Spectral efficiency ~ 2 better than 4G.
- Low power consumption of both access point and terminals.
- Efficient support of Machine Type Communication (MTC).
- Flexible spectrum usage.
- Self-optimization of ultra-dense deployment of access points.
- Support of multi-hop.
- Simple and low cost design.

1.3 Objectives

We present in this paper a software defined radio designed with LabVIEW and adapted for the NI USRP hardware, see Appendix B. We aim to realize a testbed within a real-world indoor scenario. The purpose is to address the effective potential of promising 5G techniques in boosting the network throughput.

On the one hand, we design and implement a complete wireless digital communication system on LabVIEW. The transmitter design comprehends from the bits generation to the OFDM symbols and frame transmission. Then, the USRP performs the up-conversion from digital data to analog signal and the transmission through the air over the channel. The USRP at the receiver down-convert the signal and the digital frames are processed in the LV receiver modules. These modules by means of transmitted symbol pilots estimate the channel. Moreover, we target to develop a reconfigurable testbed which will allow us to adapt our communication system for further novel techniques currently in development.

On the other hand, we build a USRP network in a real-world scenario where we employ some of the 5G techniques, particularly precoding schemes for MIMO scenarios. We aim to obtain experimental measurements that proof our theoretical assumptions on the improvement of the network performance. Basically, we focus on local area networks where the performance of space diversity techniques such as codebook based precoding matrices is analyzed.

The main features of these novel techniques of the fifth mobile generation allow us to cope with the data traffic demands in the next decade. Furthermore, we look forward to meeting the 5G key indicator requirements implementing these techniques and testing them over a real-world deployment scenario.

Apart from the proof-of-concept goal, this project provides a transition from the current software platform ASGARD to LabVIEW, which allow us to design a communication system in an easier graphical way and adaptable to different scenarios and hardware equipment.

We present the main goals of this project along these lines:

- Design and implementation of a complete SISO digital communication system with LabVIEW.
- Enhancement of the LV design from SISO to SIMO and MIMO.
- Integration of the LV SDR with the USRP radio hardware.
- Realization of a 2x2 MIMO real-world testbed scenario.
- Verification and validation of the precoding techniques potential in boosting the network throughput.

One of the main reasons and motivation for the transition from ASGARD to LabVIEW is because of the NI USRP equipment configuration simplicity, that allows us to build a test environment with two USRP in parallel obtaining a MIMO system. Moreover, the SDR consist of the equivalent modules of a real communications system but more flexible to different parameters configurations and features. Hence, we aim to design and implement the essential communication modules to simulate a wireless digital transmission-reception system. Furthermore, we adapt it with the USRP radio hardware within a real-world indoor testbed scenario.

To summarize, this project aims to give an insight into the performance of some of the 5G novel techniques that may be part of the next mobile broadband generation after the 4G-LTE Advanced. The software defined radio using LV and the USRP hardware are the basis of the performance testbed.

1.4 Guidelines

This report is structured as follow:

- Chapter 1 Introduction.** We describe the motivation of this project. Following, we overview the background scenario and we present the objectives.
- Chapter 2 Literature Review.** In this chapter, we start outlining the theory that encloses the concepts related to a basic digital communication system. First, we examine the main communications system blocks of a wireless communication channel. Second, we define the main performance measurements implemented for the validation of the simulations. Then, in the next sections we address the theory behind the modules and techniques implemented in LV such as source and channel coding, convolutional codes, modulation schemes and multi-carrier system fundamentals. Afterwards, we target the wireless channel presenting the models and its characteristics. Moreover, we present channel estimation techniques over inter-symbol interference environments. We follow by describing the diversity techniques employed in multi-antenna systems such as maximal ratio combining and channel precoding. The last section of this chapter summarizes the main characteristics of the current 4G-LTE mobile broadband, as a reference point for the design of the 5G generation.
- Chapter 3 LabVIEW Design.** First of all, we describe the software implementation of a SISO system, its model, performance analyzers and graphical user interface. Secondly, we enhance the implementation for a SIMO system. Then, we adapt our LabVIEW

software design to the USRP radio hardware, being able to transmit and receive data. Finally, we realize further LV implementations for a 2x2 MIMO system using our USRPs.

- Chapter 4 SDR and USRP Real-World Scenario Testbed.** First, we describe the 2x2 MIMO real-world tesbed scenario using the USRPs. Then, we determine the testbed USRP configuration parameters and its performance analysis.
- Chapter 5 Results.** We provide the performance measurement results of the simulations, in particular the BER vs. SNR for the SISO and SIMO channels. Then, we present the performance results of the MIMO USRP testbed. Finally, we expose the results obtained boosting the network throughput by exploiting the MIMO diversity using precoding techniques.
- Chapter 6 Discussion.** In this chapter, we discuss and interpret the results presented in the previous chapter. Furthermore, we address some software and hardware development factors that influenced the system performance and determined the tesbed configuration.
- Chapter 7 Conclusion.** We describe the main findings of this project, its significance and limitations. Moreover, we review our project goals and refer them to the results obtained.
- Appendix A 120 Years of Radio Communication Engineering.** We briefly review 120 years of evolution of the radio communications, from its origins the telegraph to the digital era and smart-phones.
- Appendix B Software Defined Radio and USRP Radio Hardware.** We present the National Instruments USRPs Data-sheet and we describe its capabilities using the software defined radio.
- Appendix C LabVIEW Additional Implementations.** We present some additional LabVIEW modules that are part from the SDR. We follow the same structure as in Chapter 3.
- Appendix D Additional Simulation Features.** We introduce in this appendix some extended features implemented in LV and the MATLAB scripts employed for the throughput calculation using the precoding matrices and the estimated channel from our real world scenario testbed.

Chapter 2

Digital Communications Systems

In spite of the excellent literature resources available about wireless communication systems, the author of this project prefers to briefly review the radio fundamentals and the theory of the basic modules of a digital transmission system. In the first section we present the modules from the transmitter and received part, besides the channel model. Then, we examine the main characteristics of current techniques and designs employed in digital communications systems in almost every device and station.

2.1 Elements of a Digital Communication System

In this section, we describe the components needed to generate a basic digital communication system. We present sequentially the system block from the transmitter to the receiver. The goal of a digital communication is to send bits in the most efficient and fastest way possible between two distant points, using a channel medium, in our case the air. Furthermore, the original bits have to be recovered without bit errors or keeping them as less as possible.

Before a signal is called a digital signal or even thinking on processing a signal into a digital communication system, we must perform the process of conversion of an analog signal into an appropriate form in the digital domain. Analog-to-digital conversion is a very basic process which is implemented in the input stages of a digital communication system. Many of the signals that derive from the physical world are inherently analog in nature. Hence, if we aim to transmit these signals digitally, it is evident that it is necessary to convert these analog signals to digital ones. While doing so, we must ensure that within the conversion process we do not introduce inaccuracies or at least not beyond a limit, so that the quality of the produced signal somewhere in the receiver is always satisfactory.

After the conversion, comes the point of signal representation, or to be more precise the signal sampling. We use an approximate method called quantification and after it we represent the quantized versions in some different ways (coding), such as Pulse Code Modulation (PCM), Differential PCM and Linear Predicting Code (LPC).

In this project, because we design a digital system we will not describe these coding steps in the quantization process. We directly assume our information source as a digital signal, namely stream of bits. The topic of baseband transmission is elemental for understanding the concept of signal shaping, digital modulation, digital reception and other related topics.

Inter-symbol interference is a limitation which is found in many occasions when we try to send pulses. The signal duration in time domain increases when we reduce the bandwidth of the signal. That means that if we transmit in a high bitrate, the transmission bandwidth is limited and if the pulses are not designed properly, there is the possibility that a single pulse interferences the ones next to it and consequently the performance of the communication system decreases.

Digital modulation and demodulation translate the signal from one frequency band (baseband) to another high frequency band, intermediate frequency (IF), or even to an actual transmission bandwidth called Radio Frequency (RF) bandwidth. A digital communication system is an information processing system whose objective is to send the information of a transmission user and ensure that it is delivered without distortions and errors to the designated destination. However, errors are inevitable and by source coding we are able to correct most of them and also protect our information against distortions mainly caused by the additive channel noise, interferences or fading.

In overview, the communication link is composed by the transmitter unit, the channel and the receiver unit. We depict in Figure 2.1 this basic structure.

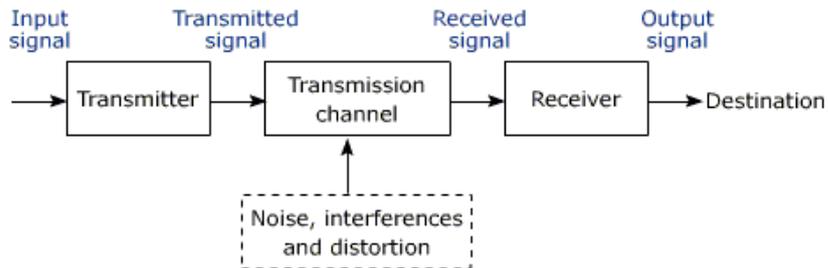


Figure 2.1: Basic communication system principle.

We assume for our channel model a wireless communication through the air. The two units consist of several blocks or subsystems and their design depend on the characteristics of the physical channel.

In Figure 2.2 we illustrate in more detail a representative example of the block diagram of a complete digital communication system, where AMU is the antenna matching unit module and f_I and f_{LO} are the intermediate frequency and the local oscillator frequency, respectively.

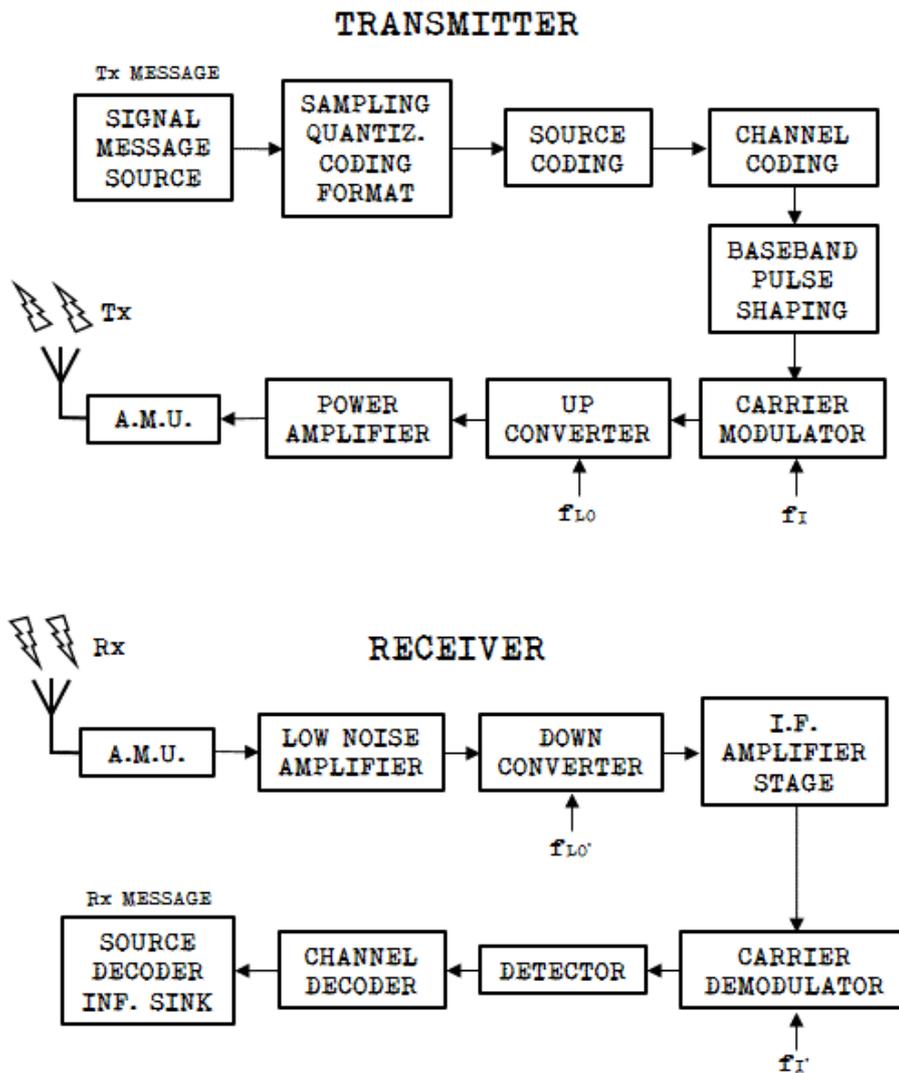


Figure 2.2: Block diagram of a communication system.

2.1.1 Transmitter

Starting with the transmitter design, the first block that we find in the system is the signal or *message source*. As explained before, the signal in our context is an electrical representation of a physical manifestation. For instance, the human speech after conversion to an electrical signal using a transducer (microphone).

We name it message source because it has something of interest that we want to transmit from the transmitter to the receiver. In order to design the communication system, we must consider as important the features of the electrical output signal. Such features are the bandwidth, power and the statistical characteristics, which are stationary that means that they do not change on time [10].

If the signal is analog in nature, the next module is considered the sampling, quantization and associated coding for a better representation format. Otherwise, if it is digital, then this module only formats the signal in order to send it more conveniently. Moreover, this module is a pre-processing unit that gives as an output a digital signal which conserves its initial statistic features and whose bandwidth depends on the sampling rate. We define this output signal as the *information bearing sequence*.

The Figure 2.3 shows the block diagram of the above mentioned units, known as the *information source*.

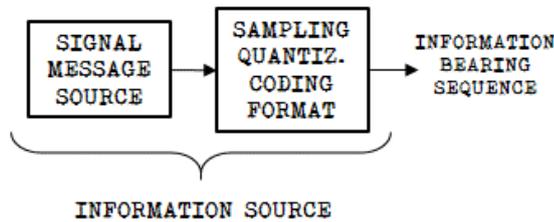


Figure 2.3: Information source Block.

Now, the next two units process the information bearing sequence in several ways. The first one has to know the properties of its predecessor blocks and the signal source. The second unit takes into account the properties of the coming transmission modules. Hence, it prepares the information in the best way for the subsequent blocks before the signal reaches the receiver. We depict them in the Figure 2.4, which represents the information processing block.

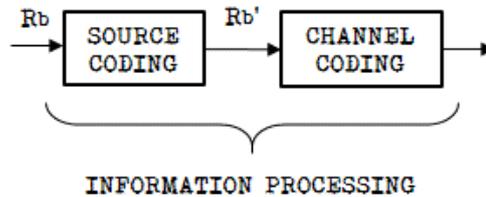


Figure 2.4: Information processing Block.

If the sampling rate is high, between two consecutive samples the difference on magnitude is small. That means that exists a strong correlation between near points, obviously due to the continuous nature of analog signals. Hence, high sampling rates produce a discrete signal whose values are correlated with their neighbor ones. Considering this fact, as example for an eight bits representation at the quantization block we assure that two consecutive samples differ in only the two or three less significant bits (LSB). In these cases, it is possible to process our information bearing sequence, namely *source encoding* or *data compression*, in such a way that we transmit less bits for representing the same information [11].

That procedure has two important consequences. First of all, the source coding technique has to be known by the transmitter and the receiver, because this process has to be reverted at the receiver in order to obtain the original bit stream. Secondly, if our input signal has a defined bit rate (e.g. R_b), due to this source coding the output signal will present a different bit rate (e.g. $R_{b'}$) which is lower than the input signal, ($R_{b'} < R_b$). We say then that the purpose of this block is meaningful because the effective bit rate is reduced and consequently the bandwidth too. That benefits easing the complexity of the successive blocks and also shortens the message into elementary symbols, which are limited in number while transmitting them in real communication systems.

In order to preserve the quality of the information, we have to assure that the source coding block does not produce any distortions to the input signal that the receiver could not reverse. In practice there are two major schemes for this method of reducing the bit rate, *source coding*. One technique allows some level of signal quality degradation. The other one does not allow any losses.

The second unit of the information processing blocks is known as *channel encoder* or *channel coding*. It tries to prevent some of the problems that may occur in the channel, which is discrete and not related to the channel medium. In other words, the channel coding safeguards the information bearing sequence against any possible distortion or error that may take place during the transmission. It usually introduces additional redundancy in the binary information sequence that is used while decoding in order to avoid some of the effects of noise and interference during the transmission. Thus, the redundancy increases the fidelity of the received signal and the reliability of the received data. This is possible because of the removed redundancy at the *source encoder* has no correlation to the controlled redundancy added at the *channel encoder*, which is defined.

The next block, especially important for wireless communications, is the baseband pulse shaping unit. Its output are analog pulses but it does not have a correspondence to a digital-to-analog converter. In fact, we allocate this block within the digital modulation unit which follows. We illustrate in Figure 2.5 how a digital signal represented by ‘0’ and ‘1’ is pulse shaped into an analog pulse which bears the same information with less bandwidth compared to the input signal.

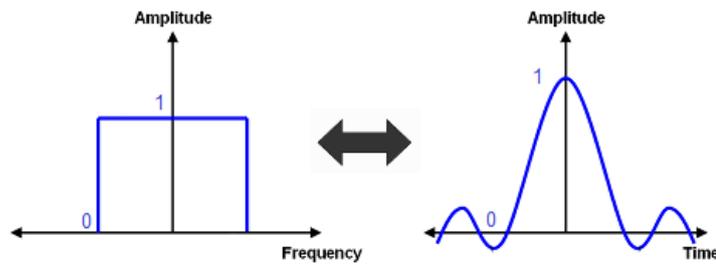


Figure 2.5: Pulse shaping.

Our information bearing sequence has low frequency components and consequently is considered a baseband signal. On the other hand, the carrier modulator unit employs a carrier of intermediate frequency (IF) of sinusoidal nature in order to shift the baseband signal. As explained in the evolution of wireless communications in the Appendix A, while translating low frequency signal to a higher frequency we are able to successfully transmit a signal via radio.

The baseband pulse shaping unit and the carrier modulator unit form the digital modulator block. The output of this block is an analog continuous-wave bandpass signal, so it converts the input binary stream into a waveform sequence suitable for transmission over the channel. Its bandwidth depends on the type of modulation used and it is centered at the IF. That means that the digital modulator block is a narrow-band modulator.

The next block is conceptually a frequency translator or frequency up-converter block. This unit requires another input defined as F_{LO} (local oscillator frequency), which is a high frequency carrier. The up-converter conceals the idea of a simple translation among the frequency axes to the actual frequency band which has been allocated for the purpose of transmission. As output we still obtain a narrow bandpass signal. In fact, it is narrower in comparison to the center frequency. However, this signal is not suited for direct transmission, mainly because its low power. Therefore, in order to get a high power signal we implement a block whose major function is the amplification of the signal. As a result, the signal is boosted in power obtaining a narrow band signal much stronger in terms of power.

Before the transmission of the signal through the antenna, we must match the output impedances of the power amplifier and the load of the antenna. That is realized at the next stage unit called antenna matching unit (AMU). Without this unit, the power amplifier gets damaged because of the power not transmitted which is dissipated within the amplifier mostly by heat. Hence, the AMU ensures that most of the power is transmitted to the channel medium through the antenna or transmitting point.

The figure 2.6 shows the above mentioned blocks which form the last units of the transmitter in our case example of communications system.

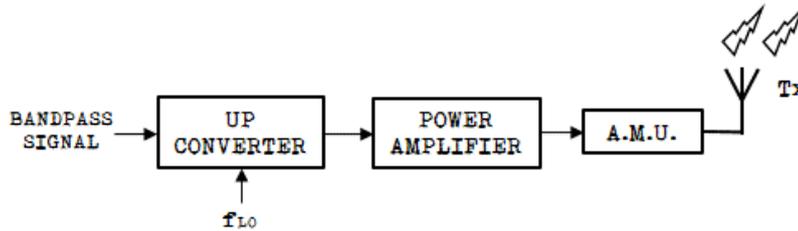


Figure 2.6: Modulation and Radio Channel transmitter Blocks.

2.1.2 Channel

The physical channel in a wireless system is the air and it is considered as the free space, although small particles, as for example dust, distort the signal. On the other hand, telephony channels may employ as physical medium wire lines, optical fiber cables and radio [12]. This scenario is represented in Figure 2.7. Whatever the medium, the signal travels losing some strength and being randomly corrupted until it is finally captured by the received antenna.

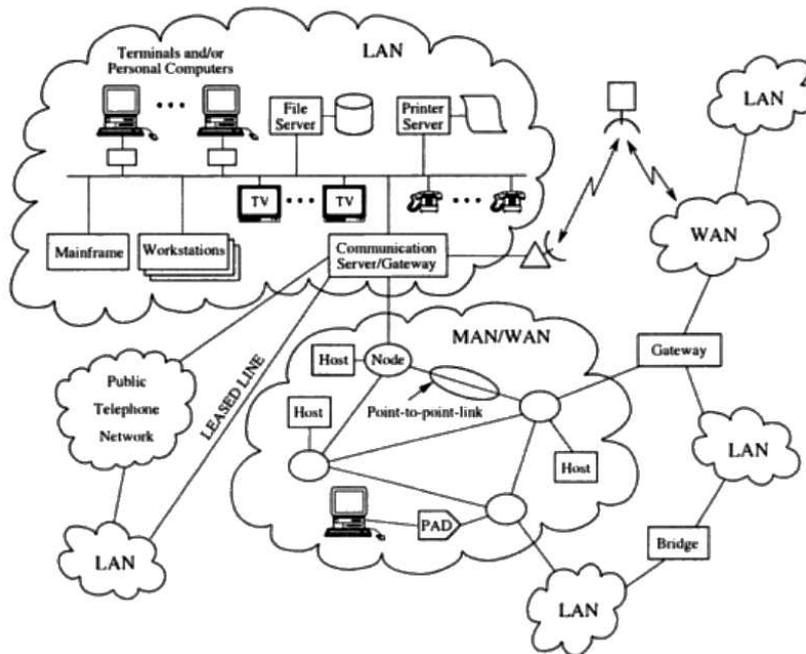


Figure 2.7: Local area networks (LAN), metropolitan area networks (MAN), and wide area networks (WAN) interconnected through several telecommunication links [12].

The wireless channel specially adds significant amount of noise during the process of transmission. We define by *propagation channel* the free space link between antennas, and by *radio channel* when we consider the antennas. Then, the attenuation of the signal, referred to as *fading*, occurs within the propagation channel and the additive thermal noise is added at the received antenna, within the radio channel. It is for these reasons, that a good communication system design is able to counteract these effects so as to obtain a faithful estimate of the transmitted signal.

2.1.3 Receiver

The receiver in a communications system has basically the same units as the transmitter but with the purpose of reversing the transmitter transformations, detecting and recovering the original bit stream. Moreover, additional features and units are implemented in order to correct bit errors caused by the radio channel.

We expect to receive at the receiver antenna a RF analog signal with low power and narrow band. However, the received signal may contain other interfering signals which come from other transmitters. In other words, the complexity of the receiver relies on its capacity to detect any signal and to select the desired one, diminishing the interference from the others.

This inconvenience is mostly solved by the design in the transmitter part, where our desired signal is centered in a defined frequency which is also known by the receiver. Then, the receiver filters the adjacent frequencies avoiding detection errors. There are also other interference rejection techniques and modulations that protect the information against distortions and errors caused by other signals and even by the desired signal due to multi-propagation paths.

As we mentioned before, the receiver is a frequency selective device which covers in a reverse way the same steps performed by the transmitter units. Therefore, the first three units that process the RF signal are the antenna, the AMU and the amplifier, as depicted in the Figure 2.8 at the end of this section.

The received signal is a very weak signal and it goes through the matching unit which also performs a bandpass filtering. After the suppression of the undesired signals at other frequencies, our signal is still weak and we need to amplify it in order to obtain the required levels of signal power for the subsequent processing blocks. Thus, the next unit is the low noise amplifier named like this because it should be designed in order to not generate much noise. Otherwise, our signal would be amplified but also more noise would be added by this unit. After these units we obtain a moderated amplified bandpass signal. We down-convert it in the so called unit using the same or similar local oscillator frequency (F_{LO}) from the transmitter. As a result, our weak IF signal with some background noise is finally boosted at the IF amplifier unit.

Recalling the diagram, the two units that follows form part of the *digital demodulator*. The first one is the *carrier demodulator*, which translate the signal to a baseband frequency component. The second unit is the *detector*. The digital demodulator processes the transmitted waveform distorted by the channel, and estimates the data transmitted symbols. Then, the output of the digital demodulator is passed to the *channel decoder*, which delivers the

decoded signal to the last unit, the *source decoder* or *information sink*. The channel decoder task is to recover the transmitted original information sequence using the same rule as the encoder using the redundancy contained in the received data. Finally, the source decoder obtains the message sink that is understandable to the recipient. The end signal is an approximation of the original one, and its difference is a measure of the distortion introduced during the transmission. The complete block diagram is represented in the Figure 2.8.

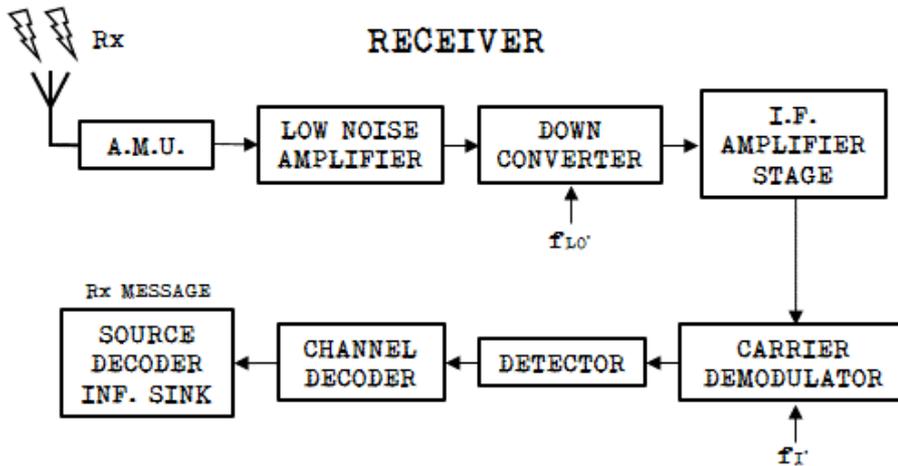


Figure 2.8: Block diagram of the receiver.

We point out that this is an elemental communication system representation, and not all of them have to have the same units or blocks. For instance, in many systems the source coding and decoding may not be there, or the initial analog-to-digital process it is not needed as in this case for a digital communication system perspective.

2.2 System Performance Measures

The concepts of *Signal-to-Noise Ratio* (SNR), *Bit Error Rate* (BER) and *Probability of Error* (P_e) justify the purpose of this section, which provides several measures of performance related to a communication system.

2.2.1 Signal-to-Noise Ratio (SNR)

This is the most common performance characteristic of a digital communication system and it is often measured at the receiver's output. It serves as indicator of the overall fidelity (quality) of the system. As its name suggests, it compares the level of our signal to the level of the background noise, defined as the ratio of signal power to the noise power, expressed in decibel (dB).

The mathematical expression of SNR is

$$SNR = 10 \log_{10} \frac{\text{Average signal power}}{\text{Noise signal power}} \text{ (dB)}. \quad (2.1)$$

The ratio of the modulated energy per information bit to the noise spectral density is

$$SNR = \frac{\text{Modulated energy per bit}}{\text{Noise spectral density}} = \frac{E_b}{N_0} \text{ (dB)}. \quad (2.2)$$

2.2.2 Probability of Error

Let us consider the example of two signal waveforms $s_1(t)$ and $s_2(t)$, antipodals $s_1(t) = -s_2(t)$, of non-zero values within the interval $0 \leq t \leq T_b$ and zero elsewhere, pulse energy of ε_b and geometrical representation of $s_1 = \sqrt{\varepsilon_b}$, $s_2 = -\sqrt{\varepsilon_b}$.

Then, let us assume that $s_1(t)$ was transmitted and the received signal form is

$$r = s_1 + n = \sqrt{\varepsilon_b} + n \quad (2.3)$$

where n represents the additive Gaussian noise component, which has a mean and variance $\sigma_n^2 = \frac{1}{2}N_0$. Moreover, the two conditional probability density functions of r for the decision rule based on geometric threshold zero are

$$p(r|s_1) = \frac{1}{\sqrt{\pi N_0}} \exp^{-(r-\sqrt{\varepsilon_b})^2/N_0} \quad (2.4)$$

$$p(r|s_2) = \frac{1}{\sqrt{\pi N_0}} \exp^{-(r+\sqrt{\varepsilon_b})^2/N_0}. \quad (2.5)$$

The probability error if $s_1(t)$ was transmitted is the probability that $r < 0$, i.e.,

$$\begin{aligned} p(r|s_1) &= \int_{-\infty}^0 p(r|s_1) dr \\ &= \frac{1}{\sqrt{\pi N_0}} \int_{-\infty}^0 \exp\left[-\frac{(r-\sqrt{\varepsilon_b})^2}{N_0}\right] dr \\ &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{-\sqrt{2\varepsilon_b/N_0}} e^{-r^2/2} dr \\ &= \frac{1}{\sqrt{2\pi}} \int_{\sqrt{2\varepsilon_b/N_0}}^{\infty} e^{-r^2/2} dr \\ &= Q\left(\sqrt{\frac{2\varepsilon_b}{N_0}}\right) \end{aligned} \quad (2.6)$$

where $Q(x)$ is the Q -function. Assuming that $s_1(t)$ and $s_2(t)$ are equally likely to be transmitted, the average probability of error is

$$p_b = \frac{1}{2}p(r|s_1) + \frac{1}{2}p(r|s_2) = Q\left(\sqrt{\frac{2\varepsilon_b}{N_0}}\right). \quad (2.7)$$

We usually call ε_b/N_0 the *signal-to-noise ratio per bit* as defined in 2.2.

Next, for binary orthogonal signals and following the same steps before we obtain the average probability for binary orthogonal signals as

$$p_b = Q\left(\sqrt{\frac{\varepsilon_b}{N_0}}\right) = Q(\sqrt{\gamma_b}), \quad (2.8)$$

where γ_b is the SNR per bit.

We observe that orthogonal signals increase an energy factor of two in contrast to antipodal signals. Hence, since $10 \log_{10} 2 = 3 \text{ dB}$ we realize that orthogonal signals are 3 dB poorer than antipodal signals, shown in Figure 2.9 [11].

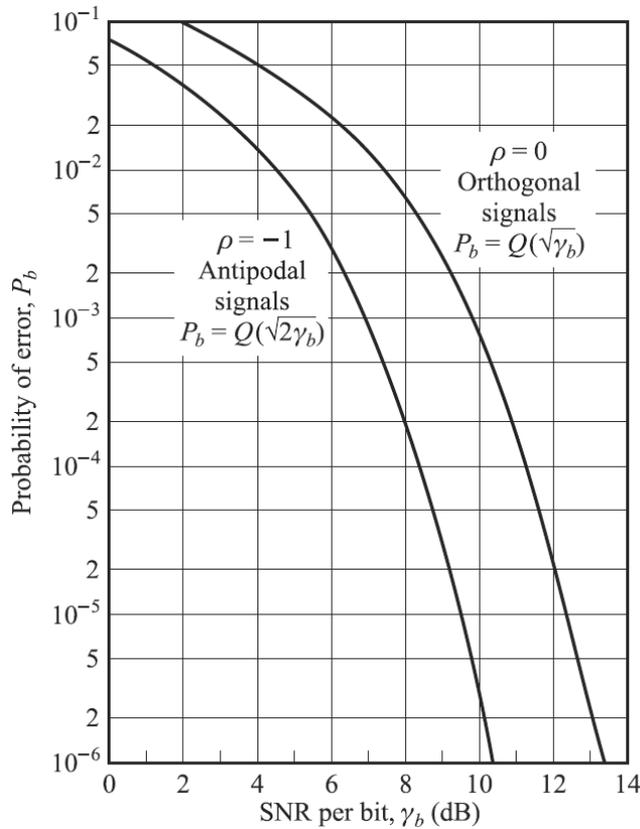


Figure 2.9: Error probability for binary antipodal and binary orthogonal signaling [11].

For M -ary orthogonal signals, we observe that the SNR is expressed in terms of symbol, ε_s/N_0 . However, with $M = 2^k$, each symbol M conveys k bits of information, and hence $\varepsilon_s = k\varepsilon_b$. We refer to the bibliography [11] for a more detailed mathematical description.

2.2.3 Bit Error Rate (BER)

This performance criterion is the most revealing about the system behavior. It is computed dividing the detected bits that are incorrect by the total number of bits sent. The wireless communications functionality is usually described using plots between BER vs. SNR. The worst-case scenario presents a maximum value BER of 0.5 (50%) for a channel where the noise dominates.

For instance, in the case of a *QPSK* modulation and AWGN channel, the BER as a function of the ε_b/N_0 is given by:

$$BER = \frac{1}{2} \operatorname{erfc}(\sqrt{\varepsilon_b/N_0}) \quad (2.9)$$

where *erfc* is the complementary error function [13].

The *packet error rate* (PER) or *block error rate* (BLER) is the number of incorrect data packets received in relation to the total sent. In this case, a packet or block is considered as error if at least one bit within is incorrect.

2.2.4 Probability Distribution Function (PDF) and Cumulative Distribution Function (CDF)

The probability and stochastic processes are the backbone of the communications systems [11] [14]. Therefore, we summarize in the following paragraphs some useful expressions that are related to our system design, such as the Gaussian, Rayleigh or Rice distributions. These distribution functions are present on the fading profiles in the wireless channel models.

We understand by PDF a function that describes the likelihood for a random variable to take on a given value. It is non-negative everywhere and the complete area under the function has maximum value of one. On the other hand, the CDF describes the probability that a random variable with a given PDF has a value less than or equal to x . It gives the area under the PDF from minus infinity to x .

Gaussian distribution (normal distribution) Its PDF for a real-valued random variable is

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(x-m_x)^2/2\sigma^2}, x \in \mathfrak{R} \quad (2.10)$$

where m_x is the mean and σ^2 is the variance.

The CDF is

$$F(x) = \frac{1}{2} + \frac{1}{2} \operatorname{erf} \left(\frac{x - m_x}{\sqrt{2}\sigma} \right) \quad (2.11)$$

The Figures 2.10 and 2.11 represent both PDF and CDF of a gaussian-distributed random variable.

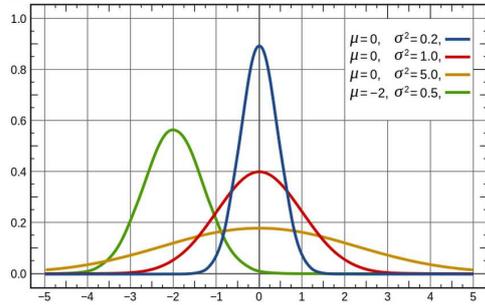


Figure 2.10: Gaussian Probability distribution function, (μ) mean and (σ^2) variance [Source Wikipedia].

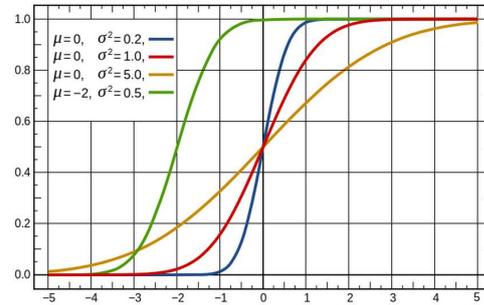


Figure 2.11: Gaussian Cumulative distribution function, (μ) mean and (σ^2) variance [Source Wikipedia].

Rayleigh distribution The equation 2.12 and 2.13 are respectively the PDF and CDF of a Rayleigh distribution which models mostly the statistics of radio signals such as cellular radio.

$$p_R(x) = \frac{x}{\sigma^2} e^{-x^2/2\sigma^2}, x \geq 0 \tag{2.12}$$

$$F_R(x) = 1 - e^{-x^2/2\sigma^2}, x \geq 0 \tag{2.13}$$

Figures 2.12 and 2.13 show their representation for a different values of σ .

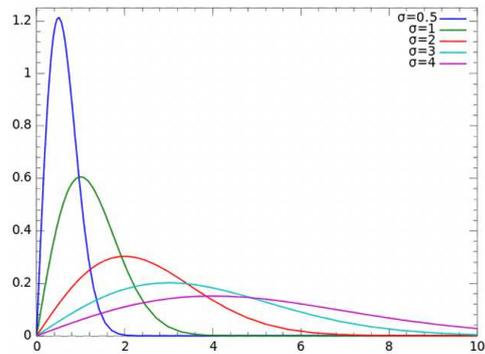


Figure 2.12: Rayleigh Probability distribution function, (σ) variance [Source Wikipedia].

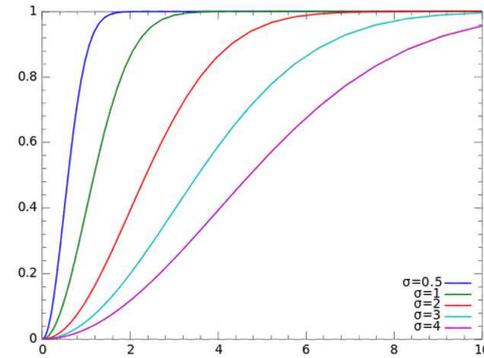


Figure 2.13: Rayleigh Cumulative distribution function, (σ) variance [Source Wikipedia].

Since the Rayleigh fading channel is the most commonly used in digital communications system analyses for wireless environments, a detailed mathematical description of the PDF and CDF for fading applications is found in "*Digital Communication over Fading Channels*" [15].

2.3 Source and Channel Coding

We compress the data of a binary source in order to transmit the same message with less bits, increasing the bit rate of the system. We call this method *source coding* and it reduces the binary data redundancy.

By grouping the binary information sequence into symbols (e.g. $x_1, x_2, x_3, x_4 \dots$) we assign to the symbols a *code word* which increases the efficiency, see table 2.1. Furthermore, the encoding method used do not introduce distortion since the encoding of symbols or block of symbols into code words is unique. There are coding algorithms such as *Huffman Coding* that guarantees an optimum coding, namely the use of the minimum bits for representing the source symbols. Moreover, symbols that occurs frequently are assigned to short codewords, whereas longer codewords are used for symbols with low probability [10]. If our source is analog then techniques such as *Pulse Code Modulation (PCM)*, *Differential Pulse Code Modulation (DPCM)* or *Delta Modulation (DM)* are used [11].

Symbol	Code Word
x_1	0
x_2	01
x_3	10
x_4	100

Table 2.1: Example of source symbols encoded into code words.

On the other hand, *channel coding* is used to introduce controlled redundancy which allows a more reliable transmission ensuring error detection and/or correction at the receiver. The channel coding applied depends on the required transmission quality. In wireless channels the problems associated to fading and interference affect the performance of the system and therefore by coding we achieve the highest possible rates. Particularly, for OFDM systems the coding is implemented in time and frequency domain.

If we take into account the operating mode of the demodulator and the decoder, the receiver performs *hard-detection* if the input signal at the decoder is strictly derived from the data. If the demodulator derives statistics to the decoder, which uses it for a better estimation of the transmitted signal, the receiver performs *soft-decoding*. Figure 2.14 represents the block diagram of a communication system using channel coding techniques [12].

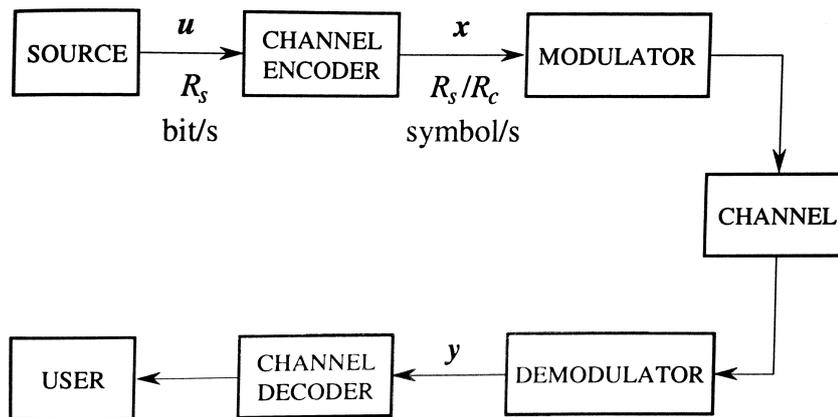


Figure 2.14: Block diagram of a transmission system employing channel coding.

2.4 Block and Convolutional Codes

When a source generates a *data stream* and inputs it to the channel encoder, the result is a *code stream*. For the generation of the code, we note that the data stream is segmented into pieces called *data words* and by the same way the code stream is segmented into *code words*. Then for a block code (n, k) , the data words contain k bits and the code words n bits. We also define as *code* the collection of code words, while *encoder* is the correspondence between data words and code words.

Finally, we say that a (n, k) code is a *block code* when the same k bits in the data stream corresponds to the same n code word bits. Furthermore, each data word is encoded independently of the other data words. But when there exist a dependence between data words we call it *tree code*. A tree code splits the data stream into segments named *data frames*. Tree codes with a special memory and linearity structure are called *convolutional codes*.

2.4.1 Linear Block Codes

A class of linear block code are the *Parity-check* codes, which adds to a sequence of k bits one additional bit at the end (last position). In this way, if an error occurred in the transmission and the parity of this bit has changed, the decoder detects it. Figure 2.15 shows the block diagram of a parity-check encoder [12].

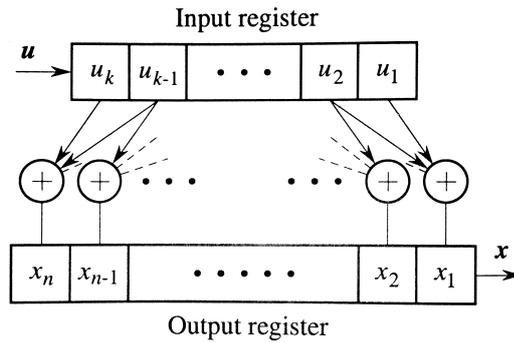


Figure 2.15: Block diagram of a the encoder for a parity-check code.

Other examples of linear block codes are the *Hamming Codes*, the *Hadamard Codes* and the *Golay Codes* [11]. Higher symbol alphabets are used in block codes in particular the widely used Reed-Solomon code. This code is effective against burst errors. Another technique for burst errors is the *interleaving* which reorders the bits following a pattern and the inverse process is done at the *de-interlaver* in the receiver.

2.4.2 Convolutional Codes

The deep difference of *convolutional codes* from block codes resides in their structure, analysis and design tools. In contrast to block codes, the convolutional generates an output which depends on previous states or messages. Then a block code is interpreted as a particular case of convolutional code (n, k, N) , with the constraint length $N = 1$. We depict in figure 2.16 an example of convolutional encoder [12].

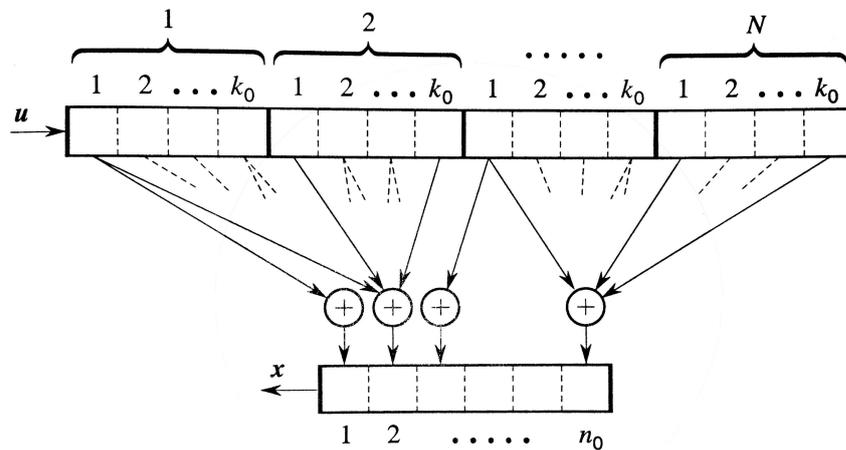


Figure 2.16: General block diagram of a convolutional encoder in serial form for a (n, k, N) code.

A shift register of length $N - 1$ feeds the input of the encoder continually. The decoding is easy which makes convolutional codes efficient. Since a convolutional code is described by a state diagram, the *Viterbi algorithm* is a sequence estimation technique which solves problems such as finding the shortest path and it has desirable computation properties. Let us note that finding the route or path on the trellis diagram is equivalent to the estimation of the message sequence at the decoder [10]. Thus, this recurrent process in which the results obtained previously are used in the next time instants make the algorithm computationally efficient.

2.5 Modulation Schemes

We classify the types of modulation techniques according to the carrier attribute, such as amplitude, phase, or frequency that is being modulated, the number of levels and the degree of the receiver to detect the unknown carrier phase. Although there are many combinations possible, we present in this section the ones most popular for wireless channels with additive white Gaussian noise (AWGN).

2.5.1 Quaternary Phase-shift keying (QPSK)

This modulation scheme shifts the phase of a carrier signal in function of the source symbols. Let us consider a sequence of K symbols represented by the signal

$$v_k(t) = \Re \left\{ \sum_{k=0}^{K-1} e^{j\phi_k} s(t - kT) e^{j2\pi f_0 t} \right\}, \quad 0 \leq t < KT \quad (2.14)$$

where each discrete phase ϕ_k takes values in the set

$$\left\{ \frac{2\pi}{M}(i - 1) + \Phi \right\}_{i=1}^M \quad (2.15)$$

with Φ an arbitrary constant phase and M the number of mapping symbols.

Assuming now $s(t)$ to be a rectangular pulse of amplitude A and duration T $u_T(t)$, we write

$$v_k(t) = A \sum_{k=0}^{K-1} u_T(t - kT) \cos(2\pi f_0 t + \phi_k) \quad (2.16)$$

$$= I(t) \cos(2\pi f_0 t) - Q(t) \sin(2\pi f_0 t) \quad (2.17)$$

where we define the *in-phase* and *quadrature* components respectively as follows

$$I(t) \triangleq A \sum_{k=0}^{K-1} \cos(\phi_k) u_T(t - kT) \quad (2.18)$$

$$Q(t) \triangleq A \sum_{k=0}^{K-1} \sin(\phi_k) u_T(t - kT). \quad (2.19)$$

For a quaternary phase-shift keying scheme (Q-PSK) [12], we depict its geometrical representation in Figure 2.17, the modulator in Figure 2.18 and demodulator in Figure 2.19.

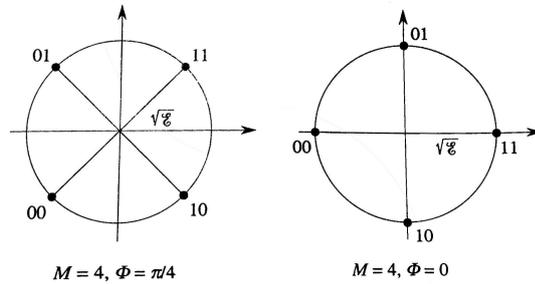


Figure 2.17: Geometrical representation of Gray-mapped QPSK signal.

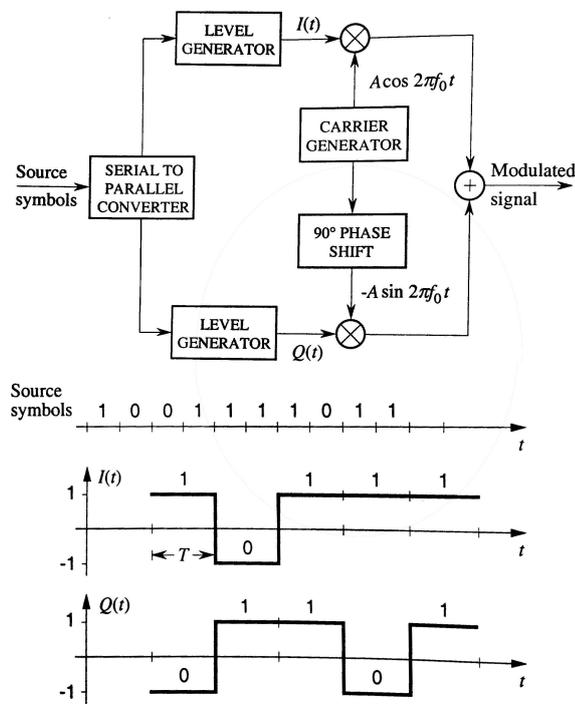


Figure 2.18: Quaternary PSK modulator.

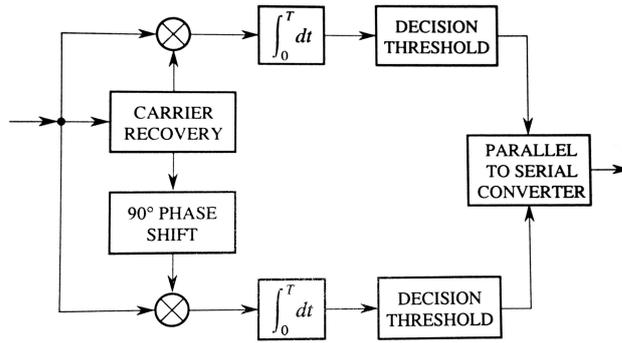


Figure 2.19: Quaternary PSK demodulator.

In order to evaluate the performance of a QPSK modulation over an AWGN channel, Figure 2.20 represents the theoretical error probability as a function of the Signal-to-Noise ratio (SNR) for different values of M .

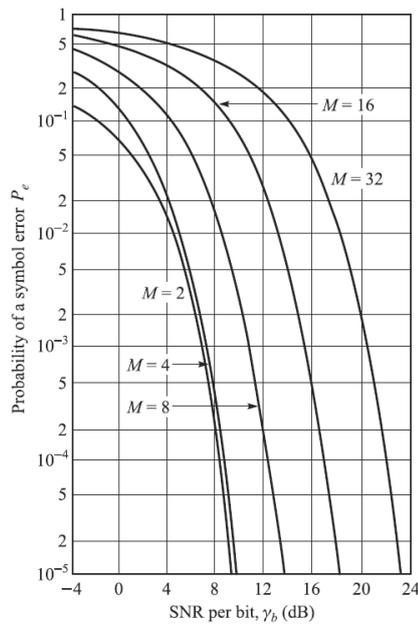


Figure 2.20: Probability of symbol error for a PSK signal [11].

2.5.2 Quadrature amplitude modulation (QAM)

For this modulation scheme the source symbols not only determine the phase of the carrier signal, but also the amplitude. Hence, the equation is written following the QPSK steps as

$$v_k(t) = \Re \left\{ \sum_{k=0}^{K-1} A_k e^{j\phi_k} s(t - kT) e^{j2\pi f_0 t} \right\}, \quad 0 \leq t < KT \quad (2.20)$$

and $s(t) = u_T(t)$ then

$$= A \sum_{k=0}^{K-1} \{I \cos(2\pi f_0 t) - Q \sin(2\pi f_0 t)\} u_T(t - kT) \quad (2.21)$$

Two signals are said to be in *quadrature* when they are 90 degrees apart in phase. By convention, the in-phase component is the cosine wave and the quadrature component the sine wave.

There are several QAM families depending on the number of constellation symbols or possible transmitted waveforms (M), as Figure 2.21 shows.

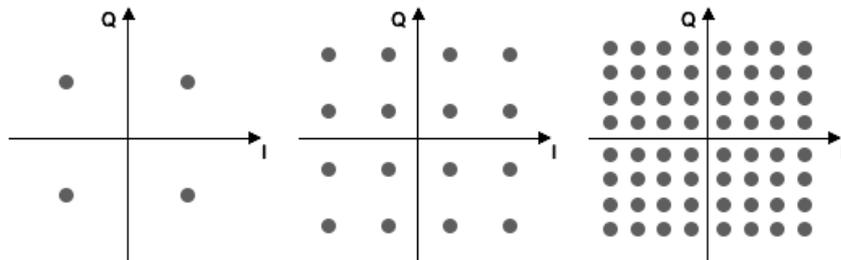


Figure 2.21: M -QAM modulation constellations for $M = 4, 16, 64$ in respective order.

The digital information is used to control the I and Q values, which are added together with its respective in quadrature phase. That is the basic implementation of a digital QAM modulator, illustrated in Figure 2.22.

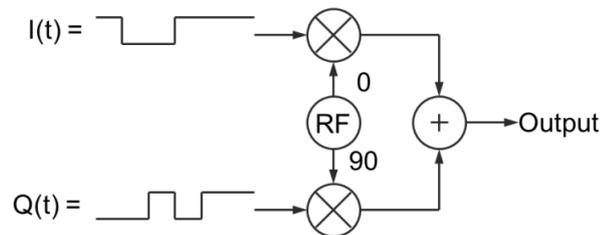


Figure 2.22: Block diagram of a QAM modulator.

The inverse process is done to demodulate the signal and obtain the I/Q components, see Figure 2.23.

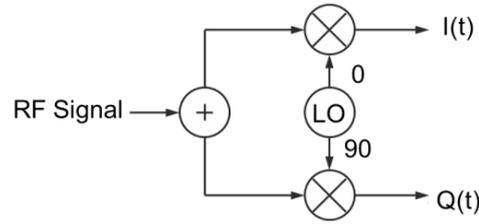


Figure 2.23: Block diagram of a QAM demodulator.

In communications systems the allowable power is determined by the maximum value of the signal at the input of the amplifier without saturating it, also called maximum peak power. These limitations are decisive in order to find the best modulation that ensures the lowest error probability if the signal is transmitted over an AWGN channel [10]. Another decisive factor is the sensitivity of the receiver performance.

We represent in the Figure 2.24 the theoretical error probability of M -QAM modulations over a AWNG channel.

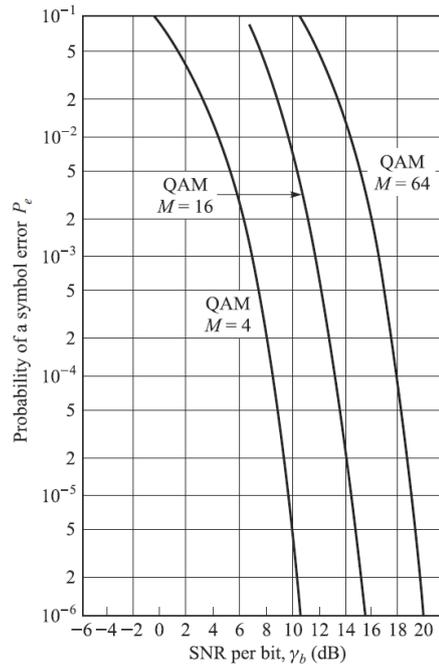


Figure 2.24: Probability of symbol error for a QAM signal [11].

From Figure 2.20 and Figure 2.24 we observe that QAM is more power efficient in comparison to PSK, however the PSK modulation preserves constant its envelope, which means that the constellation points are equidistant to the origin.

2.6 Multi-Carrier System Fundamentals

Among many digital transmission techniques, the Frequency Division Multiplexing (FDM) method splits the spectrum into completely separated individual subcarriers. That allow us to transmit data over separate carrier signals. The individual subcarriers have narrow bandwidth (BW), but the complete signal have a broad bandwidth.

The advantages of multi-carrier modulations are specially associated to wireless channels, where the fading caused by multipath propagation has a strong effect on the transmitted signal. Therefore, narrow BW subcarriers offer more immunity and robustness to interferences. However, the signal bandwidth can be very large if the number of subcarriers to be sent are high. A solution for that is to use *Orthogonal Frequency Division Multiplexing* (OFDM) modulation.

2.6.1 OFDM

As a derivation of the FDM systems, the OFDM modulation splits the spectrum into densely spaced subcarriers with overlapping spectra. The reason for that is the mutual orthogonality between the time-domain waveforms which are generated using a *Fast Fourier Transform* (FFT) at the transmitter and receiver. Although the overlapping on spectra, OFDM is significantly less sensitive to inter-symbol interference, because the set of signals used are orthogonal with each other. Each symbol occupies a frequency-time window with little distortion of the waveform.

We show in Figure 2.25 the OFDM signal spectrum, which consist of many bits in parallel. Let us note that rectangular pulses in the time domain produce *sinc*-functions in frequency domain.

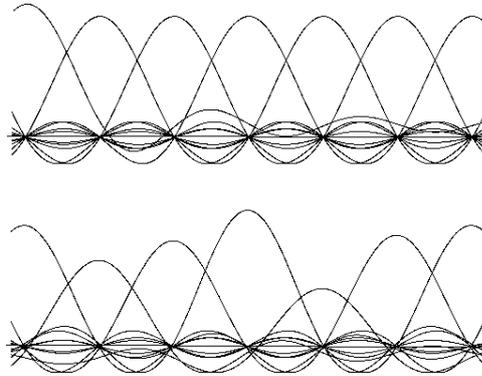


Figure 2.25: (Above) signal spectrum transmitted. (Below) signal received over a dispersive, time-variant channel.

Let us suppose that $C(f)$ is the frequency response of a band-limited channel with bandwidth W . We split W into $N = W\Delta f$ sub-bands of relatively narrow width Δf [11]. We illustrate this subdivision in Figure 2.26.

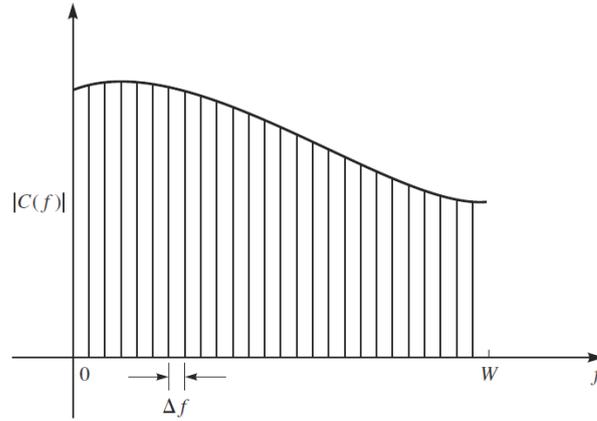


Figure 2.26: Narrowband subchannels of width Δf of the channel bandwidth W .

Then, we associate a sinusoidal carrier signal to each subband with the form

$$s_k(t) = \cos 2\pi f_k t, \quad k = 0, 1, \dots, N - 1 \quad (2.22)$$

where f_k is the mid-frequency in the k -th sub-band. We define a symbol rate $1/T$ to be equal to the frequency separation Δf . With orthogonal subcarriers over the symbol interval T , we define

$$\int_0^T \cos(2\pi f_k t + \phi_k) \cos(2\pi f_j t + \phi_j) dt = 0 \quad (2.23)$$

where $f_k - f_j = n/T$, $n = 1, 2, \dots, N - 1$, independent of the phases ϕ_k and ϕ_j . As a result, the Equation 2.23 represents OFDM signals, which are modulated for transmitting our data information.

Through the selection of a large N , the symbol interval of the OFDM system $T = NT_s$ increases, and if it is larger than the time duration of the channel-time dispersion, it reduces the inter-symbol interference.

The demodulation of the received signal is accomplished by cross-correlators and matched filters in each subchannel [11]. Therefore, for N subchannels the demodulator requires a parallel bank of $2N$ cross-correlators and $2N$ matched filters. Furthermore, the modulation process of an OFDM signal is implemented with $2N$ filters with symbols from a M -QAM signal constellation. The computation with $2N$ parallel filters that generate the modulated signal are done with the discrete Fourier transform (DFT) and its inverse (IDFT). For a more efficient computational performance, the DFT is implemented with the Fast Fourier Transform (FFT) algorithm and IDFT with the respective IFFT.

2.6.2 IFFT and FFT

A simply efficient computational algorithm of a multicarrier communication system is the fast Fourier transform, whose block diagram is shown in the Figure 2.27.

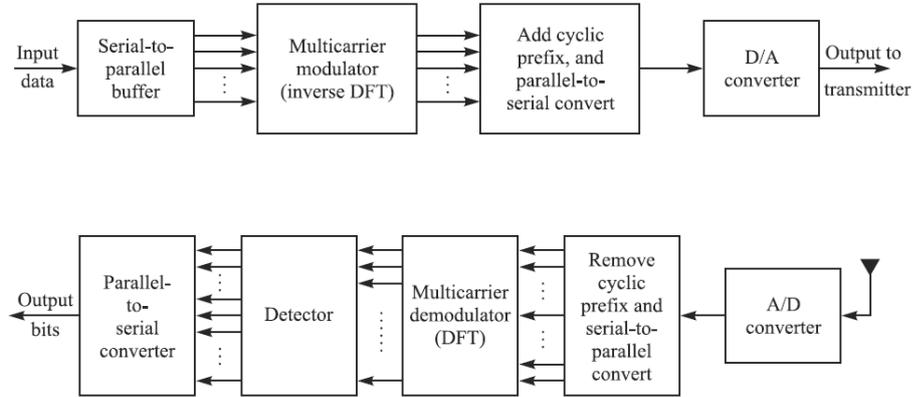


Figure 2.27: Multicarrier communication system [11].

As described for the frequency division multiplexing modulation (FDM), the data is split into parts which are multiplied with subcarrier (sinusoidal signals). Then, they are added together generating the OFDM symbols. This is equivalent to perform the IFFT, which spares us the individual multipliers, as shown in Figure 2.28.

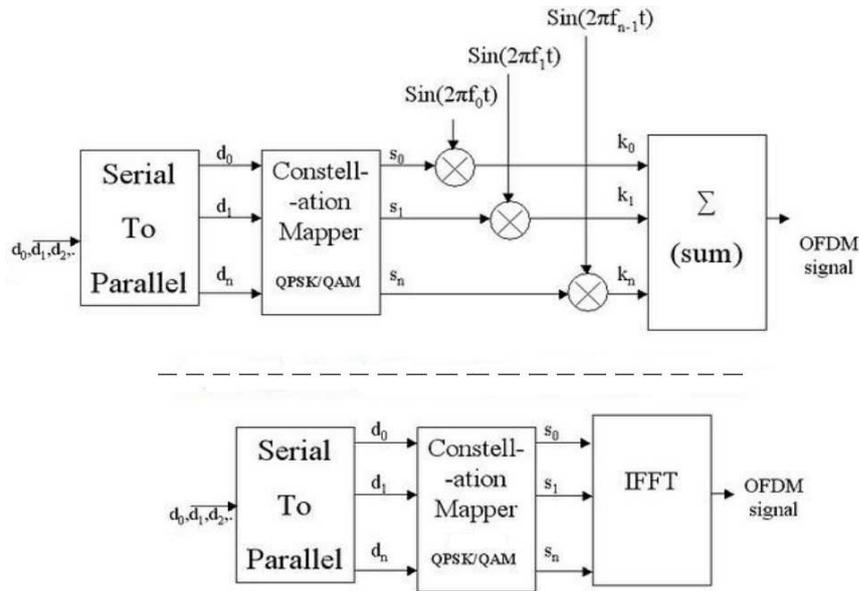


Figure 2.28: Digital implementation of OFDM system using IFFT.

That is the key feature of a OFDM communication system. While the OFDM technique uses multipliers for each sinusoidal subcarrier as shown in equation 2.24, the Fourier transform separates the signal into different frequency sub-channels by multiplying the signals with a series of sinusoids, see equation 2.25 for the FFT and equation 2.26 for the IFFT.

Let us consider the general representation of an OFDM signal as

$$\text{OFDM Signal} = c(t) = \sum_{n=0}^{N-1} s_n(t) \sin(2\pi f_n t) \quad (2.24)$$

where $s(t)$ are the symbols mapped to a chosen constellation (QPSK/QAM etc...) and f_n orthogonal frequency subcarriers.

The FFT is represented as

$$X(k) = \sum_{n=0}^{N-1} x(n) \sin\left(\frac{2\pi kn}{N}\right) + j \sum_{n=0}^{N-1} x(n) \cos\left(\frac{2\pi kn}{N}\right) \quad (2.25)$$

and the IFFT as

$$x(n) = \sum_{k=0}^{N-1} X(k) \sin\left(\frac{2\pi kn}{N}\right) - j \sum_{k=0}^{N-1} X(k) \cos\left(\frac{2\pi kn}{N}\right). \quad (2.26)$$

Both equations perform the same operation although the FFT performs a conversion from the time domain to the frequency domain, while the IFFT the inverse process. Since our OFDM signal $c(t)$ is in the time domain, we use the IFFT in the transmitter which generates the time-domain OFDM signal. The entire block diagram of a basic OFDM system for the transmitter and receiver part is illustrated in Figure 2.29.

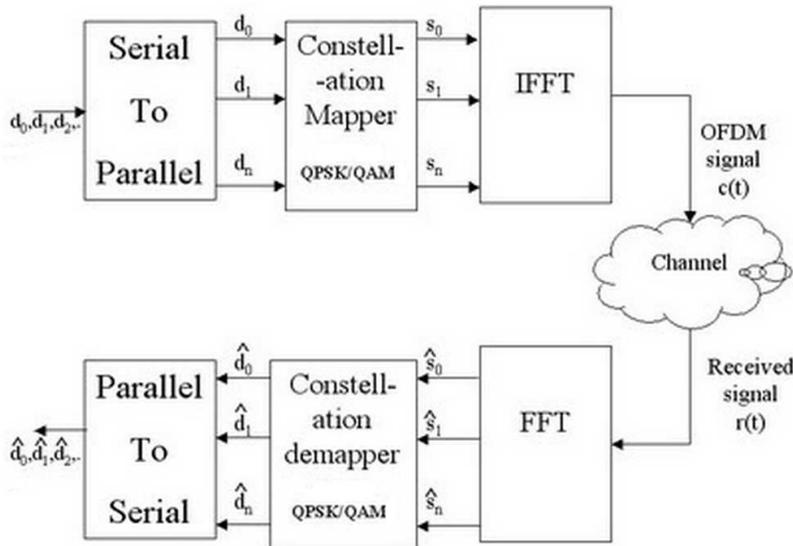


Figure 2.29: Complete digital OFDM system using Fourier Transform.

In an OFDM system not all the subcarriers are used for data transmission. Commonly, some subcarriers are reserved for pilot carriers, used for channel estimation and equalization, and as guard band (null subcarriers), which eases the requirement on the transmitter front-end filters.

2.6.3 Cyclic Extension/Prefix (CP)

Apart from the guard band, adding a cyclic prefix (CP) to our OFDM symbols is of great benefit for wireless communications. First of all, it mitigates the inter-symbol interference (ISI) problems caused by the delay dispersion generated in the propagation channel. Secondly, it helps to reduce the inter-carrier interference (ICI) due to the non-orthogonality of the received signal. Finally, the cyclic prefix convert linear convolution in to a circular convolution which eases the detecting process at the receiver. Figure 2.30 illustrates the adding prefix process for N values of a single OFDM symbol.

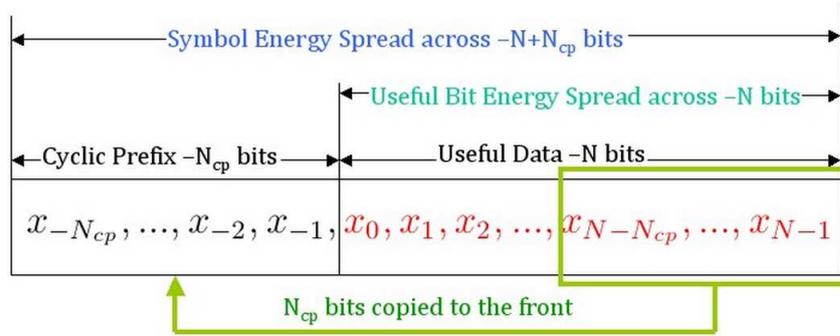


Figure 2.30: OFDM symbol with a cyclic prefix of length N_{cp} .

Since the circular convolution in the time domain translates to multiplication in the frequency domain, we describe the received signal in the time domain as

$$r = h \otimes x \quad (2.27)$$

where h is the channel impulse response and x the OFDM symbol with cyclic prefix. Then, in the frequency domain we obtain

$$R = HX. \quad (2.28)$$

In order to estimate the received signal R at the receiver, we just have to solve the equation 2.28. Moreover, the phase and amplitude is equalized using this process, which is represented as follows

$$\hat{X} = \frac{R}{H}. \quad (2.29)$$

We observe that by adding the cyclic prefix to our OFDM symbol and transmitting it through the channel, we obtain a circular convolution. The ratio of the channel delay spread to the OFDM symbol duration determines the relative length of the cyclic prefix. At the receiver, we extract the samples and perform the FFT, which converts our time-domain signal into the frequency domain. Using the circular convolution properties, we extract our OFDM signal equalized with reduced ISI and ICI.

2.7 Wireless Channel Characteristics

The channel characteristics of a wireless communication system has an important role regarding the implementation and design of the block components of the receiver. For instance, a band-limited channel determines the range of applicable carrier frequencies. Moreover, fluctuations of the amplitude or nonlinearity phase levels imply the necessity of equalization components at the receiver. Therefore, we present in this section the basic characteristics of channel models.

The received signal propagated over radio is made up not only from the signal that comes directly from the transmitter, but also the combination of reflected, diffracted and scattered copies of the original one. This effect is known as *multi-path propagation*.

Reflection occurs when the signal hits a surface and part of its energy is absorbed by the surface and the other one continues until it reaches the receiver antenna. It is possible and common that the signal has been reflected several times by more than one surface before it ends at the receiver. When the signal is derived into secondary ones, this phenomenon is called *diffraction*. The scattering occurs when rough surfaces spread the signal energy in all directions providing additional energy at the receiver [16]. We illustrate a typical wireless propagation scenario in Figure 2.31.

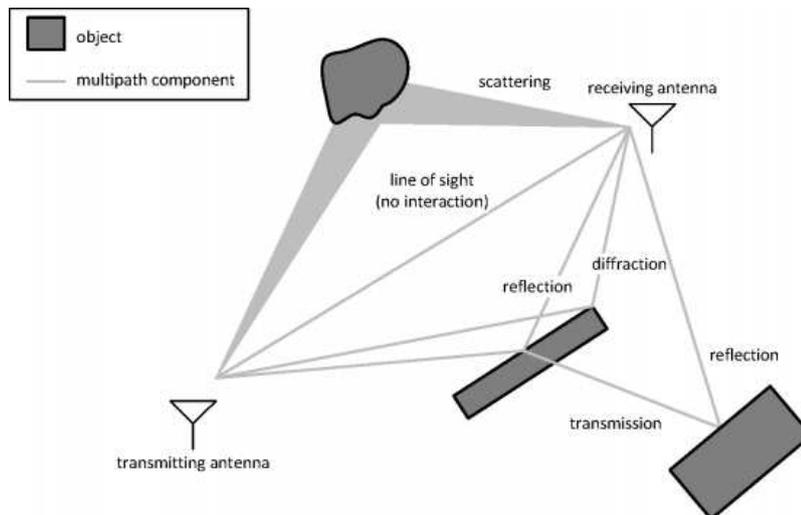


Figure 2.31: Wireless Propagation multi paths [Source: Universiteit Gent].

In a radio channel where the air is the medium, the direct path between the transmitter and receiver diminishes the power of the signal by a factor proportional to the square carrier frequency and square of the distance. This effect which dissipates the power in the free space is called *Path Loss*. On the other hand, when the medium includes obstructions the variations on the signal are random. This behavior is called *Shadowing*. Figure 2.32 represents the multipath propagation components between a base station (BS) and mobile unit.

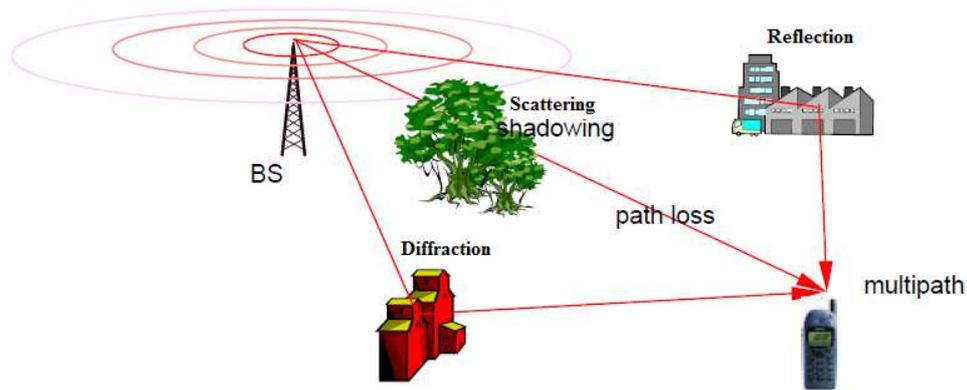


Figure 2.32: Multipath scenario of radio propagation [Source: Intechopen].

Besides multipath propagation, if there is movement of the system the *Doppler effect* has also a negative influence on the signal. The movement of the mobile unit causes a frequency shift of each of the waves. That causes a frequency difference, named Doppler frequency, between the received signal and the transmitted. This change in frequency is called Doppler effect or Doppler shift. The measure of the maximum broadening of the spectrum due to the Doppler shift is called *Doppler Spread* [14].

Due to the movement and the path distances, the reflected parts arrive with a delay respect to the original signal. This effect is known as the *Delay Spread*. The *Coherence Bandwidth* quantifies the channel frequency response, which is inversely proportional to the delay spread.

2.7.1 Fading Channels

The time varying attenuations and delays that degrade the performance of a communication system define the channel model as *Fading Channels*. We have review in the previous paragraph some causes for these effects and its study is essential in order to design the countermeasures against them, such as diversity and coding [15].

We classify the fading channels in relation to the following definitions [16]:

Fast Fading. Fast fading occurs when the coherence time, which is the time period wherein two received signals have high amplitude correlation, is smaller than the symbol duration of the signal.

Slow Fading. When the shift in frequency due to Doppler is insignificant, the channel impulse response changes slowly. We define this channel as a slow fading channel.

Frequency-Flat. If the channel bandwidth is greater than the signal bandwidth, the channel is considered a frequency-flat fading channel. The main distortions caused by this fading are the random variations of the gain and frequency spectrum. That derive into deep fades and nulls due to the destructive interferences. This model is commonly use for mobile radio channels.

Frequency-Selective. On the other hand, if the channel bandwidth is smaller than the signal bandwidth, the channel becomes a frequency-selective fading channel. That leads to overlapping of the symbols in the time domain, known as ISI.

Rayleigh and Rice Reference fading Models. If there is no direct path between transmitter and receiver, the flat fading channel is approximated by Rayleigh distribution, which is essentially the sum of two equal independent orthogonal Gaussian random variables. On the other hand, when there is a line of sight, the strongest component comes from the direct path and it fades deeper compared to the multipath components. This behavior is characteristic of a Ricean distribution [16].

2.8 Channel Estimation and Equalization over Inter-symbol Interference Channels

A wireless physical channel used for digital transmission cannot be modeled only with a non-distorting channel with additive white Gaussian noise (AWGN). As we presented in the previous section, the radio channels are characterized by time and frequency disruptive effects caused by the transmission medium and the design imperfections of filters and amplifiers of the communication system. Furthermore, multipath propagation channels generate several secondary signals that reach the receiver and cause echoes. In a flat fading model, these echoes produce overlaps in the receiver signal, also known as inter-symbol interference. Thus, it is important that under these conditions the communication system presents some mechanisms to mitigate these distortions. The part of the receiver that counteracts the ISI is called *equalizer* [10].

The channel characteristics is not always know at the beginning of the transmission or it can be time variant. Therefore, it is necessary to adapt our equalizer to these variations. In order to achieve that, the equalizer has to track the changes of the channel characteristics, a process that is called *Channel Estimation*. The design of receivers with channel estimation and adaptive equalization is known as *adaptive receivers* [15].

While tracking the channel response variations in time, the equalizer can adaptively compensate the distortions introduced in the signal. The adaptation process of a typical equalizer is done in two steps. First, the *training mode* which consist on transmitting a training sequence known by the transmitter and receiver in order to estimate the channel characteristics. Second, *equalizing mode* using the channel characteristics calculated from the received training sequence, which presents the distortions introduced by the channel. In the second step, the real data is received and equalized without channel estimation. Then, every time a training sequence is transmitted, the channel is estimated and the equalizer adapted.

There is also another approach where *pilots*, which are similar to training sequences, are inserted in the data stream at the transmitter. Then, the receiver extracts these pilots and uses them for channel estimation. These pilots are know for the transmitter and the receiver. Furthermore, each transmitted data sequence contains pilots which spare the use of an additional step, as in the training sequence. With this method, estimation is performed in time while the data is being transmitted simultaneously. There are defined standard pilot arrangements , which specify which positions in an OFDM symbols are reserved for pilots and which for data [16].

Figure 2.33 illustrates the pilot arrangements in time-frequency plane, while Figure 2.34 depicts the block diagram of an OFDM system based on pilot channel estimation and equalization.

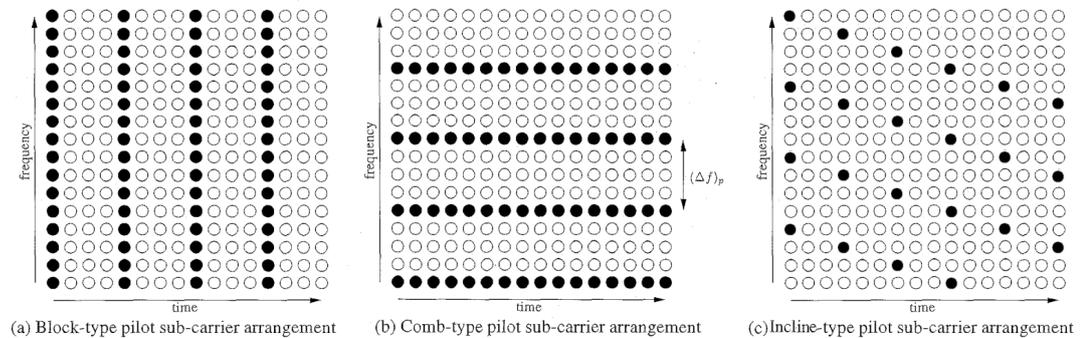


Figure 2.33: Pilot positioning in time and frequency.

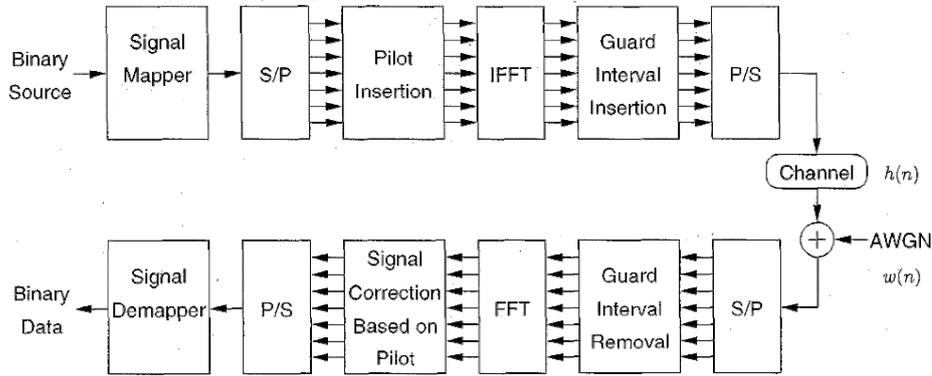


Figure 2.34: Baseband model of a pilot-based OFDM system.

After the pilot extraction, by using time and frequency interpolation the receiver calculates the impulse response and frequency response of the entire channel. Then, the equalizer uses the channel characteristics to compensate the distortions on the data stream and the equalized data is fed in the detector.

The detection of the original bits is done at the receiver applying coherent or non-coherent detection. Coherent detection uses reference values as explained with the training sequence or pilots. Non-coherent detection uses no reference values but differential modulation where the data information is transmitted in difference of two successive symbols [16]. The receiver uses two adjacent symbols in frequency or in time to perform the comparison.

2.9 Diversity Techniques

We have seen that the channel model has an important relevance for the communication system design. The presence of fading on a wireless channel affects the received signal power and consequently the error probability over an AWGN channel. To mitigate fading, an effective technique called *diversity* is used particularly over radio channels.

The idea is to send to the receiver several replicas of the same signal over separate fading channels. These replicas are affected by the same channel but have independent SNRs. If the number of replicas is high enough, there exist a high probability that one of these replicas reaches the channel without a deep fading on its signal power. This multiple transmission over diversity branches mitigates the fading and assures the minimum SNR required at the receiver [16].

The most important techniques are categorized as follows:

Space diversity. Using separate antennas far enough to obtain sufficient decorrelation, the transmitted signal is received in each of the antennas.

Polarization diversity. This case consist of two cross-polarized antennas in the receiver that obtain the orthogonal polarized signals transmitted. Due to orthogonality, the receiver antennas do not need a large physical separation.

Frequency diversity. Different frequency carriers are used simultaneously to send the same signal. This technique decreases the bandwidth efficiency.

Time diversity. The same signal is transmitted at different times, using time slots separated by a defined interval. This method introduces large delays which are not suitable for mobile radio systems.

2.9.1 Spatial Diversity and MIMO

In mobile radio communications, space or spatial diversity increases the capacity and performance of the communication link by adding multiple antennas at the receiver and the transmitter. This configuration is generally called a *multiple-input multiple-output (MIMO)* system and the spatial channel is named *MIMO channel*. The case where only one antenna is used at the transmitter and receiver is called *single-input single-output (SISO)* system, with its respective *SISO channel*. We also derive to advanced configurations such as *SIMO* or *MISO* systems.

Figure 2.35 shows a generic MIMO channel model, and Figure 2.36 illustrates a MIMO OFDM system.

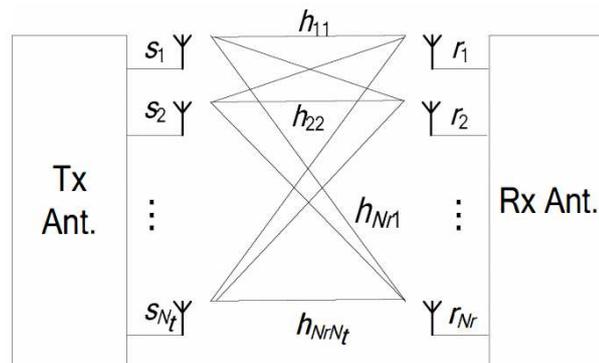


Figure 2.35: MIMO Channel model.

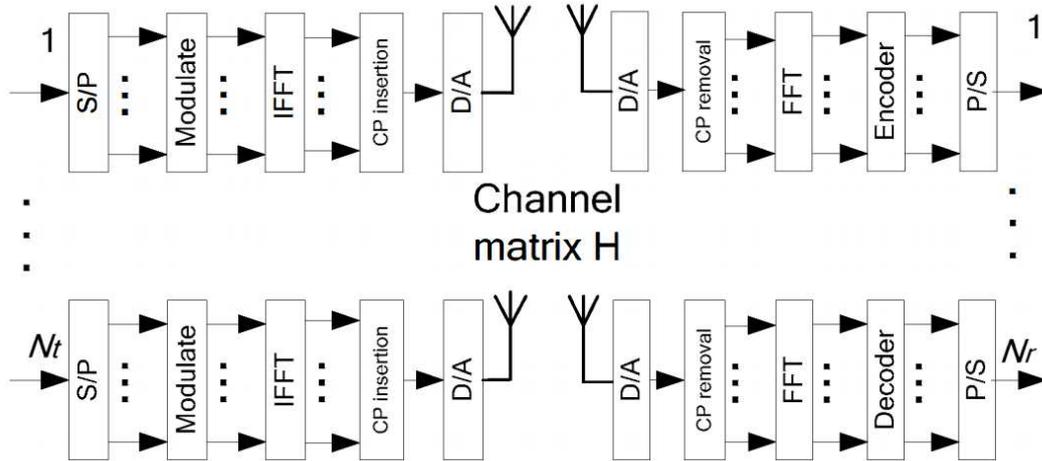


Figure 2.36: Block diagram of a MIMO OFDM System.

In order to understand the channel capacity improvement of a MIMO system, let us consider the ideal case of channels mutually orthogonal and without interference between channels. For a channel model of equal number of transmitter and receiver antennas $N_r = N_t = n$ the final expression of the channel capacity for a MIMO system is described by

$$C = n W \log \left(1 + \frac{P}{\sigma^2} \right) \quad (2.30)$$

where W is the channel bandwidth, P the total power of the transmitted signal and σ^2 the noise power. The most important deduction of this formula is that the channel capacity depends linearly on the number n of antennas. This space diversity technique is used for the design of high capacity systems. However, we presented in equation 2.30 an idealized case while in practice there is dependence between particular channels and they are not completely orthogonal.

Although MIMO systems offer a high capacity, its design complexity makes its implementation difficult. One of these inconveniences is the channel estimation and channel equalization in MIMO-OFDM systems, where its computational complexity is proportional to the number of antennas.

The channel estimation becomes crucial, since the signal transmitted by the other transmitter acts as interference. One procedure to deal with that is to use pilot tones orthogonal in time. That means that every time a pilot tone is inserted in a subcarrier, all the other transmitters do not send anything in that subcarrier [16]. MIMO channel equalization is implemented by using *linear equalizers* or *decision feedback equalizers*. However, they are computationally very complicated and require many calculations. For that reason, another solution commonly used in modern wireless systems is the multicarrier MIMO transmission. As the subcarriers remain orthogonal, we equalize them using independent blocks, as we illustrate in the Figure 2.37 [10].

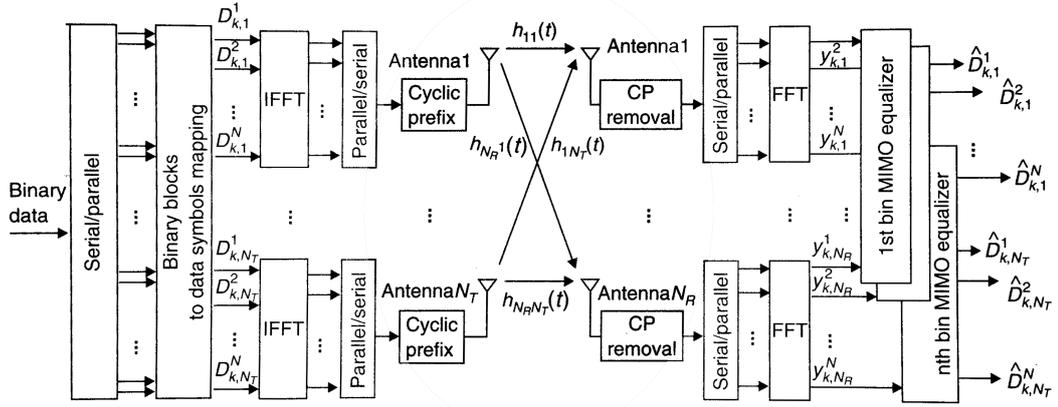


Figure 2.37: MIMO-OFDM system with the equalizer.

2.9.2 Maximal Ratio Combining (MRC)

Considering that the receiver is provided with multiples copies of the transmitted signal, we assume that in the channel these copies experience uncorrelated fading. In this case, the probability that all the copies fade simultaneously is considerably reduced with respect to the probability that a single copy experiences a fade. We write this relationship as $P_e \propto \frac{1}{SNR^L}$, where P_e is related to the BER and L the diversity of Lth order.

Being able to transmit or receive multiple signals such as in the SIMO channels, at the receiver side we reconstruct our original signal by combining the received copies. We combine them coherently with equal gain (EGC), or we weight the signal copies and then combine them coherently, which is known as *Maximal Ratio Combining* (MRC) and achieves the best SNR.

Let us consider a 1xN SIMO channel with N receiver antennas written as:

$$\begin{bmatrix} y_1 \\ y_2 \\ \cdot \\ \cdot \\ y_N \end{bmatrix} = \begin{bmatrix} h_1 \\ h_2 \\ \cdot \\ \cdot \\ h_N \end{bmatrix} x + \begin{bmatrix} n_1 \\ n_2 \\ \cdot \\ \cdot \\ n_N \end{bmatrix} \quad (2.31)$$

where $\{y_1, y_2, \dots, y_N\}$ are the received signals, $\{h_1, h_2, \dots, h_N\}$ the uncorrelated path channel, x the transmitted signal and $\{n_1, n_2, \dots, n_N\}$ the independent AWGN. Then, we obtain our reconstructed signal by following the maximal-ratio-combining (MRC) method written as:

$$\tilde{x} = \frac{h_1^* y_1 + h_2^* y_2 + \dots + h_N^* y_N}{|h_1|^2 + |h_2|^2 + \dots + |h_N|^2} \quad (2.32)$$

which means that the signal received in each antenna is weighted according to its channel path, such that the combination of all the antennas result in maximal ratio between signal and noise terms.

In Figure 2.38 we illustrate the receive diversity gain improvement for AWGN channel, which is the same obtained for the MRC diversity case. We depict in Figure 2.39 the SNR improvement using MRC, which exploits the receive diversity.

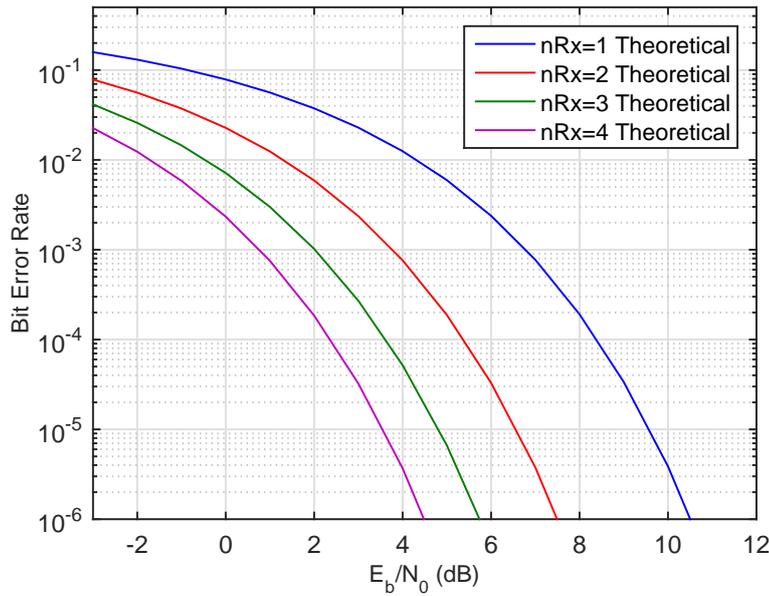


Figure 2.38: BER for QPSK modulation in AWGN with receive diversity.

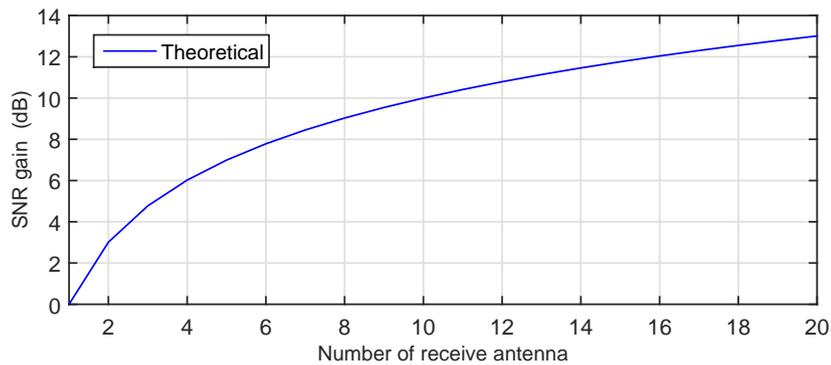


Figure 2.39: SNR improvement with Maximal Ratio Combining (MRC).

2.9.3 MIMO Channel Precoding Systems

Since the data rate demand is expected to grow exponentially, MIMO techniques have been adopted for LTE, WiMAX and WLAN in order to cope with this prevision. The main reason, as explained, is its ability to increase the maximum transmission rates. However, it also produced non desirable effect that deteriorates the transmission quality.

On the one hand, if we consider multiple antennas at the base station (BS) and the user equipment (UE), we define this scenario as Single-User MIMO system (SU-MIMO). On the other hand, if we consider multiple antennas at the BS but a single antenna at the UE, but multiple users, we define this scenario as Multiple-User MIMO (MU-MIMO). In the case of a SU-MIMO perspective, we exposed that the channel capacity increased with the number of antennas. However, as our communication link is based on a wireless channel where the UE is a mobile unit, the key factors for a mobile broadband communication have to take into account the power and computation limitations of the UE. Therefore, the MU-MIMO system is categorized as the most expected scenario.

Having multiple users connected to the same station implies a shared BW, time and transmit power. These factors are decisive for the system performance. Therefore, in order to enhance this communication link without increasing the receiver complexity, the space diversity offered in MIMO systems is exploited to estimate the channel characteristics, namely *channel state information* (CSI). Then, the receivers transmit the CSI to the transmitter (BS), which uses the channel information to pre-equalize the transmitted signal in order to make it more robust to the channel distortions. This pre-equalizing stage at the transmitter side is called *Precoding*.

There exist several precoding techniques depending on the quantity of CSI that the transmitter knows. This CSI at the transmitter depends on the channel estimation capacity of the receivers and also their ability to transmit it to the BS (feedback). Then, we assume that the fading channel remains unchanged for a complete frame transmission. Hence, the transmitter with the channel matrix performs equalizing and interference cancellation. For example, the Zero-Forcing (ZF) technique set the signal interferences to 0 by applying the inverse of the channel matrix to the signal.

Another feature of the precoding, is its beamforming ability. It changes the directivity of the antenna by shaping the channel input and output in order to obtain a more suitable channel. Figure 2.40 illustrates the beamforming precoding concept.

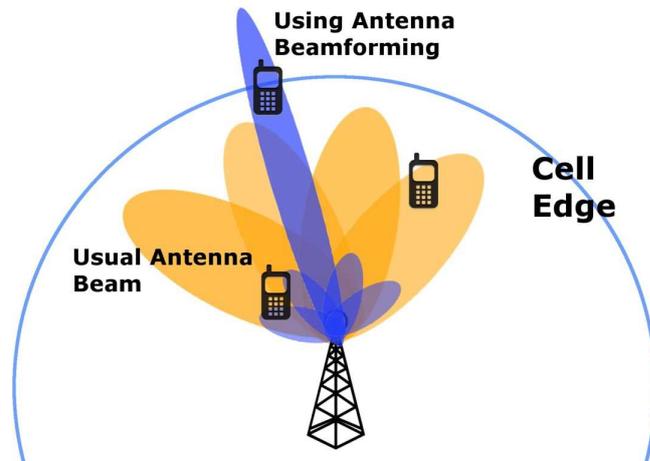


Figure 2.40: Beamforming precoding scheme of an usual antenna beam.

Knowing the channel matrix, it is possible using the *Single Value Decomposition* (SVD) to decompose it into two matrices. One of the matrices is multiplied at the transmitter with the signal (data). That process is called precoding. Then, at the receiver side, the other derived matrix is multiplied by the received signal, namely shapping. As a result, we obtain an independent channel with increased performance.

To conclude, we justify the benefits of precoding techniques by stating that by means of modifications of the transmitted signal (precoding) we reduce the performance loss caused by interference and channel fading. Furthermore, reduces the computation complexity at the receiver, saving power consumption and implementation cost at the UE. Moreover, the high computation capabilities of the BS permits to design and implement powerful precoding techniques at the transmitter side.

We describe in the next subsections a channel estimation methods based on pilots and also the codebook based precoding technique, which employs a set of precoding matrices and the channel estimation matrix to boost the network throughput.

MIMO-OFDM Pilot-aided Channel Estimation

At the receivers we use known symbols, called *pilots*, to estimate the channel. These pilots are inserted in the transmitter into specific time-frequency positions of our OFDM symbols. After the channel, the fading distortions can be recognized by comparing the received pilots with the transmitted ones. We note that transmitter and receiver know the pilots sequence. Furthermore, they are inserted into the symbol structure following a scheme, such as interleaved, which is called pilot pattern. This pattern has to be know also at the receivers. The design of the pilots pattern is also important, hence the *pilot overhead* is preferred low.

As a result, by comparing the transmitted symbols to the receiver over the different transmission paths of our MIMO system, we are able to characterize the channel and obtain its estimated matrix, which can be used for precoding.

Let us present a 2x2 MIMO system, with two transmitters sending pilots symbols over the fading channel and two receivers processing them in order to estimate the channel. We deduce that exist four propagation paths, from antenna Tx1 to antenna Rx1 and Rx2 (h_{11} , h_{21}), and from antenna Tx2 to antenna Rx1 and Rx2 (h_{12} , h_{22}), as we illustrate in Figure 2.41.

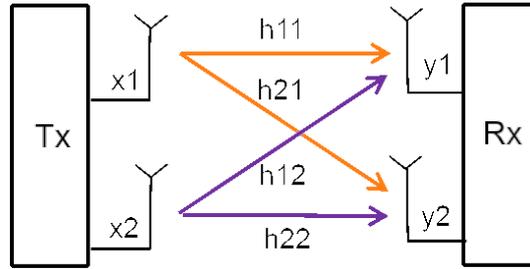


Figure 2.41: 2x2 MIMO channel propagation paths diagram.

Then, let us generate two independent pilots pattern P1 and P2. Both share the same known pilot sequence, such as CAZAC, but are inserted in different subcarriers locations in such a way that if we add them together we obtain a sequence of interleaved pilots, namely with zero valued subcarriers between them. We depict the pilots patterns in Figure 2.42.

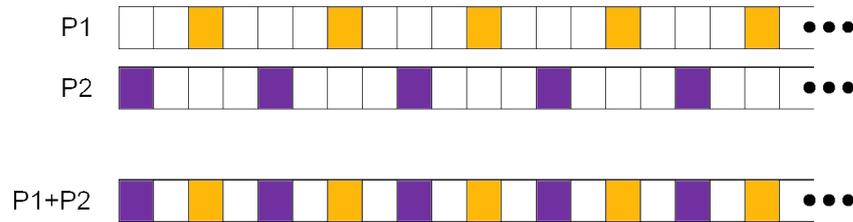


Figure 2.42: MIMO Interleaved Pilots Pattern.

A 2x2 MIMO transmission is written as:

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \end{bmatrix} \quad (2.33)$$

where y are our received signals, h the channel parameters, n the AWGN and x the transmitted signals.

Then, let us assume that our transmitted signals are the pilots pattern presented above, such as $x_1 = P_1$ and $x_2 = P_2$.

Since the pilots are interleaved in frequency among the subcarriers, we express each received signal as:

$$[y_1] = [h_{11}*p_1 + h_{12}*p_2] + [n_1] \quad (2.34)$$

$$[y_2] = [h_{21}*p_1 + h_{22}*p_2] + [n_2]. \quad (2.35)$$

Then, by calculating at the received signal $\frac{y_1}{p_1}$ and $\frac{y_1}{p_2}$ (the same for y_2), we obtain the respective $\tilde{h}_{11}, \tilde{h}_{12}, \tilde{h}_{21}, \tilde{h}_{22}$, which form the estimated channel matrix. we note that these parameters are calculated only where the pilots P1 and P2 where located in the received signal (P1+P2), as shown in Figure 2.42. Therefore, we assume that the interleaved channel estimation is afterwards interpolated in order to obtain the channel parameters for every subcarrier. We predict that for flat fading channels, the channel remain steady for all the subcarriers.

To summarize, by using pilot-aided channel estimation we obtain the CSI at the receiver. Then, selecting a precoding scheme we enhance our transmission performance protecting our transmitted signal from the channel distortions. For instance, for a frame made of 1 pilots symbol and 13 data symbols, we obtain the channel matrix from the pilots and assuming that the channel do not varies within the frame, we can also equalize the signal at the receivers, sending also a low feedback overhead of the CSI to the transmitter.

Codebook based Precoding

When the channel is unknown there are standard precoding matrices available, such as the standardized in LTE [17]. These matrices exploit the MIMO diversity boosting the network throughput. We employed the precoding matrices shown in Table 2.2 for our MIMO real-world testbed scenario.

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

Table 2.2: Codebook based Precoding Matrices

2.10 4G-LTE and Beyond-5G

It is predicted that due to the fast evolution of the technology regarding communication devices the worldwide subscribers will require high-speed packet access to the network. Furthermore, high-speed data transmission rates, low latency and greater spectral efficiency are the key elements of the future mobile communication systems.

However, every decade a new mobile standard is proposed and slowly implemented for the subscriber use. For instance, the most advanced techniques consist on orthogonal uplink multiple access, MIMO antennas technologies and inter-cell interference mitigation, among others. But the standard development of a new packet-optimized system referred to as *Long-Term Evolution* (LTE) is an aggressive countdown, where researchers and engineers try to publish their proposals. The standard development phase consist of an extensive evaluation of the sources and simulations of the proposed technologies, from its performance to its implementation complexity point of view. Finally, only the high-quality proposals and ideas are integrated into the commercialized standard [18].

We aim in this section to present the main characteristics of the 4G-Advanced LTE mobile communication systems. Then, move to some proposals and ideas for the 5th Generation (5G). Although most of the design concepts of the 4G standard will still be present on the 5G, the next generation will offer better performance on data-rate transmission, spectral efficiency and latency. That is necessary to cope with the exponential growth of data traffic in the coming years [4].

The Table 2.3 summarizes the related attributes of the LTE system [18].

Bandwidth	1.25-20 MHz	
Duplexing	FDD, TDD, half-duplex FDD	
Mobility	350 km/h	
Multiple access	Downlink	OFDMA
	Uplink	SC-FDMA
MIMO	Downlink	2x2, 4x2, 4x4
	Uplink	1x2, 1x4
Peak data rate in 20 Mhz	Downlink	173 and 326 Mb/s for 2x2 and 4x4 MIMO, respectively
	Uplink	86 Mb/s with 1x2 antenna configuration
Modulation	QPSK, 16-QAM and 64-QAM	
Channel coding	Turbo code	
Other techniques	Channel sensitive scheduling, link adaptation, power control, ICIC and hybrid ARQ	

Table 2.3: LTE system attributes.

The radio-interface attributes for 5G are similar to the ones presented in the previous table. For example, TDD duplexing techniques, MIMO schemes and OFDM modulations are the basis of the next ambitious mobile broadband generation. Therefore, we present in this section a review of the current 4G system that will help us to understand the design requirements for the enhancement of the attributes for the following 5th generation.

Network architecture and protocols

In order to support seamless mobility and advanced QoS mechanism, the LTE system is based on a flat hierarchical network. This simplified architecture between nodes and user equipment (UE) offers low-latency connections, reliable data transmission and power savings. In contrast to the previous generation, in LTE the radio network controller has been suppressed and its functions are integrated into the nodes, namely evolved Node-B (eNB).

Downlink Access

The wideband code division multiple access (WCDMA) scheme is used in the 3G in a scenario where transmissions on different codes do not interfere with each other. However, due to multi-path propagation typical in cellular environments the non-orthogonality of the codes causes inter-user and/or inter-symbol interference (ISI). A solution to mitigate the interference is to use advanced receivers such as linear minimum mean square error (LMMSE) receiver. However, high-data rates increase the complexity of the LMMSE receivers and a new access scheme is necessary in the LTE downlink.

As explained in the previous section, the OFDM scheme offers protection against ISI and improves the spectral efficiency. Therefore, employing OFDM in the LTE systems we are able to eliminate multi-path interferences, which degrades the performance of the wireless system considerably, particularly the achievable peak data rate and system capacity.

Uplink Access

The design requirements for the uplink access is more complex than for the downlink. The intra-cell interference limits the achievable data rates and capacity, while the codes are non-orthogonal even in the absence of multi-paths. In order to mitigate this issues and provide low signal peakiness, the two candidate schemes for orthogonal access are orthogonal frequency division multiple access (OFDMA) and single-carrier frequency division multiple access (SC-FDMA) [18].

Low signal peakiness is achieved with the Single-carrier FDMA scheme, where each data modulation symbol is spread out on all the subcarriers used by the UE. Furthermore, an advantage of SC-FDMA is that the UEs coverage is increased as the signal-peakiness is low. However, the additional DFT and IDFT operations at transmission and reception respectively increases the system complexity.

In contrast, the OFDMA scheme allocates to a user a fraction of the total bandwidth, so multiple users can transmit simultaneously using orthogonal subcarriers. While multiple users transmit simultaneously, orthogonal schemes such the OFDMA performs better as the number of the UEs increase.

Transmit Diversity and MIMO Spatial Multiplexing

In an OFDM system, the diversity techniques that can be used include time-diversity, frequency diversity, receive diversity and transmit diversity. In the LTE system frequency diversity is achieved by scheduling transmissions over distributed resources and time diversity is exploited by using hybrid ARQ. Two antennas configuration or more at the transmitter and/or receiver offers transmit and receive diversity, respectively. Examples of diversity techniques for the LTE and beyond mobile generations are cyclic delay diversity, block codes and multiple transmitter and receiver antennas [18].

Transmission diversity improves the link performance by enabling multiple data streams between the nodes and UE using MIMO spatial multiplexing. It also increases the capacity and throughput, as described in the previous sections. Therefore, LTE systems supports up to 4x4 MIMO in the downlink that increases the the peak data rates. On the other hand, multi-user MIMO operation in the uplink improves cell capacity but not UE peak data rates [18].

Cell Search and Reference Signals

In order to detect the cell identity and acquire time and frequency synchronization within a cell, the UEs perform cell search and reference signal acquisitions procedures. This is based on two downlink signals, the synchronization signals and broadcast control channel (BCH).

The sync signals enable symbol timing, frequency and cell identity detection of the downlink signal. The BCH signals carry system-specific information. Overall, the UEs obtain information about the cell ID, number of transmit antenna ports and cyclic prefix length, etc.

The reference signals are used for measure channel quality features such as scheduling, link adaptation and handoff, etc. The reference signals are categorized as cell-specific dedicated reference signals and UE-specific dedicated reference signals. All the downlink reference signals are multiplexed with data using hybrid TDM/FDM approach. In the uplink, reference signals are only defined for demodulation and channel sounding for channel quality measurements. Moreover, they are time-multiplexed with other transmissions in order to guarantee the single-carrier property [18].

We present two sequences used in LTE systems:

PN sequence. Its various purposes include scrambling of reference signals, scrambling of downlink and uplink data transmissions and generation of hopping sequences. Linear feedback shift registers (LFSR) can generate PN sequences.

Zadoff-Chu (ZC) sequences. They are used for primary synchronization signals, uplink reference signals, uplink physical control channel and random access channel [19]. A Zadoff-Chu sequence of length N_{ZC} is defined as:

$$x_u(m) = \begin{cases} e^{-j \frac{\pi u m^2}{N_{ZC}}} & \text{when } N_{ZC} \text{ is even} \\ e^{-j \frac{\pi u m(m+1)}{N_{ZC}}} & \text{when } N_{ZC} \text{ is odd} \end{cases} \quad m = 0, 1, \dots, (N_{ZC} - 1), \quad (2.36)$$

where u , the sequence index, is relatively prime to N_{ZC} [18]. For a fixed u , the ZC sequence has an ideal periodic auto-correlation property. For different u , ZC sequences are not orthogonal but present low cross-correlation.

A process known as *random access* is performed by the UEs when they turn on from sleep mode, perform hand-off or lose timing synchronization. When that happens, by a random access preamble the UE acquires first downlink timing synchronization. There are several random access preambles available. For example, the Zadoff-Chu sequences are used as preambles because their low signal peakiness as well as good correlation properties. The Doppler frequency shift, more pronounced at higher UE speeds, affects the detection. This problem is solved by using a set of cyclic shifts at expenses of reduction in the number of available random access sequences [18].

Channel Coding

Since the proposed channel coding schemes *Turbo Code* and *Low Density Parity Check* (LDPC) for the 4G-LTE were similar in performance, the decision leaned towards turbo codes because they were employed for the 3G systems. However, the turbo code is modified by changing its internal interleave with a contention-free interleaver which offers parallel processing. For more detailed description we suggest to consult some of the mentioned references such as [18].

Scheduling, Link Adaptation and Hybrid ARQ

In fast-fading environment typical of a wireless communication system, the LTE system employs some techniques to enhance its performance. These techniques are the channel-sensitive scheduling, link adaptation and hybrid automatic repeat-request (ARQ). First, the channel-sensitive scheduling, also referred to as multi-user diversity, selects the UE with better channel conditions on time, frequency and spatial resources. Then, link adaptation matches the modulation and coding, MIMO rank and precoding to the current channel conditions. Finally, the hybrid ARQ mechanism corrects possible errors in the link adaptation by adding incremental redundancy transmission, which makes use of acknowledgments and timeouts to achieve reliable data transmission.

Power Control

While low power signals derive into poor quality transmissions and increase of error rate, high power signals generate interference and extra battery consumption. Hence, the power control aims to adjust efficiently the power between UEs and nodes. Furthermore, power control schemes compensate path-losses and shadowing. For the LTE uplink, the power control algorithm is an event-based combined open-loop and closed-loop principle, which adjust the transmit power of the different physical channels. On the other hand, the downlink power control do not have an explicit feedback from the UEs, therefore its principle is based on a power allocation scheme.

A way to support inter-cell interference coordination (ICIC) is to allocate different power levels to different resource blocks used for data transmission. In the downlink, there are two available different power levels between OFDM symbols with reference signals and the ones with no reference signals. In the uplink, the two types of power adjustments are called accumulated and absolute power corrections.

Uplink and Downlink Control Signaling

There are multiples modes and formats to optimize performance in various deployment scenarios. The signaling related to the uplink feedback are the channel quality, MIMO channel rank, preferred precoding matrix and hybrid ARQ ACK/NACK. The channel quality information is required for the channel-sensitive scheduling and the link adaptation. The MIMO rank and precoding information determines the selection of the MIMO transmission format.

For hybrid ARQ operations the hybrid ARQ ACK/NACK is used.

On the other hand, the downlink control information consist of control format indication, uplink and downlink scheduling assignments and the ACK/NACK feedback for the uplink HARQ transmissions. The control information and the data are together time-multiplexed. The number of OFDM symbols used for control information are indicated in the control format indicator. With that information the resources are efficiently used while the OFDM symbols not used for control are employed for downlink data transmission. Time-multiplexing allows micro-sleep mode and lower latency, thanks to the decoding of the first OFDM control symbols. In absence of scheduling assignments, the UE turns off its transmitter and receiver, saving battery consumption and avoiding unnecessary interference. Furthermore, the low latency is achieved by the fact that the control information is decoded before the data decoding starts.

Inter-cell Interference Control

Nowadays, we identify two main development scenarios for a mobile UE. The first one is a rural scenario, where the communication performance between the UE and the cell nodes is determined by the AWGN. That is compensated with more power gain, more directivity in the antenna or a better coverage deployment. The second scenario is the urban, where apart from the AWGN the inter-cell interference (ICI) is a decisive factor. In this case, the use of more transmission power will increase proportionally the inter-cell interference.

The performance of the UE connectivity varies according to the distance to the main cell node and the interference of a neighbor cell node. While the UE moves away from the main node, its path loss increases worsening the signal. Furthermore, the more close to the edge of the main cell, apart from the maximum path loss value, the distance to the neighbor cell increases. That means that the interference power of the neighbor cell increases. All together form the particular scenario where it is specially necessary to employ inter-cell interference (ICI) mitigation techniques.

We classify the three types of ICI mitigation techniques as ICI randomization, ICI cancellation and ICI co-ordination or avoidance. The ICI randomization consist on interference suppression by randomization at the receiver due to processing gain. The ICI cancellation suppress the interferences by means of multiple antennas at the receiver, known also as *Interference Rejection Combining* (IRC). The ICI co-ordination or avoidance scheme applies certain restrictions on the resources used in different cells in a co-ordinated way. These restrictions are related to time-frequency resources or transmit power limits. We deduce that this interference avoidance techniques requires from inter-node communications, which is scheduled daily or in periods of seconds or hours.

Spatial Channel Model

One of the most important phases in the design and implementation of a wireless communication system is the study of the radio propagation, known as propagation channel model. The specifications and characteristics determine the performance of our communications and therefore a good understanding of the channel behavior will lead us to higher performances

at the least possible cost. Furthermore, the performance evaluation and comparison of competing wireless technologies can only be done when the propagation model is defined.

The multi-antenna technologies used for MIMO techniques are based on spatial correlations between them. However, the conventional propagation channel models were developed for single-antenna transmission scenarios assuming independent channels between each pair of transmit and receive antennas. Therefore, the *spatial channel model* (SCM) was developed. It provides a propagation platform for multi-antenna techniques evaluation.

The SCM is based on stochastic modeling of scatters which allows to model spatial correlations. The SCM has been widely used for MIMO system performance evaluation. The channel scenarios are described as *suburban macro-cell*, *urban macro-cell* and *urban micro-cell*. Depending on these scenarios, the system evaluation is characterized by parameters such as line-of-sight (LOS), path-loss model, number of paths and subpaths, delay and angle spread, etc.

5G

Although it is foreseen to introduce commercially the 5th Generation (5G) around 2020, we are still adapting the radio network for the 4G and most of the subscribers are using the 3G. Furthermore, the characteristics and novel techniques of the 5G are not fully defined for the coming standard. In spite of that, we are able to present some main key design elements and performance indicators of the standard beyond the 4G-LTE.

Based on the technology requirements to cope with the expected data traffic growth rates, peak data rates should reach values of 10 Gbit/s, in comparison to the 1 Gbit/s of the 4G. The Round Trip Time (RTT) should be in order of $1ms$ in contrast to the $5ms$ of the 4G. Moreover, the spectral efficiency should be at least two fold better than the 4G [1].

In order to achieve the specifications we describe some of the novel techniques that are currently being tested for performance evaluation [6]. Let us note that the 5th generation is focussed on boosting local area networks.

Frame structure. In order to achieve the RTT of $1ms$, our frame duration should be of $0.5ms$ with a dynamic UL/DL ratio, see Figure 2.43 [1].

Modulation. OFDM modulation format is being used for the 4G due its performance an lower complexity implementation. In spite of other alternatives such as filterbank multi-carrier modulation, the OFDM modulation is the most likable candidate for the 5G.

Link adaptation and Hybrid ARQ. As explained in the previous sections, link adaptation offers dynamic data rate which is suitable for channels with non-stable conditions. Furthermore, the power consumption is reduced and the resources are exploited more efficiently. Another important key element for a radio link design is the memory requirements which determine the size of the buffers, costly elements. Therefore, HARQ processes should

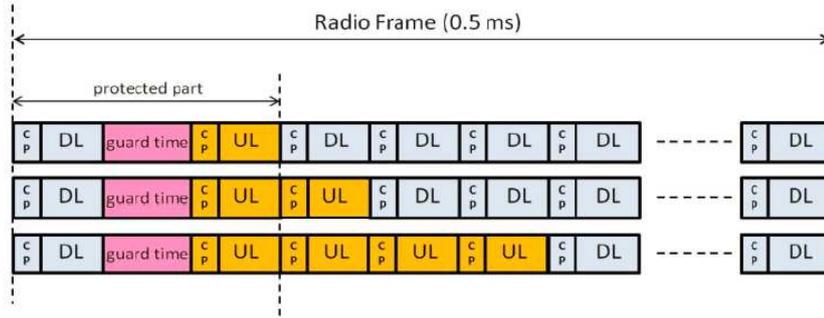


Figure 2.43: Radio frame structure.

be minimized in order to reduce buffers' size.

MIMO techniques. Due to the high data rates demand, MIMO technologies are one of the backbones of the future radio links. There are two approaches to improve the channel capacity. The first one is massive MIMO array antennas at the Base Stations. The second one is 4x4 single user or multiuser MIMO deployment in devices.

Interference mitigation techniques. Since the spectrum is a scarce resource, it is necessary to share it but in dense areas the interference levels lead to a poor network performance. Therefore, *Inter-cell Interference Coordination (ICIC)* is expected to improve considerably the network throughput. Moreover, advanced receivers allow us to use *Interference Rejection Combining (IRC)* or *Successive Interference Cancellation (SIC)* techniques, which help to suppress or cancel a limited number of interferers [1].

To summarize, we presented some of the requirements and novel techniques to cope with the data rate growth demands expected for the 5G. Moreover, we describe briefly some of the key design elements of the 4G and beyond. In order to define the standard of the 5G, it is of important relevance to evaluate and test the performance of such novel techniques within a simulated and real world testbed. We present in the next section the software tools that are capable of such simulations and tests.

Chapter 3

LabVIEW Design

In this chapter we describe our SDR design and implementation in LabVIEW. We also simulated its performance in contrast with the theoretical results. First of all, we developed a SISO system over a AWNG channel with Rayleigh fading profile. Then, we extended it to a 1x2 SIMO system and simulated its performance. A step forward was the design of a 2x2 MIMO system. In the last section, we describe the adaptation of the LabVIEW design for the USRPs. In particular, the transmitter and receiver parts were implemented separately, since they were connected to different USRPs. In order to do that, in the SISO-MIMO simulation design some modules were edited and new were added. Overall, in this chapter we present the main SDR LabVIEW development for our simulated and real-world testbed scenarios.

3.1 SISO System

In this section we present the SISO design blocks from the transmitter to the receiver. In order to verify the design we realized the transmitter-receiver modules in parallel. That means that the transmitter blocks were implemented with its correlated blocks at the receiver. For example, the encoder was only tested with its decoder at the receiver. Hence, we followed a linear design from the outer blocks to the inner ones of a digital communication system. The last modules, in the center, were the AWGN channel and fading profile. However, for descriptive purposes we present first our LabVIEW transmitter modules, then the channel model and finally the receiver modules. In addition, for the system performance evaluation we implemented the required blocks for BER/BLER calculations and its graphical SNR versus BER representation.

3.1.1 Model Design

We defined a frame structure made of 1 *Pilots* OFDM symbol and several *Data* OFDM symbols. In the frame structure, the first symbol was the pilot, which was used to estimate the channel at the receiver, and the consecutive symbols were designated for data.

We implemented the complete communication system as follows:

- A random bit sequence was generated, whose number of bits depended on the FFT size, modulation selected and number of symbols per frame.
- The complete bit stream was encoded with a defined code rate.
- The encoded bits stream was split into packets of bits, whose size depended on the number of bits per symbol. That was done inside a FOR loop.
- In each loop iteration, one OFDM symbol was generated. A packet of bits went from the channel encoder to the cyclic prefix adder block in each iteration. At the end of the FOR loop all the packets generated, known as OFDM symbols, were concatenated forming the frame structure.
- The pilots were inserted in an extra FOR loop iteration and concatenated at the first position of the OFDM frame structure.
- Then, the frame was fed into the channel, where a Rayleigh fading profile was applied and afterwards the AWGN added.
- The receiver had to split the frame into OFDM symbols. That was carried out by a FOR loop, which allowed us to process each symbol independently from the removal of the cyclic prefix to the channel decoding block.
- The pilots were extracted from its respective position in the frame structure with an extra loop iteration. Then, the channel was estimated.
- Each OFDM received symbol was equalized and after going through the receiver blocks was concatenated with the other processed symbols.
- Finally, the estimated encoded data bit stream was decoded and we obtained the estimated received bits.

This design offered faster computational performance. Figure 3.1 illustrates the flowchart of the LabVIEW implementation of the transmitter, while Figure 3.2 shows the flowchart of the implementation of the receiver in LabVIEW.

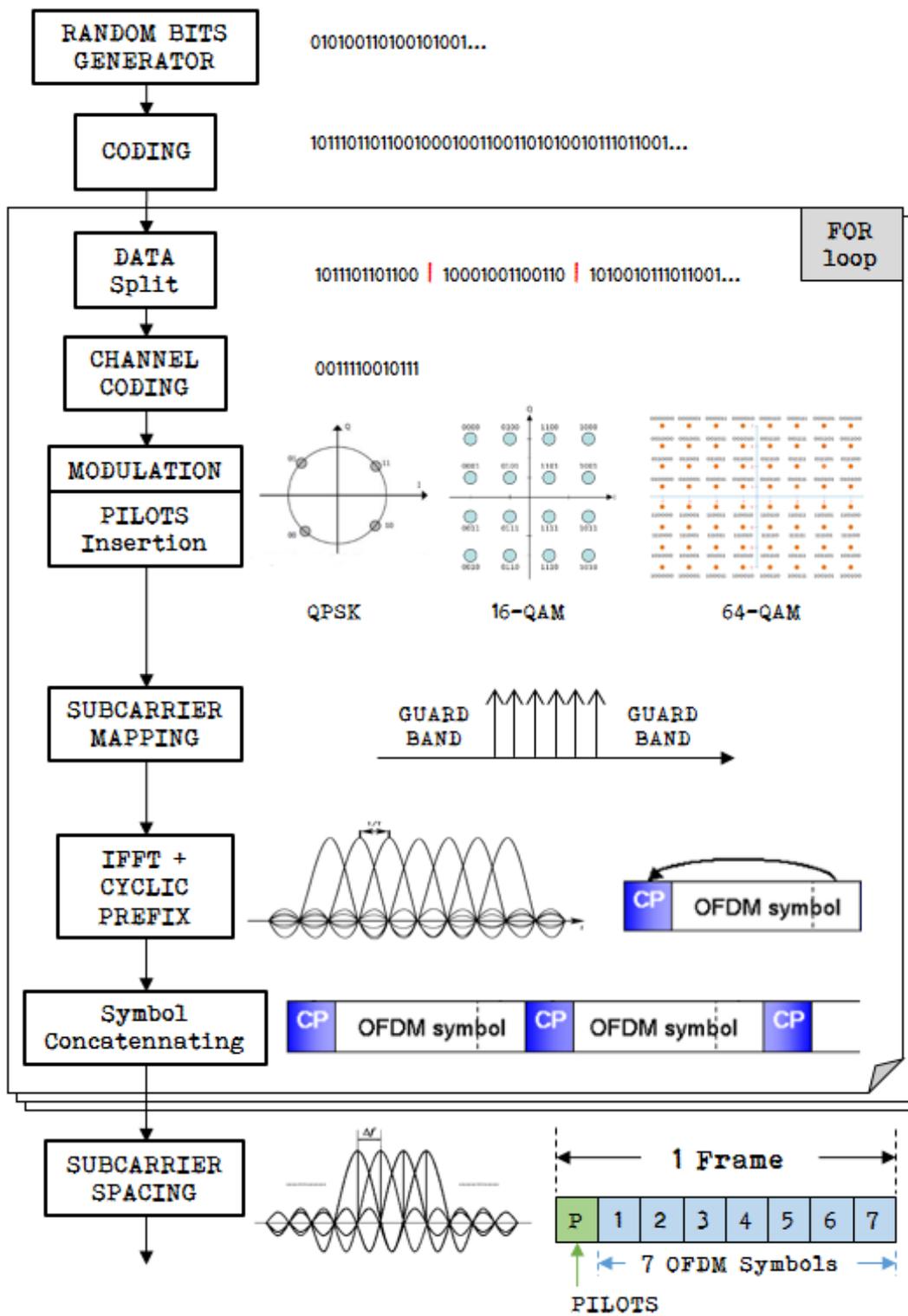


Figure 3.1: Flowchart of the transmitter stages in LabVIEW.

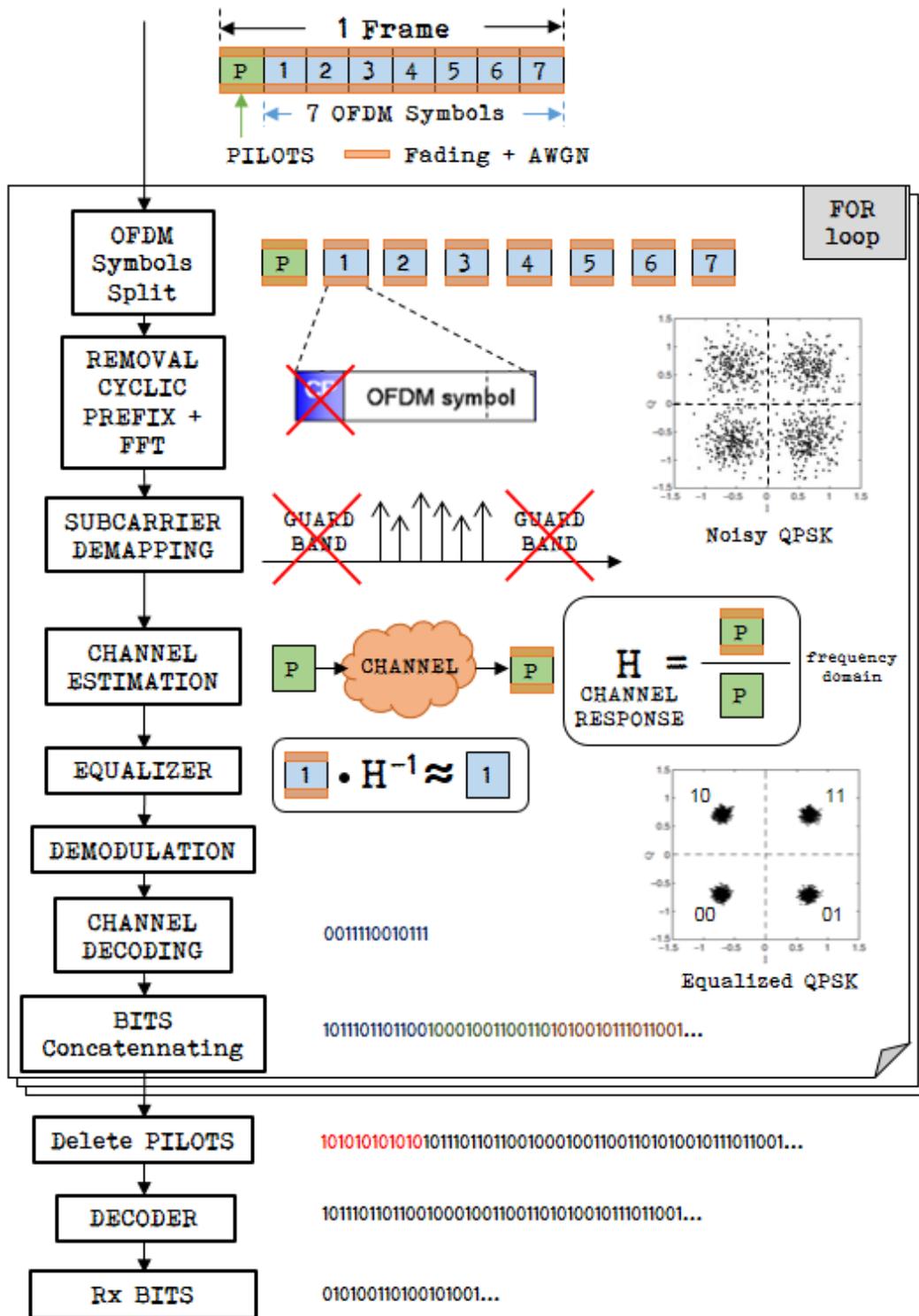


Figure 3.2: Flowchart of the receiver stages in LabVIEW.

As we present in the Figure 3.3, the frame structure determined the number of random bits that had to be generated. The number of OFDM symbols per frame determined the necessary FOR loop iterations to generate them, with an extra iteration for the pilot symbol insertion. The communication system ran for each frame structure once, while the FOR loops at the transmitter/receiver split the bit stream into packets that were used for the modulation/demodulation of the OFDM symbols.

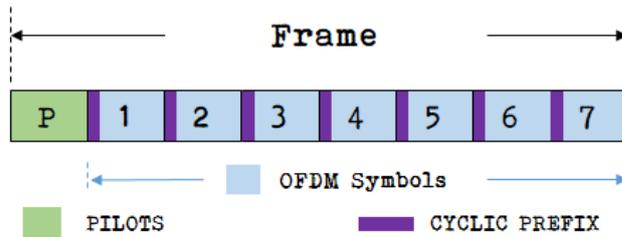


Figure 3.3: Frame structure of our OFDM communication system.

There were some parameters that defined the behavior of all the modules and LabVIEW elements (e.g. FOR loops), and also the desired modulation, subcarrier spacing, frame structure, etc. We named these parameters as *Global Parameters*, which are described in the next section.

Global Parameters

The Table 3.1 shows the global variables that defined the characteristics of our digital communication system within its LabVIEW design. Some of the parameters determined the behavior of the programming elements such as FOR loops. Furthermore, some variables were interconnected and thus its values bounded them. We show also this dependency.

Table 3.1: Global parameters of the LabVIEW design of a digital communication system.

Name	Default Value	Description	Dependency
FFT Size	512	Number of subcarriers per OFDM symbol	M-QAM, Number of Subcarriers, Number of bits per symbol, Total number of bits, Subcarrier occupation index, Sampling frequency, Cyclic prefix.
M-QAM	QPSK	Selected modulation (QPSK, 16-QAM, 64-QAM, 256-QAM)	Number of Subcarriers, Number of bits per symbol, Total number of bits.
Number of Symbols	8	Number of symbols within the frame.	Total number of bits, FOR loop iterations.
Position Pilots	1	Position of the Pilot symbol within the frame.	Channel Estimation.
Number of Subcarriers	300	Number of available subcarriers per OFDM symbol, since the rest is used as Guard Band.	FFT Size, M-QAM, Number of Subcarriers, Number of bits per symbol, Total number of bits, Subcarrier occupation index.
Number of Bits per Symbol	600	Number of bits transmitted per OFDM symbol.	FFT size, M-QAM, Code rate, Total number of bits.
Encoder Code Rate	1/2	Code Rate of the encoder.	Number of bits per symbol.
Total Number of Bits	4.200	Number of bits transmitted per Frame.	FFT size, M-QAM, Code rate, Number of bits per symbol.
Percentage Cyclic Prefix (%)	10	Amount of cyclic prefix added to the OFDM symbols.	FFT size.
Eb/N0 (dB)	0	Ratio of Energy per Bit and Noise power of AWGN, which is used as SNR indicator.	
Subc. Occupation Index	0,5	Position of the used subcarriers within the OFDM symbols of FFT size.	FFT Size, Number of Subcarriers.
Subcarrier Spacing (Hz)	15k	Frequency spacing between subcarriers.	Sampling frequency.
Sampling Frequency (Hz)	7,68M	Frequency of the samples.	FFT size, Subcarrier spacing.

We defined default values which allowed us to run simulations without defining parameters every time. However, the main parameters related to a wireless communication system were selected with a friendly user interface, see Appendix C Figure C.1. Depending on the selection of such parameters, the other ones were automatically calculated and determined by the LabVIEW implementation scheme.

As explained in the previous section, our frame was made of data symbols and one pilot symbol. Depending on the modulation selected and the FFT size, the number of random bits necessary were generated. At the same time, the encoder was turned ON and OFF with a switch named *Encoder*. Depending on the code rate, the number of bits generated also varied. The following equation describes the mathematical relation between these variables. These calculations were implemented in LabVIEW as the Figure 3.4 shows.

Let us define the necessary bits to be generated N_{bits} as

$$N_{bits} = (\text{Number of Symbols} - \text{Number of Pilot Symbols}) \cdot \text{Number of Subcarriers} \cdot \text{bits per symbol} \cdot \text{coding rate}. \quad (3.1)$$

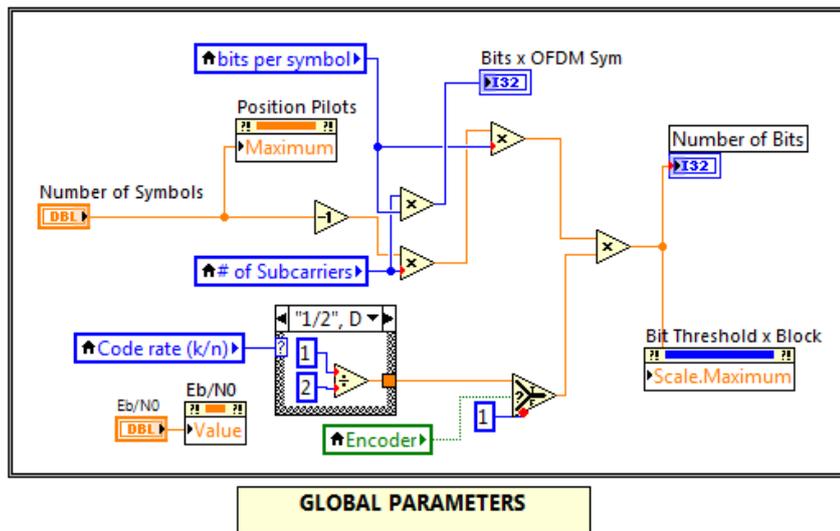


Figure 3.4: Design scheme for the calculation of the global parameters.

As the *number of pilots symbol* was always one per frame, we implemented a *decrement of 1* for the calculation. Moreover, by multiplying the *number of bits per symbol* and *number of subcarriers* we obtained the *number of bits per OFDM symbols*. The switch case element selected the rate that had to be applied, while turned off its value was 1. The E_b/N_0 variable was linked with the addition of the AWGN in the channel. In this LabVIEW design we obtained the values of the parameters using local variables, a exception of the *number of symbols*, *bits per OFDM symbol* and E_b/N_0 .

Transmitter

We describe in this section the design and functionality of each block independently, from the bit generation to the subcarrier spacing which fed directly the channel. We illustrate in the Figure 3.5 the block diagram of our transmitter design implemented on LabVIEW.

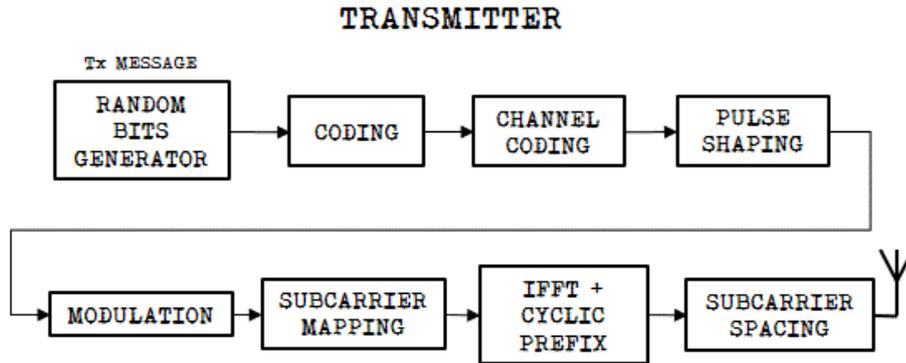


Figure 3.5: Transmitter model block diagram.

Random Bits Generator. From the global parameters, we obtained the *number of bits* necessary by the formula of the equation 3.1. Then, we created a uniform random number between 0 and 1, which was compared with 0. The output of the comparison was a boolean value (TRUE or FALSE), which was converted to a binary value (1 or 0), respectively. All together was implemented inside a FOR loop which iterated so many times as the *number of bits* calculated. The result of using a loop was that each of its iteration output was *indexed* in serial. In other words, every result of the calculation was added to an array. When all the iterations were done, the FOR loop outputted an array of '0s' and '1s', namely the *Random Bit Stream*. We generated our bit sequence using the design of the Figure C.2, in Appendix C.

Coding/Encoder. We chose a convolutional encoder for the coding of the entire data stream generated in the previous block. We employed for this module the *convolutional encoder Virtual Instrument* (VI) offered by LabVIEW in its communications library, described in Figure C.3, Appendix C. Since the block offered a wide configuration possibilities, we set some values when needed and the rest of them were internally defined by default (not wired).

We implemented this module as the design of the Figure 3.6 illustrates. First of all, in order to enable or disable the encoding, we utilized a boolean switch called *Encoder*. We created a case structure which allowed us to implement parallel design, one for a 'True' case and the other for 'False' case (ON /OFF). We see in the 'False' case (OFF) that the same blue wire that contained the *random bit stream* crossed the case structure without modifications, which means that no encoding was done. On the other hand, the 'True' case (ON) performed the encoding.

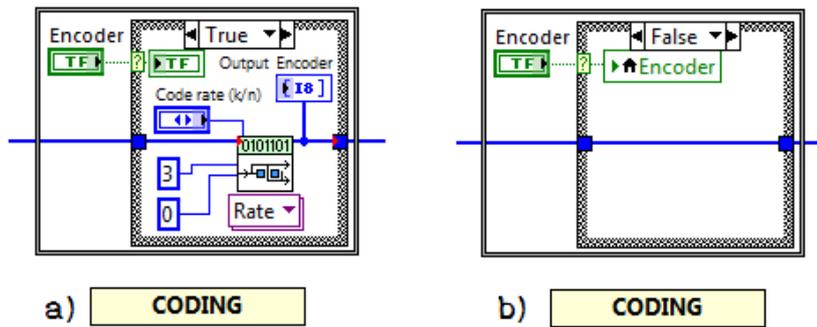


Figure 3.6: Encoder in LabVIEW. a) ON b) OFF.

DATA Split. We computed the splitting of the binary data stream following the design of the Figure 3.7. Because the encoder delivered a long stream of bits, we split it using this additional block. As illustrated in Figure 3.1, from this point we separated the bits into small packets, namely *bits per OFDM symbol*. Then each packet was processed independently and transformed into OFDM symbols. In order to do that, we employed a FOR loop which processed each symbol per iteration. At the end, all the symbols were concatenated forming the frame structure.

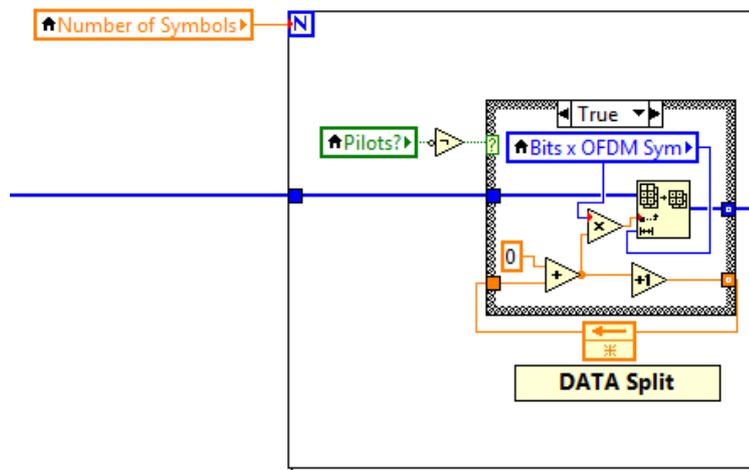


Figure 3.7: FOR loop and Bit stream splitting into symbols in LabVIEW.

The local variable *Number of Symbols* connected to the node **N** determined the number of loop iterations. The data stream was separated into small packets. However, an addition packet was inserted in the packet sequence, the pilot symbol. In order to do that, we designed a *Pilots?* boolean variable, which was TRUE when the current loop iteration was inserting the pilots. Since the FOR loop needed an extra iteration for the pilots, we designed this block not only to split the data stream, but also to coordinate the insertion of the pilot packet into the frame. The description of the FOR loop of LabVIEW is shown in Figure C.4, Appendix C.

The procedure was the following; First, depending on the position of the pilot symbol, which was defined in the global parameters (by default 1), the loop counter enabled the variable *Pilots?*. In the case of no pilots, the 'True' box case was enabled, which is shown in Figure 3.7. The data stream was divided using the VI called *Array Subset* into packets of *bits per OFDM symbol* size, see Figure C.5 of Appendix C. However, the selection of the pilot position had to be coordinated in order to avoid the loss of any data packet while computing.

Hence, we created a counter wired to a feedback operator (orange backward arrow) that obtained the value of the previous iteration. In this way, every time a data symbol was processed the counter increased, but when pilots were inserted the counter kept the same value. This could seem trivial but we have to mention that in every iteration all the elements in LabVIEW were reseted to zero. Using buffers, registers and feedback operators we shared values between iterations. Thus, each time the data was packeted the counter increased one, keeping the value in the feedback operator. While inserting a pilot symbol, the counter remained constant and the box case was set to 'False', giving no output packet as the design of the Figure 3.8 illustrates. We note that the pilots insertion was processed in the *modulation* block.

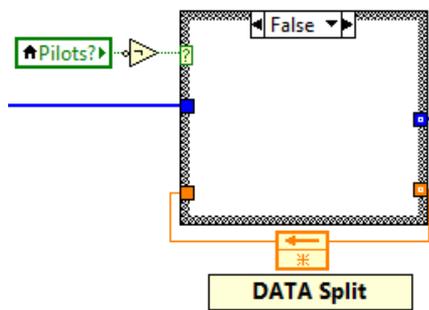


Figure 3.8: DATA split 'False' case in LabVIEW.

Channel Coding/Encoder. The functionality of the channel encoder performed an interleaving of the data input according to a permutation array. This operation was computed using the LabVIEW VI shown in Appendix C, Figure C.6. On the other hand, Figure 3.9 describes in detail the permutation process done inside this VI, which for our implementation we used the permutation array [1,3,5,2,4,6,0,7].

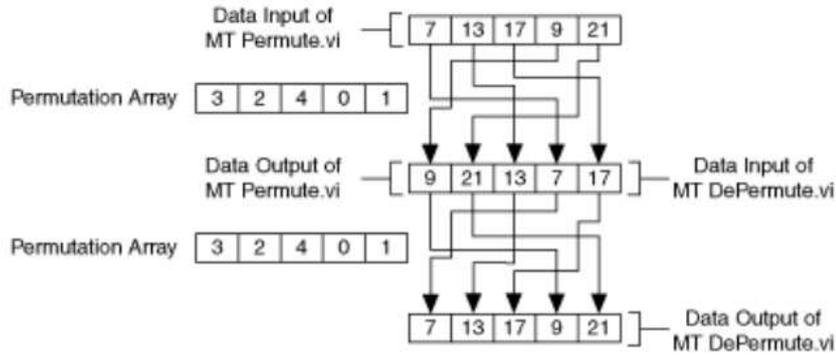


Figure 3.9: Channel Coding Interleaving and Deinterleaving. If the data in array is [7,13,17,9,2], and the permutation array is [3,2,4,0,1], the data out array would be [9,21,13,7,17] [20].

Since the pilots symbol were inserted in the modulation block, we designed a scheme that prevented the system to stop when no data packet was being processed. That happened if one of the VIs tried to compute input data but there was none, which was the case of the pilots insertion loop iteration. That also resulted in no need of interleaving or channel coding, which outputted an empty array. If the modulation block received an empty array (nothing to modulate), then the simulation stopped due to an error. As a solution, if an empty array was given, the 'True' case of the operational switch delivered an array of size 2000 with zeros. Its values and size were trivial, since it only fed the modulation block with random data which was substituted with the pilots symbol. If the interleaver VI delivered data (not an empty array), then the switch case outputted this data, which was actually the data packet going to be modulated. We present the implementation of the channel encoder module in the Figure 3.10.

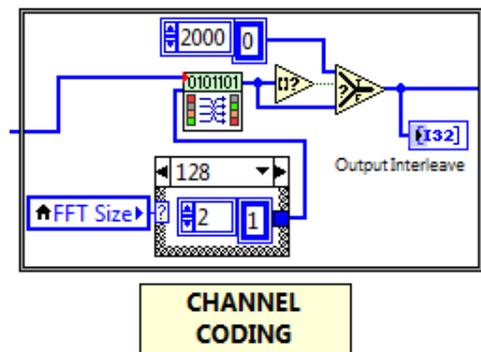


Figure 3.10: Channel Coding in LabVIEW.

Pulse Shaping. For the digital modulation in LabVIEW it was necessary to define the filter coefficients for pulse-shaping. We defined them as shown in the design of the Figure C.7 of the Appendix C. We implemented this block outside the FOR loop because these parameters were constant for all the symbols. Figure C.8 presents the VI responsible for the generation of the filter coefficients. Let us mention that for our communication system testbed we set always OFF the pulse shaping filter.

QAM System Parameters. Apart from the filter coefficients, the modulator VI was set up with an additional VI, which grouped parameters such as modulation type and samples per symbol into a cluster, which was a configuration input for the modulator block. This was realized by the LabVIEW VI depicted in Figure C.9 of the Appendix C. We present the QAM parameters module in the Figure 3.11.

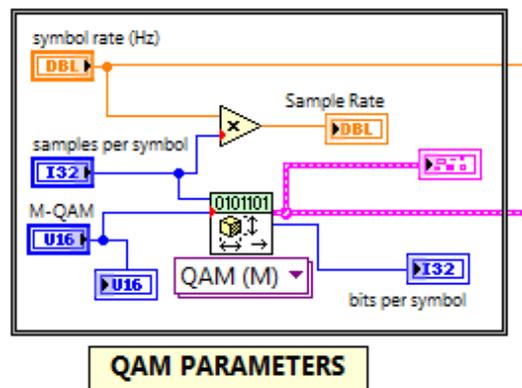


Figure 3.11: QAM parameters in LabVIEW.

QAM Modulator and Pilots Insertion. In this module we generated our QAM symbols and inserted the pilots symbol. The module design is illustrated in Figure 3.12. As we explained before, the QAM modulator VI from LabVIEW (see Figure C.10 Appendix C), modulated the input bit stream and delivered it to the *insert pilots* block. This block while enabled inserted the pilots. While disabled, let the modulated data symbols pass through it without alterations. The FOR loop counter [i] was compared with the position of the pilots and if they matched the pilots had to be inserted, namely the insert pilots module was enabled. The number of subcarriers were required in order to insert the pilots only in the occupied subcarriers, not in the guard band. Furthermore, in order to monitor the inserted pilots we added an output variable named *Pilot Array*.

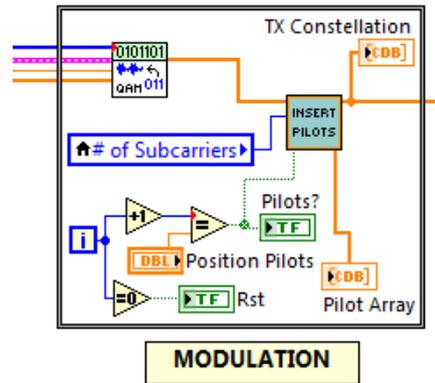


Figure 3.12: QAM modulator in LabVIEW.

When the module was enabled ('True' case), the input symbols were discarded (no wired). Instead, a pilot array was generated using a pattern. For testing purposes we transmitted a QPSK constellation as pattern, presented in Figure C.11 Appendix C. Then, the number of subcarriers were considered in order to define the pattern size. The pilot array had a feedback which in case of more than one pilot symbol per frame, allowed to use the same pattern in more loops iterations.

For our simulations we generated the pilots incorporating a *Zadoff-Chu* or *Cazac* sequence, frequently applied for wireless communications, see Chapter 2 Section 2.10 equation 2.36. One of the relevant properties of such sequences were the zero autocorrelation of its complex-signal.

We produced it following the next mathematical equation:

$$x_u(n) = e^{-j \frac{\pi u n (n + 1 + 2q)}{N_{ZC}}}, \quad (3.2)$$

where

$$\begin{aligned} 0 &\leq n < N_{ZC} - 1, \\ 0 &\leq n < N_{ZC} \wedge \gcd(N_{ZC}, u) = 1, \\ q &\in \mathbb{Z}, \\ N_{ZC} &= \text{length of sequence.} \end{aligned}$$

The calculation of the sequence array was implemented with a LabVIEW script-box as Figure C.12 illustrates. We set as parameters a length of 29 and a 25th root. Then, the CAZAC sequence was repeated until all the occupied subcarriers were filled. The selection of the pilot pattern was defined as default for the Zadoff-Chu sequences. The implementation of the insert pilot module is shown in Figure 3.13.

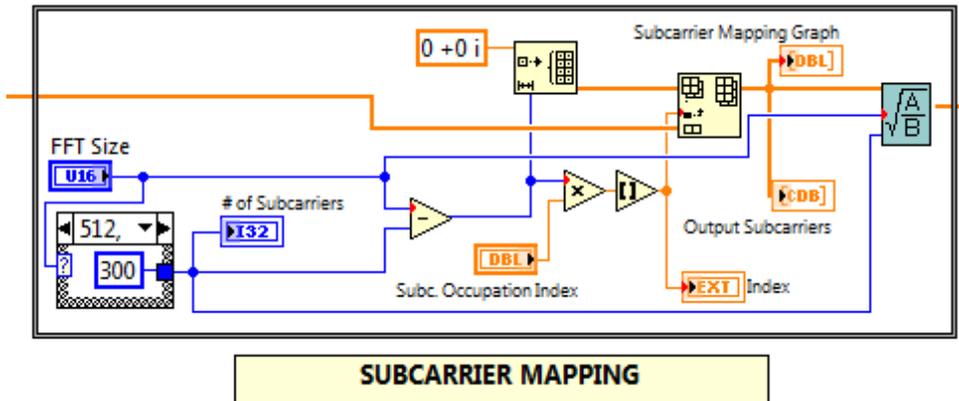


Figure 3.14: Subcarrier Mapping in LabVIEW.

The FFT size determined the occupied subcarriers. For instance, for a FFT size of 512 the box case outputs a value of 300 subcarriers, following the Table 3.2. Then, we created an array of zero complex values ($0 + 0i$) which formed the guard band. The size of this array was the result of the subtraction ($FFT\ size - Number\ of\ Subcarriers$).

Having the guard band array generated, we inserted the data (or pilots) with an *Insert into Array* LabVIEW block. With the *Occupation index* we selected the position (*index*) of data/pilots within the symbol and as a result we obtained the subcarriers mapped. The modulated symbols had an average power of 1. After the mapping it was necessary to normalize using this relationship $\sqrt{\frac{FFT\ size}{No.\ of\ Subcarriers}}$, which was implemented in the last module.

IFFT and Cyclic Prefix. We concluded the last block inside the FOR loop of our transmitter with the inverse Fourier transform (IFFT) and the insertion of the cyclic prefix (CP). The first operation that was realized in this block was the IFFT, which generated the Orthogonal Frequency Division Multiplexing (OFDM) symbols. Here, the subcarriers of the previous block were orthogonal spaced into multiple carrier frequencies. The module that performed such operation was the *Inverse FFT* LabVIEW VI, shown in Figure C.14 Appendix C. As the input data was complex valued, the setup of the module was defined for '1D Complex' data input. Afterwards, we normalized the signal using the relation $\sqrt{FFT\ size}$. Figure 3.15 depicts the block design.

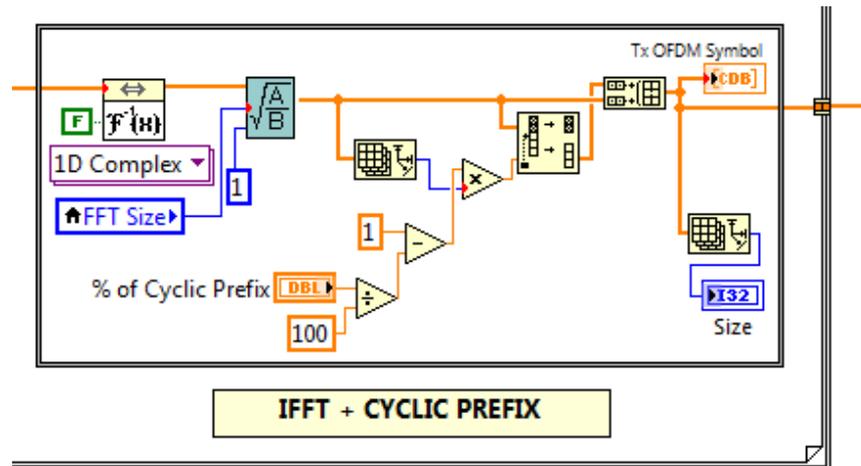


Figure 3.15: IFFT and Cyclic prefix block in LabVIEW.

After generating the OFDM symbols, we added the cyclic prefix. We computed this operation by using the LabVIEW basic VIs shown in Figure C.15 Appendix C. With the size of the OFDM symbols, we calculated its percentage of cyclic prefix and separated the OFDM symbol part which was going to be used as CP. As a result, the last part of the OFDM symbol was extracted and appended at the beginning of the OFDM symbols, obtaining *Tx OFDM symbol* with CP.

One OFDM symbol with CP was generated in each FOR loop iteration. Therefore, the complete frame structure was created concatenating the symbols generated in each FOR loop iteration, forming a data stream. Let us mention that without this concatenating, each OFDM symbols would have been given as an array, in parallel.

Subcarrier Spacing. From the previous block we received a frame made of the pilots symbol and the data symbols. These were OFDM modulated and in this block named *Subcarrier Spacing* we spaced each orthogonal subcarrier with a specific sampling period. The sampling period was calculated as inverse of the sampling frequency, which was computed as $S_f(\text{Hz}) = \text{Subcarrier spacing} \cdot \text{FFT size}$. Then, we built the waveform which was sent directly to the channel. We present the design scheme of this block in the Figure C.16, Appendix C.

Channel

The OFDM frame was fed into the channel, where first a fading profile was applied and afterwards the AWGN added. Then, the channel output was delivered directly to the receiver. We designed both blocks independently from each other, which means that fading and AWGN were enabled/disabled when desired. The channel model for our digital communication system is represented in Figure 3.16.

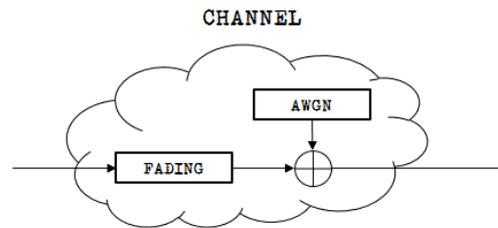


Figure 3.16: Channel model block diagram.

Rayleigh Fading Taps. The fading applied to the signal followed a Rayleigh distribution, which was generated using taps. First, the fading taps module, illustrated in Figure C.17, generated the profile using a typical urban or indoor power delay profile (PDP), which is shown in the Table C.1 at the Appendix C. In order to match our PDP taps to our sampling frequency, we executed a script which generated the time realization of the selected PDP and added a random value to each tap. In this way, we simulated different realizations of a fading channel. The script responsible for these operations is presented in Figure C.18.

Having the time realization of the fading profile, we convoluted it with our frame. As a result, we applied the fading profile to our signal, simulating the fading channel of a wireless communication. We depict the complete block in Figure 3.17.

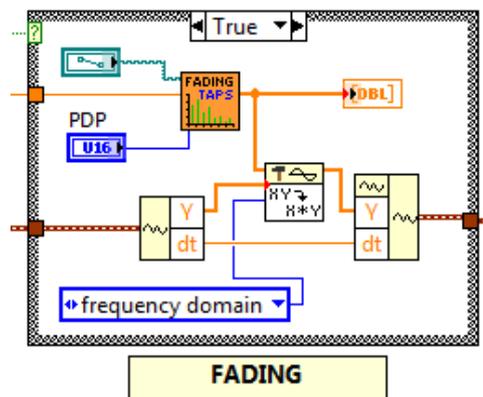


Figure 3.17: Fading block (enabled) in LabVIEW.

Receiver

The blocks of the receiver basically reversed the processes done in the transmitter, such as cyclic prefix removal, the FFT, demapping, demodulation and channel decoding and convolutional decoder. However, the receiver was responsible also to correct errors when possible. Moreover, advanced receivers were able to mitigate interferences and channel distortions. Thus, additional blocks such as channel estimator and equalizer were implemented. The complete block diagram for our receiver prototype is shown in Figure 3.19.

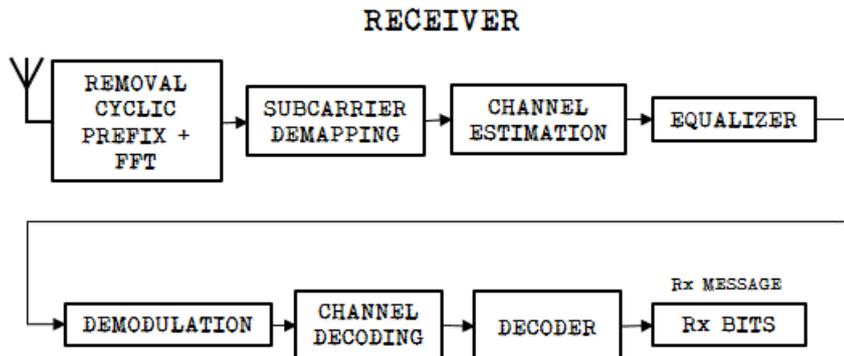


Figure 3.19: Receiver model block diagram.

Apart from the design and implementation of the receiver communication blocks, as we described in the flowchart of the Figure 3.2, we created additional modules for the OFDM symbols split and pilot symbol deletion. We describe now each LabVIEW receiver blocks in order as described in the related diagrams and flowcharts.

OFDM Symbol Split. The frame transmitted over the channel was composed of OFDM symbols which were affected by fading and AWGN. The receiver had to obtain the frame, extract each OFDM symbols and process them. In order to design this scheme we employed a FOR loop at the receiver, and the *OFDM Symbol split* block. Altogether, separated the frame into OFDM symbols, which were processed independently in each FOR loop iteration through the receiver communication modules.

The input variables of the FOR loop were the frame and the dimension of the OFDM symbols with CP. The loop iteration counter [i] was correlated to the symbol position of the frame. Thus, it was utilized for the pilot symbol localization and for the data splitting.

As outputs of this block, we obtained the data/pilot array that corresponded to an OFDM symbol. On the other hand, the boolean variable *channel estimation [Ch.Est.]* compared the defined pilot position with the current iteration. When it matched, the pilots from the frame structure were being processed and consequently the channel was estimated. In other words, this comparison activated the *channel estimation* block. We illustrate the LabVIEW implementation of this splitting in Figure C.20 Appendix C.

Removal Cyclic Prefix and FFT. We designed at the receiver the removal of the cyclic prefix before performing the fast Fourier transform (FFT), reversely to the block in the transmitter side. As required inputs we had one OFDM symbol and its size, which was measured to calculate the part that was cyclic prefix and had to be removed. Figure 3.20 shows the complete block design.

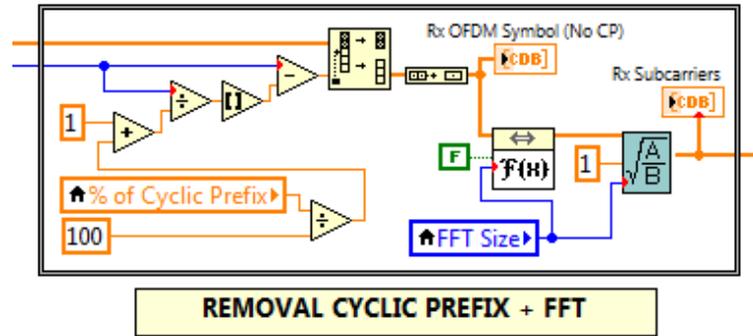


Figure 3.20: Cyclic prefix removal and FFT block in LabVIEW.

Then, the OFDM symbol without CP was fed to the FFT module, see Figure C.21 Appendix C. After the FFT we normalized the signal using the relationship $\sqrt{\frac{1}{FFT\ size}}$.

Subcarrier Demapping. The subcarrier demapping process was implemented basically extracting the portion of the array where the occupied subcarriers were placed. Then, a normalization factor was applied following the relationship $\sqrt{\frac{No.\ of\ Subcarriers}{FFT\ size}}$. The design scheme is presented in Figure C.22 Appendix C.

Channel Estimation. The channel estimation block consisted on a switch box case controlled by the boolean variable *Channel Estimation* [Ch. Est.]. While this variable was true ('True' case), the received pilots were currently processed. Then, as explained before the received pilots were compared with the known transmitted pilots sequence, giving as a result the channel response.

On the other hand, if no pilots were being processed in the current loop iteration, then the box case was set to 'False' and it delivered as output the last value calculated of the channel response. In this way, we simulated an environment where between pilots symbol the channel response was the same for each data symbols. On the other hand, at every pilots symbol received the channel response was newly calculated.

Figure 3.21 shows the designed block for channel estimation. The channel response was obtained by the relationship between the received pilots symbol and the reference pilots symbol (CAZAC sequence), see flowchart Figure 3.2.

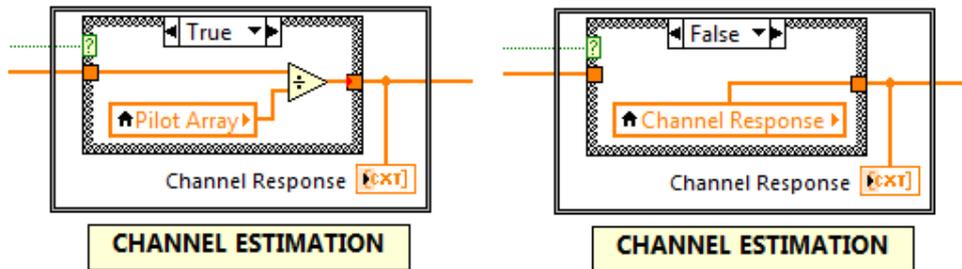


Figure 3.21: Channel estimation block. (Left) True case for ch. estimation and (Right) False case for ch. estimation.

Equalizer. In this module we employed the channel response H previously calculated to equalize the data symbols. S . The symbols were divided by the channel response, namely multiplied by the inverse of the channel response (Zero Forcing). As a result, we observed the constellation of equalized data symbols. Figure 3.22 illustrates the design of the equalizer module.

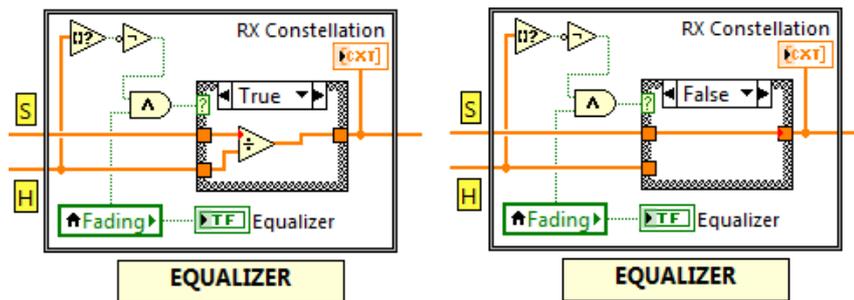


Figure 3.22: Equalizer block. [S] Symbol [H] Channel Response. (Left) True case for equalization and (Right) False case for no-equalization.

Demodulation. For the demodulation, LabVIEW offered a module which mapped the symbols into a bit stream according to a defined modulation scheme. We employed this module with some modifications, as shown in Figure C.23, Appendix C. This module in presence of an input empty array stopped the LabVIEW simulation producing errors non related to our system. Thus, we solved that issue implementing a module named *Empty?* which detected if the array was empty, and if it was substituted it with a zero array, see Figure C.24. Afterwards, the demodulation module of LabVIEW called *MT Map Symbols to Bits* generated the estimated bit stream, see also Figure C.24.

Channel Decoding. The last block inside the receiver FOR loop was the channel decoder. In the same way that the encoding was done in the transmitter, at the receiver a module named de-interleaver decoded the bits using the same permutation array. Afterwards, as each OFDM symbol was processed independently, we obtained the complete bit streams by concatenating all the bit generated in each FOR loop iteration. We depict the implementation of this block in Figure C.25, Appendix C. The de-interleaver module was available in LabVIEW (see Figure C.26), and its permutation procedure was explained in the Figure 3.9 at the transmitter section.

Delete Pilots. The digital communication system simulation ran linearly from left to right. As we inserted pilots in a new loop, it was not possible to delete them inside other loop. The reason was that for each loop some data was outputted. In the receiver part, while we processed the pilots the other blocks ran as usual and delivered not useful data. This data went through all the receiver blocks and it ended concatenated as a not desirable bit stream. These bits were related to the pilots, which did not contain data information and therefore had to be deleted. We implemented the block of Figure C.27 Appendix C in order to delete these non-desired residual bits.

Decoder. The same parameters were used for encoding and decoding, using the convolutional decoder VI based on Viterbi soft decision, see Figure C.29 at Appendix C. The decoder block was only enabled if at the transmitter the encoder was activated. Therefore, the same activation variable was used in both blocks. We designed this block as represented in Figure C.28.

Apart from the decoding module, we designed a patch module to correct some functionality issues of the convolutional decoder. When decoding, the LabVIEW module outputted one less bits. That was because of the decoder itself, which needed these bits for the decoding process, named *traceback depth* (D).

Received Bits. After all the receiver modules, we obtained the estimated bit stream. There was not much to implement here apart from its definition. However, the received bits were employed to performance simulation analysis such as BER and BLER over different channel conditions. In an ideal digital transmission system with no noise and fading, the received bit stream was equal to the bits generated in the transmitter.

Having designed the whole communication system, we continued with the implementation of the modules to calculate and plot the system performance parameters such as BER and BLER for an AWGN and Rayleigh fading channel. We present in the next section the design and implementation in LabVIEW of such performance analyzers.

3.1.2 Performance Analyzers Design

Since the LabVIEW work-flow was linear, some variables were shared and stored between iterations using registers or feedback operators. In order to calculate system performance indicators such as the bit error rate (BER) we needed these variables and its accumulated values among all the iterations, particularly the transmitted and received frames. We designed a communication system structure which was based mainly on hierarchic FOR loops. Within these loops, apart from the digital transmission we developed the blocks required for BER/BLER calculation and graphical representation of the BER/BLER versus the signal-to-noise ratio (SNR). The FOR structure of our complete digital communication system with its performance analyzers blocks is shown in Figure 3.23.

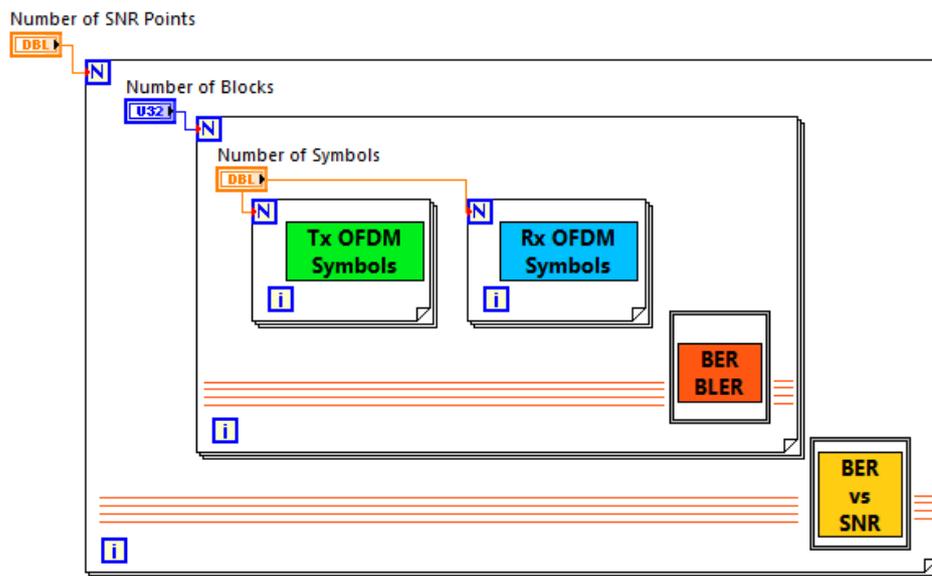


Figure 3.23: FOR loop structure of the communication system in LabVIEW.

As we described before, we created each of the OFDM symbols within a frame with a FOR loop at the transmitter, and we inverted the process of symbol extraction at the receiver with another FOR loop. Both FORs were iterated so many times as the number of symbols per frame desired, as we can see from Figure 3.23. Then, we created the operations for BER calculation which mainly took into account the number of transmitted bits and the number of received wrong bits. That was compared for the instantaneous BERs, which evaluated only one frame. However, for more frames we needed a procedure to keep in memory the values of each frame simulation, because as mentioned at every new iteration/simulation most of the variables values returned to its default value, zero. Hence, we implemented the structure illustrated above.

When the instantaneous BER/BLER block was calculated, we added a new FOR which contained all the LabVIEW design. Then, it was possible to specify how many iterations the complete system will perform. In each iteration one frame was transmitted, therefore we named *Number of Blocks* to the number of frames that were transmitted and received. On the other hand, the FOR loops offered the possibility to create *registers*, which stored variables of one iteration and shared them for the next one. Using registers we noted that one iteration delay was introduced at the simulations.

Back to Figure 3.23, we observe that parallel orange lines cross the [Number of Blocks] FOR loop and the BER/BLER calculation block. These lines connected the registers defined in both sides of the loops and allowed us to use the BER values computed for previous frames. As a result, we calculated the *accumulated BER/BLER* from the total transmitted frames.

On the other hand, in order to plot the BER versus the SNR, it was needed to transmit frames over a AWGN channel with different signal-to-noise values. In our channel implementation, we were able to set the value with the E_b/N_0 in dB. Then, we created the second FOR loop which encapsulated again the complete LabVIEW design. Doing that we designed the BER vs. SNR block which set different values of E_b/N_0 per loop iteration.

Concluding, for each SNR point we calculated the accumulated BER. Then, we plotted the SNR points versus its related BER. As we did for the accumulated BER, we needed to store the SNR vs. BER of all the defined SNR points. That was implemented with additional registers, which are represented in the previous figure as the parallel orange lines crossing the block BER vs. SNR. The outer FOR loop iteration was defined as shown in Figure 3.24, where its parameters were described in the last section of BER vs. SNR plot.

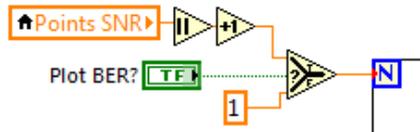


Figure 3.24: Outer FOR loop for Plotting in LabVIEW.

To summarize, the designed FOR structure allowed us to send several blocks/frames, calculate its instantaneous and accumulated BER/BLER and finally represent them versus different values of SNRs.

BER/BLER Calculation

In this section we describe the block that calculated the instantaneous and accumulated bit error rate (BER). Furthermore, it set the value of E_b/N_0 and computed the block error rate (BLER). Apart from that, we also present the design of the registers of the FOR loop and its essential functionality within the loop iterations. We illustrate this LV module in Figure 3.25.

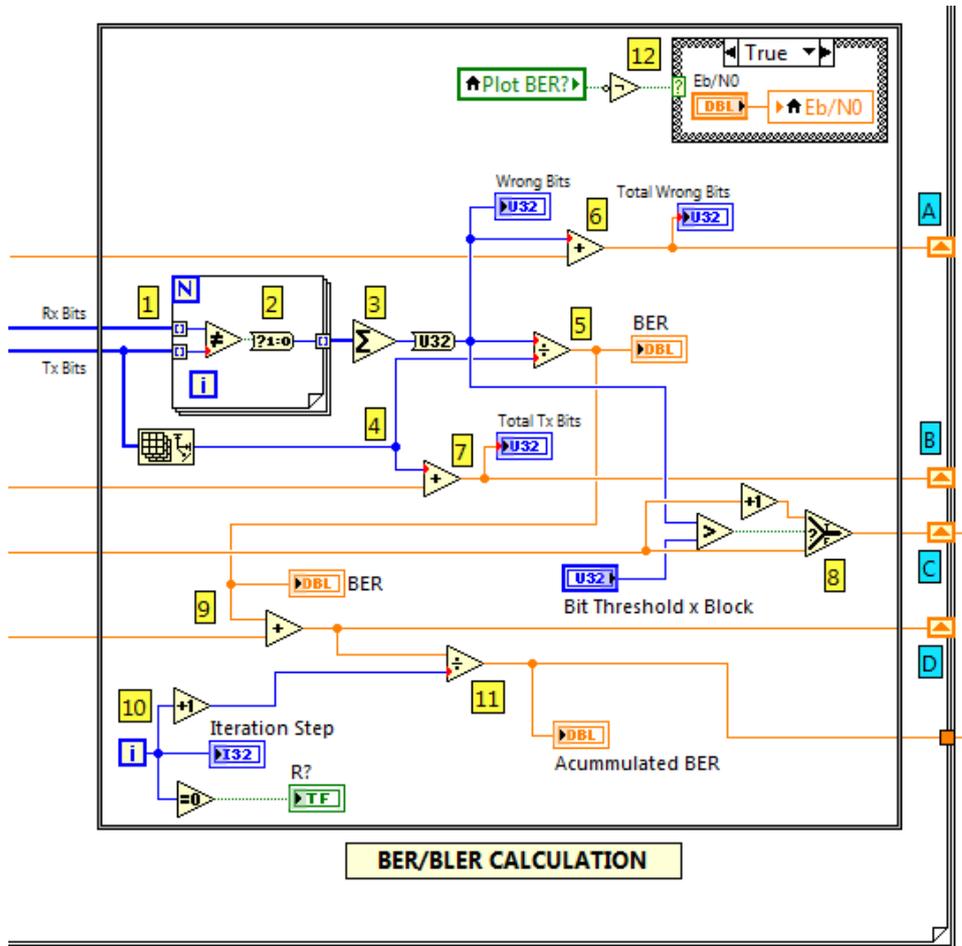


Figure 3.25: BER and BLER calculation Block in LabVIEW.

We observe several modules within the square, while we also distinguish the border of the FOR loop and how some of the variables were stored as registers, which are depicted as upper arrows within orange boxes.

For a better understanding of its computing functionality we added some numbered labels to describe sequentially the BER/BLER calculation block as follows:

1. From the transmitter and receiver part we obtained the transmitted bit stream (Tx Bits) and received bit stream (Rx Bits), respectively. We fed them in a FOR loop, indexing the streams. That means that every value was stored in a vector array.
2. As the bits were indexed, the FOR loop iterated so many times as the size of the vector array. We used this procedure to compare each bit independently between transmitted and received bit stream. The indexing allowed to compare Tx bit in position (i) with Rx bit in position (i). If they were different, a comparator outputted a 'TRUE' value, which is transformed into a digital '1'. In case of equal bits, the comparator outputted a 'FALSE' value, transformed into a digital '0'. The FOR loop stored the array of '0s' and '1s', which represented *no bit error* and *bit error*, respectively.
3. The sum module received the binary digital array and summed all its values. As a result, we obtained the total number of error bits, named as *Wrong Bits*.
4. From the transmitted bits, the *size array* module delivered the size of the array, which was the number of transmitted bits. As defined in the global parameters, the number of bits per frame were set at the beginning.
5. With the relation between number of wrong bits and number of transmitted bits, we calculated the instantaneous BER per frame.
6. The number of computed wrong bits was added to a register, labeled **A**. That worked by adding the current wrong bits value to the calculated in the previous iteration. As a result, we obtained the *Total Wrong Bits* of the frames.
7. The number of bits per frame sent was added to a register, labeled **B**. Adding the current number of transmitted bits to the bits sent previously, we calculated the *Total Tx Bits*.
8. This scheme calculated the BLER. It consisted in a counter and a comparator. The comparator was set with a number of bit error threshold. When it was reached the comparator delivered a 'TRUE' boolean value, but a 'FALSE' when not. Then the counter was connected to a register, labeled **C**. Let us define a threshold of zero error bits. Then, when there was no wrong bits, the comparator output was a 'FALSE' and the register stored the previous value of the counter in the current iteration. If there was at least one wrong bit, then the comparator output was a 'TRUE' value and the previous counter value was incremented one. As a result, the current iteration, which was related to a transmitted block, had an increment of one in that counter. That was how the block error rate was calculated, for wrong bits on the frame/block, the complete block was considered error and blocks errors were also accumulated by the registers.
9. The instantaneous BER was here connected to the register **D**, which stored in the current block iteration loop the current BER summed to the previous one calculated.
10. This FOR loop counter [**i**] was related to the number of blocks/frame sent. We monitored the iterations with the variable *Iteration Step* and we also created a reset switch *R?*, which was activated every time the counter was initialized. That was important to reset our registers while plotting BER versus SNR, because a new transmission with a different SNR was set.

11. With the number of transmitted blocks (iteration step) and the summed BER of all the blocks, we computed the accumulated BER.
12. This module was responsible of setting the E_b/N_0 value while plotting the BER. Therefore, a boolean variable *Plot BER?* activated and deactivated the parameter control within the case structure. The 'False' case was empty.

In addition to the internal behavior, in order to plot the BER vs. SNR, we needed the values of the accumulated BER and BLER. These were outputted with wires from the FOR loop, as shown in the previous Figure 3.25.

We illustrate in the Figure 3.26 the left side registers of the FOR loop. They were designed in such a way that at every reset (see label 10 Figure 3.25) its value was set to zero. Afterwards, they delivered the previous variables values **A'**, **B'**, **C'** and **D'**, related to the labels described above.

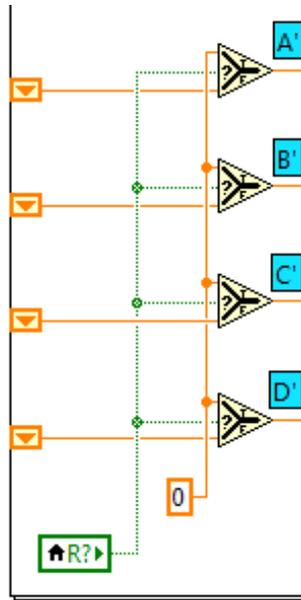


Figure 3.26: Registers of the left side of the frame FOR loop in LabVIEW.

BER/BLER versus Eb/N0

With the values of the accumulated BER and BLER of a defined Eb/N0 value, we implemented the second FOR loop to obtain the values of these parameters for several Eb/N0 values. As a result, it was possible to represent the relationship between the BER/BLER and different Eb/N0. First, we set the range of Eb/N0 values, *start* and *end points*. Then, we defined *step size* or distance between Eb/N0 points. With these parameters defined in the GUI, we calculated the *Number of SNR points* to be generated. That defined the number of iterations of this FOR loop. Finally, we created a 2D array (X-Y), which contained the SNR points and its respective BER/BLER values and were stored in new registers.

We show the block in Figure 3.27 and we describe in detail its work-flow as follows:

1. The boolean variable named *Plot BER?* was used here to enable the plot generation. It became handy for speeding simulation tests because the plotting took time while generating the BER and BLER arrays for graphical representations. We present the 'True' case which enabled the box case. The 'False' case was designed empty.
2. In the GUI it was defined the *Start* and *End* SNR value to plot, and also the steps size between points.
3. To the number of SNR iteration scaled by the step size, we added the start Eb/N0 value and set the initial Eb/N0. Then, as the iteration SNR increased per frame sent, the Eb/N0 was also increased.
4. With the range of SNR points, we calculated the number of SNR points required for representation. Then, we connected it to the small FOR loop that defined the size of the 2D arrays containing the BER vs. SNR values of the Y and X axis, respectively.
5. The FOR loop generates an array with the SNR values used for plotting with the same dimension as the number of SNR points to plot.
6. The SNR array (X-axis) was merged with the BER array (Y-axis). The output is a 2D array.
7. The SNR array (X-axis) was merged with the BLER array (Y-axis). The output is a 2D array.
8. Both 2D arrays were plotted generating the desired BER vs SNR and BLER vs SNR graphs.
9. Since the previous FOR was related to single blocks/frames, we calculated the accumulated BLER by dividing the number of total wrong blocks by the total of blocks sent. The obtained BLER was connected to the register **E**, which was required to store all the values in the BLER array. The same was implemented for the accumulated BER outputted from the frame FOR loop, which was stored in the register **F**.
10. Apart from the BER and BLER values, in order to add sequentially these variables into arrays, we had to share these arrays between SNR loop iterations, which as explained before was done by registers. The BER array was stored in register **G**, while the BLER array was stored in register **H**.

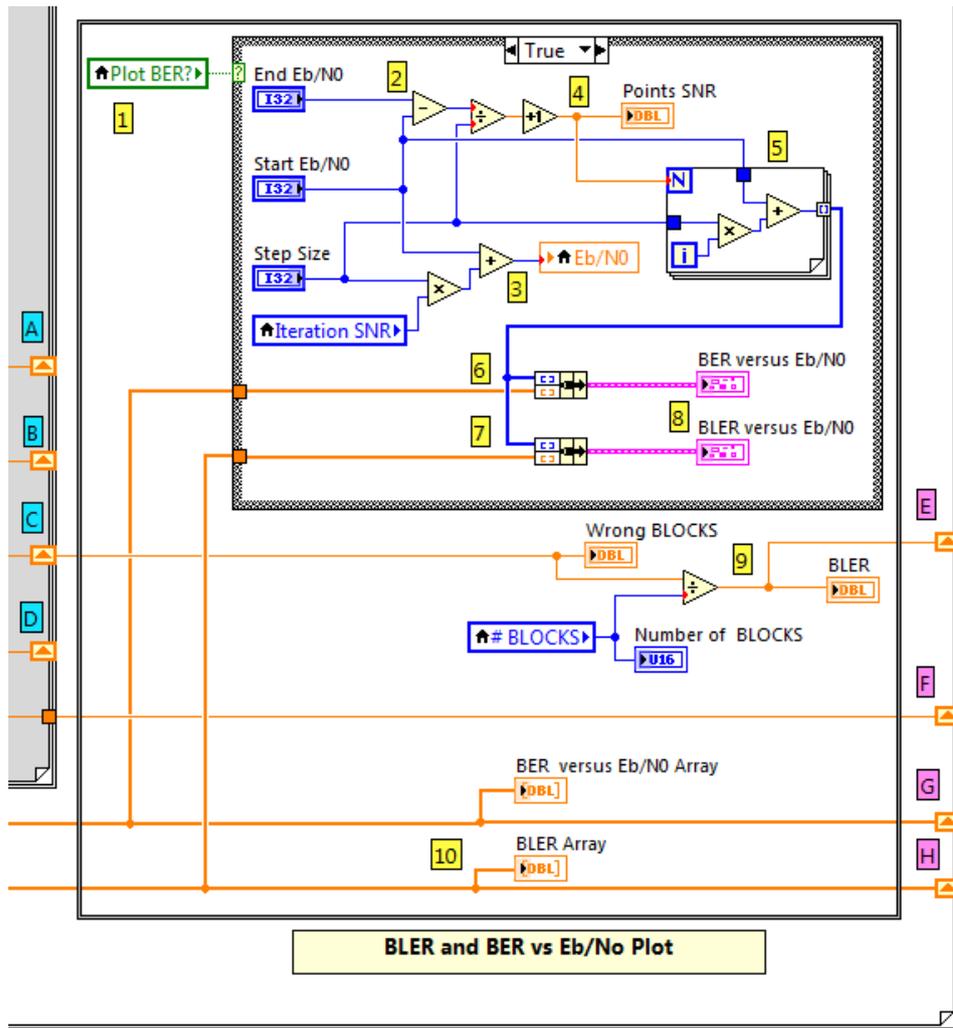


Figure 3.27: BER and BLER vs Eb/No Plot Block in LabVIEW.

The registers and array definitions for the storage of the BER and BLER values is described in three phases:

- Initialization
- Storage in arrays
- Finalization

The next Figure 3.28 shows the design of the registers at the left side of the FOR loop. The BER, BLER, array BER and array BLER were stored in their respective registers **E'**, **F'**, **G'** and **H'**.

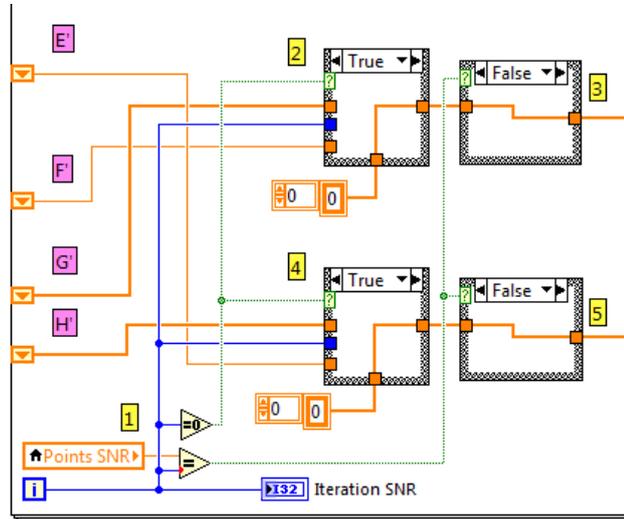


Figure 3.28: Initialization of the arrays for BER and BLER vs Eb/N0 Plot Block in LabVIEW.

We describe the array *initialization* as follows:

1. The FOR iteration counter [**i**] and the *Points SNR* were crucial for the box case structures. First, the initialization was realized in the first iteration $i=0$. The comparator to '0' outputted a 'TRUE' boolean value which activated the 'True' case of the first column of box cases (label 2 and 4). The comparator to the SNR point was designed particularly for the finalization phase, which enables the second column of box case (label 3 and 5). That happened when the iteration counter reached the last point to represent in the graph.
2. The 'True' case created and initialized an array with a zero value for the BER plot.
3. This zero value array was sent to the respective register **G'** on the right side of the FOR loop.
4. The 'True' case created and initialized an array with a zero value for the BLER plot.
5. This zero value array was sent to the respective register **H'** on the right side of the FOR loop.

Let us mention that because the registers delayed the use of the parameters one iteration, it was necessary to create a one zero value array which contained the BER and BLER values, independently.

When the first iteration was done and the first BER/BLER values were generated and stored, the next phase was the *storage in arrays* as shown in Figure 3.29 and described as follows:

6. The comparators delivered a 'False' boolean value in this phase, because the iteration loop was in the range from the second iteration until one iteration before the last SNR point.
7. The box case was set to 'False' executing the storage process. That consisted in an *Insert into Array* VI which was fed with the previous initialized zero value array, the position where to insert a new value, namely iteration $[i]$ and the last input the value to insert, the previous calculated BER. Then the array of one zero and one BER value was passed to the register. In the next iterations, the array size increased while new BER values were inserted until obtaining the last value, which changed the phase to Finalization.
8. The same description and procedure from the previous label is applied for this block where the BLER was stored in each iteration in the BLER array.

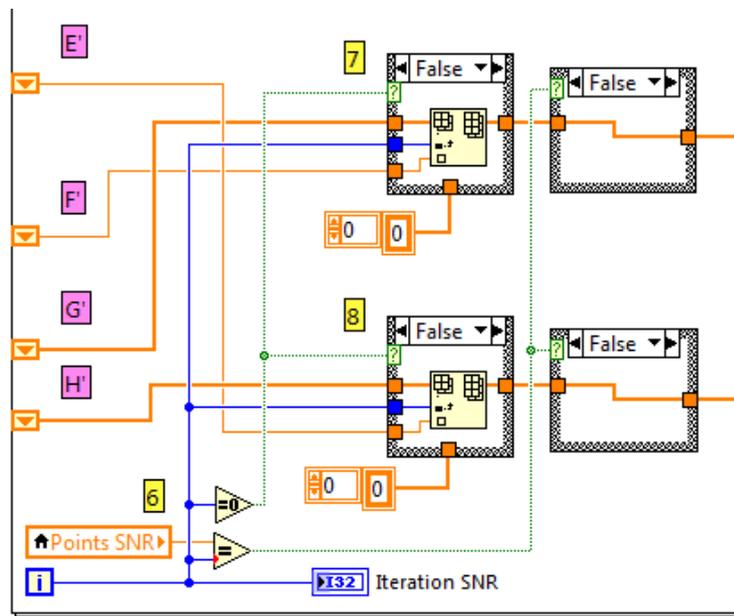


Figure 3.29: Storage in the arrays for BER and BLER vs E_b/N_0 Plot Block in LabVIEW.

The *finalization phase* was activated when the last SNR value with its respective BER and BLER was calculated. Figure 3.30 illustrates this phase and we explain its behavior as follows:

9. The last iteration of the FOR loop switched the comparator with the variable *Points SNR* to an output 'TRUE' boolean, which enabled the second column of box case (labels 10 and 11).

10. The enabled box case executed the *Array Subset* VI, which outputted a portion of an array from a starting position until the last value. We designed it selection the position 1 until the last value of the array. In other words, this module discarded the first value at the position 0 of the array, which was the zero value. As a result, we obtained our complete BER array with no redundant value.
11. The same description and procedure from the previous label is applied for this block where the zero value is discarded from the BLER array.

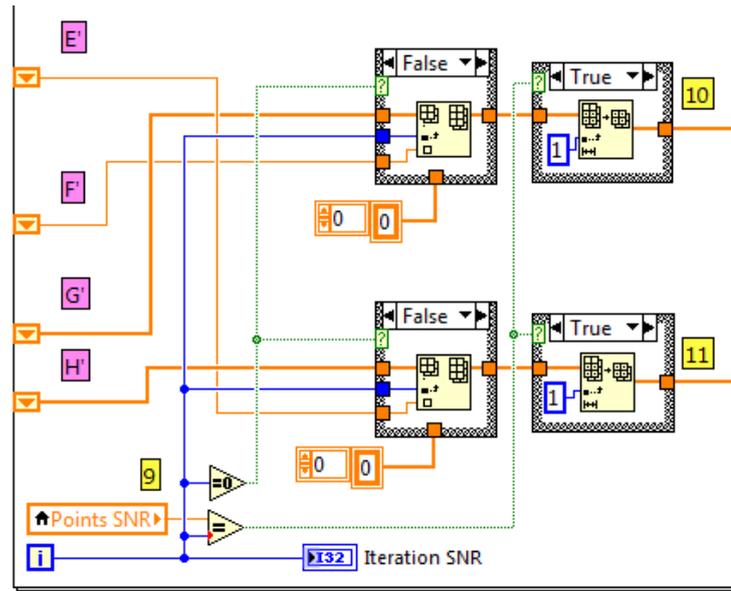


Figure 3.30: Finalization of the arrays for BER and BLER vs E_b/N_0 Plot Block in LabVIEW.

To summarize, we built with our design an array of calculated BERs and BLERs over different SNR values. The FOR loop iteration was defined as shown in Figure 3.24. Afterwards, we plotted the results in a BER vs. SNR graph fulfilling our block purpose. We illustrate the complete LV communication system design in Figure C.30 and Figure C.31 of the Appendix C.

3.1.3 Graphical User Interface (GUI)

The implementation of a communication system on a software defined radio such as LabVIEW offered us the advantage to design and utilize a friendly graphical user interface (GUI). The GUI allowed us to track and monitor all the data flow, to represent variables, to configure parameters while simulating and to generate the desired plots for performance analysis. Therefore, apart from the schematic design described in the previous section, we present in this section the basic user interface and its parameters. Furthermore, some digital communication graphs designed in the GUI allowed us to show instantaneously the symbol constellation, the fading profile and the calculated BER/BLER vs. SNR, among others.

Apart from the visual interface, the GUI permitted to simulate under several channel conditions enabling or disabling the AWGN or Rayleigh fading. Another benefit of the GUI was the selection of the global parameters of a communication system such as type of modulation, number of symbols and frame, percentage of cyclic prefix, FFT size, among others. Overall, the GUI was our control panel to define the communication system parameters and analyze its performance using graphs and plots.

At the GUI section of the Appendix C, we describe the GUI from the transmitter to the receiver blocks and the BER/BLER calculus at the end. The complete LV GUI interface for our SISO system is illustrated in Figure 3.31.

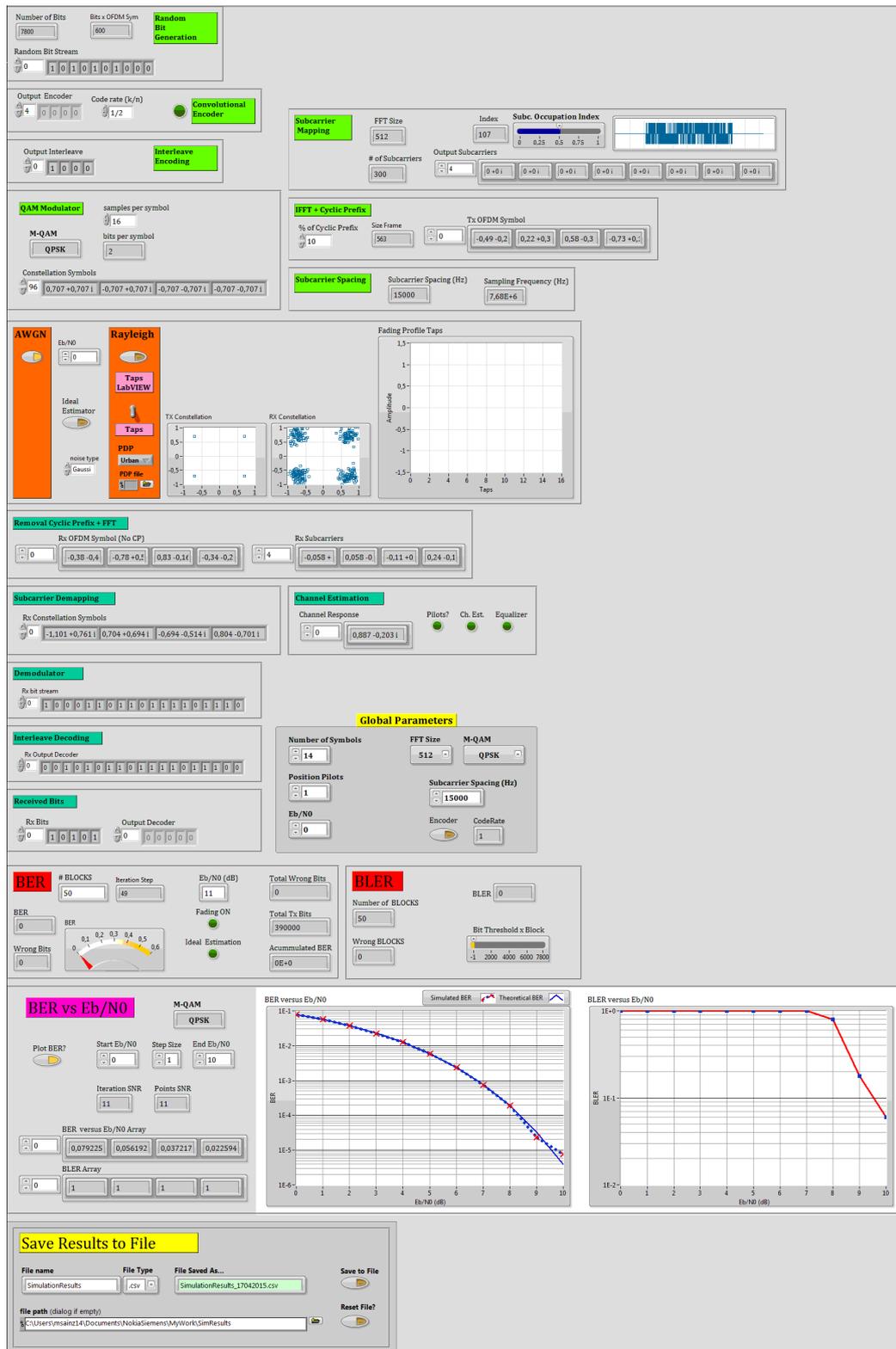


Figure 3.31: Complete GUI of the Digital Communication system in LabVIEW.

3.2 SIMO System

We present in this section the single-input multiple-output (SIMO) LV design. In this implementation we considered one transmit antenna and two receive antennas 1x2 SIMO system. Furthermore, the design allowed us to simulate more receive antennas increasing then the receiver diversity.

Having designed the SISO system, it was possible to extend its functionality modifying the receiver in order to receive multiple realizations of the transmitted signal. Thus, each received signal in each receiver antennas went through independent realizations of the channel, simulating different propagation paths. We illustrate in Figure 3.32 the SIMO concept.

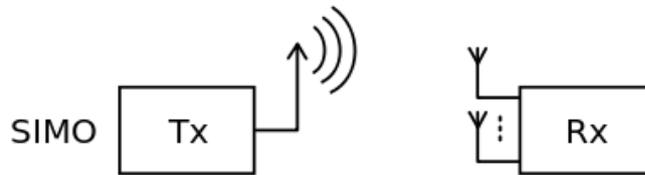


Figure 3.32: SIMO system design concept.

In order to implement the SIMO LV communication system firstly we simulated the propagation paths enhancing the channel blocks depending on the number of receive antennas. Secondly, we adapted our SISO receiver to simulate multiple antennas which captured the transmitted signal independently. Furthermore, the received data streams were used for estimate the channel in each antenna. Finally, we exploited the receiver diversity using a Maximal Ratio Combining (MRC) receiver, recovering an estimation of the transmitted signal which was used for BER/BLER vs. SNR performance calculations.

3.2.1 SIMO Model Design

As we implemented in the SISO case, the generated OFDM frame was delivered to the channel, who applied the Rayleigh fading and added the AWGN. For our SIMO design we used the same OFDM frame and we transmitted it through two independent realizations of the channel. That was realized by using a FOR loop for the complete wireless channel, as we illustrate in the Figure C.45 of the Appendix C. In this way, for each FOR iteration the transmitted signal was processed independently simulating independent paths. We note that in each loop iteration a new Rayleigh fading profile was generated and the labels [1] and [2] of the Figure C.45 defined the number of Tx and Rx antennas, respectively.

In contrast to the SISO receiver modules, the SIMO had to be modified in order to process the two received signals, independently. Then, it was necessary to combine both with its respective channel estimations within the MRC receiver. To do that we had to split our SISO receiver FOR loop in different ones. The reason behind that was that the loops iterations are independent and they do not share values between them, beside the use of register. However, in this situation we decided to simplify the design avoiding the registers.

For the implementation of multiple receive antennas we created serial FOR loops. The first one, obtained the several signals that went through different channel paths. Then, these signal were independently processes as we did before for our SISO system, symbol per symbol until the channel was estimated using the pilots symbols. As a result, we obtained as many received signals and channel estimations as the defined number of receiver antennas. Then the second FOR loop computed the estimated received signal combining all of them by means of the MRC receiver. The last FOR loop followed as in the SISO case, with the demodulation, channel and source decoding until the BER/BLER calculation.

To summarize, the new module added was the MRC channel equalization which exploited the receiver diversity improving the performance of the system. We depict the MRC labview realization in Figure 3.33, where the label [3] represents the different channel estimations, label [4] the received signals and [5] is the final estimated received signal using maximal ratio combining.

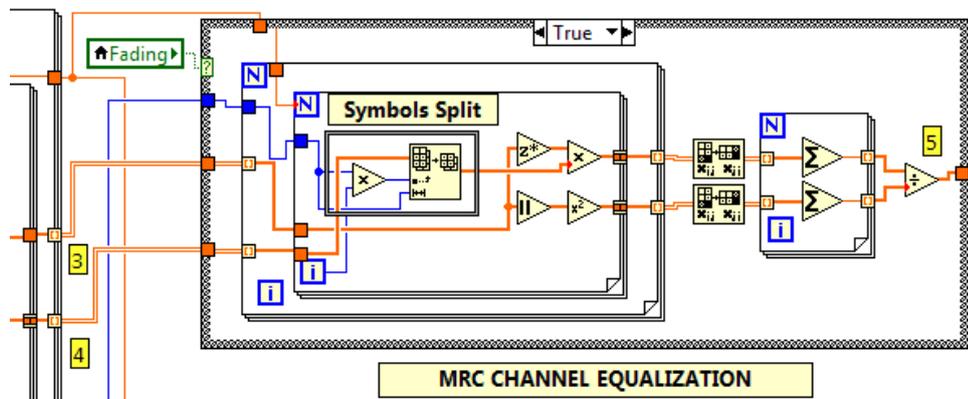


Figure 3.33: SIMO MRC Receiver LV Implementation.

Regarding to the MIMO LV implementation, we considered that adding more transmitting antennas/signals to our simulation would increase the complexity of the simulation delaying the achievement of one of our goals, the real world testbed scenario. Therefore, we decided to design our 2x2 MIMO LV scheme simultaneously within its integration into the USRP hardware.

3.3 USRP Design

We have described and illustrated in the previous section the LabVIEW simulation implementation of a digital communication system over a Rayleigh fading channel with AWGN. We calculated the BER and BLER and compared them with the theoretical expectations in order to validate our design. Furthermore, we enhanced our system from SISO to SIMO, exploiting the spatial diversity techniques frequently used in MIMO wireless communication system.

We present in this section the adaption and integration of our LV designs for the National Instruments USRPs which performed the end steps of a real communication system. As explained in the Appendix A, it was essential for the evolution of the wireless communication system to employ high frequency carriers in order to transmit successfully a signal through the air. This step called up-conversion (or down-conversion at the receiver) was realized automatically by the USRPs units. That was one of the benefits by using the LV for the testbed, a platform easily configurable to transmit and receive signals using the USRPs over a wireless channel.

Since the simulation of the channel was not needed anymore because our transmit medium was actually a real-world scenario, the transmitter and receiver LV design were split and connected to its respective USRPs. Thus, we adapted each design for a purpose, transmit or receive. In the next sections we describe the LV USRP implementation for the SISO and 2x2 MIMO systems, which were used for our real-world scenario testbed. By means of that, we addressed the effective potential of diversity techniques boosting the network throughput, as explained in Chapter 4.

3.3.1 SISO

The blocks necessary to create a Tx/Rx session using the NI USRP hardware were standard and offered by LabVIEW. These main blocks were named as *Open a Tx/Rx session*, *Configure USRP*, *Initiate Rx*, *Send Data*, *Fetch Data*, *Abort Rx* and *Close Tx/Rx Session*. We describe in this section the configuration modules that were related with our SISO testbed scenario. Further description of these modules is found in the NI LabVIEW Help [20].

Before introducing the Tx and Rx LV USRP design, we present in Figure 3.34 the core module for the configuration of our devices. As we observe, the *IQ rate*, *carrier frequency* and *gain* were defined within this block, which was employed at the transmitter and the receiver. Due to hardware specifications, the *coerced* outputs were an indicator of the real configuration values set during the Tx/Rx of data, which was done within a while loop.

Receiver Design

The receiver USRP was configured with the same USRP modules and some new ones which defined the respective receive parameters. It also differentiated from the transmitter in the way we adapted our SISO LV design. As the Tx/Rx USRP modules executed inside an endless *while loop*, it was clear that contrarily to the transmitter which sent the same packet, the receiver had to process each packet and hence the processes had to be implemented inside the while loop. That implied a trade-off between ability of data reception using the USRP modules and capacity of data processing using our LV implementation. In the SISO case, we managed to use the complete Rx LV design of our simulations without compromising the data acquisition performance.

We depict the USRP LV receiver implementation in Figure 3.36, where label [2] was the USRP configuration parameters module and label [3] the USRP data receiver module, which delivered in each while iteration new fetched data to our SISO Rx LV implementation.

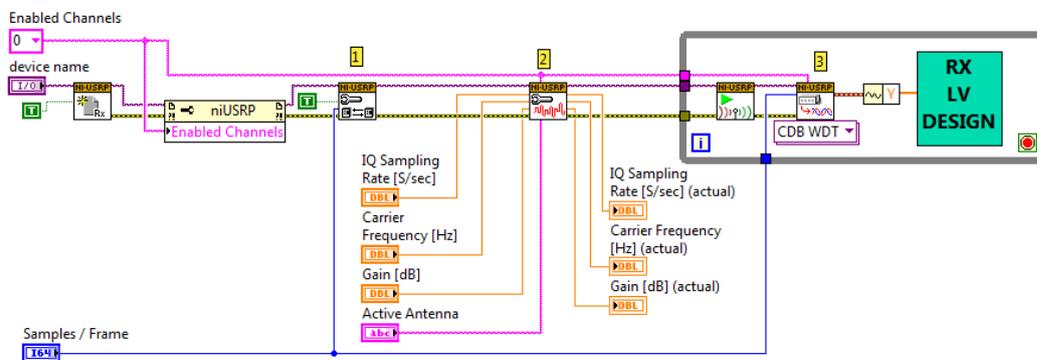


Figure 3.36: USRP SISO Transmitter Design.

We note for label [1] that at the receiver side was necessary to define a *number of samples* value. This was related to the size of the received array in each iteration, namely the resolution or sampling resolution. For example, if we set a transmission of 1 frame of 1 symbol of 512 subcarriers, then at the receiver we had to define 512 *Samples / Frame* or more in order to receive the same data. For lower values the subcarriers were merged in order to represent for instance these 512 subcarriers in 256 array samples.

3.3.2 2x2 MIMO

We re-used the SISO-SIMO LV designs and enhanced them for 2x2 MIMO, whose concept is illustrated in Figure 3.37. In the Tx implementation, we employed the same design but adding an additional signal (second Tx Antenna). In the Rx implementation, as explained before, the computing complexity of two received signals inside the while loop influenced the ability of data reception at the USRP module.

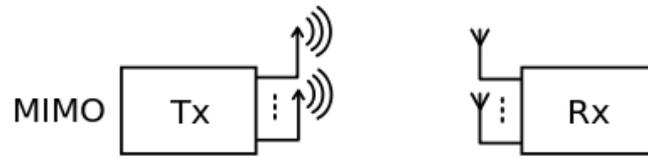


Figure 3.37: MIMO system design concept.

Since multiple computation processes inside the while loop delayed it, each iteration lasted more, decreasing the execution of the USRP data acquisition module. Therefore, we reused the essential modules of our SIMO LV design in order to adapt it to the MIMO requirements. In addition, we implemented a USRP MIMO receiver more suited for our real-world testbed objectives, which were focused in the channel estimation.

Transmitter Design

The LabVIEW 2x2 MIMO Tx design incorporated an additional branch from the *Subcarrier Mapping* module to the *Subcarrier Spacing*, which delivered the data to the channel. In our testbed, we aimed to transmit CAZAC pilots to estimate the channel. The procedure we followed as explained in Chapter 2, was to send interleaved pilots through these two branches, namely Tx Antennas. In this way, at the receiver we obtained the channel matrix.

Firstly, our 2x2 MIMO transmitter split our originally CAZAC pilot sequence into two signals, mapping them following the interleaved pattern. Then, the two independent frames of pilots were given to the MIMO USRP module inside the *while loop*. We present the bifurcation of the signal into the two branches of our re-edited LV transmitter in Figure C.46 at Appendix C.

Similarly to the SISO USRP case, we concatenated the frames of both signals in order to enlarge the size of the transmitted data arrays. We show in Figure 3.38 the MIMO USRP LV design, where label **[1]** and **[2]** referred to the signals transmitted through the two USRP antennas. Then, label **[1*]** and **[2*]** represented a concatenation of several frames of these respective signals, which were added together inside a 2D array. This array was fed into the MIMO Tx USRP module, which was configured to transmit each signal through the defined USRP devices. As a result, in each iteration the USRP antennas were transmitting interleaved pilots continuously.

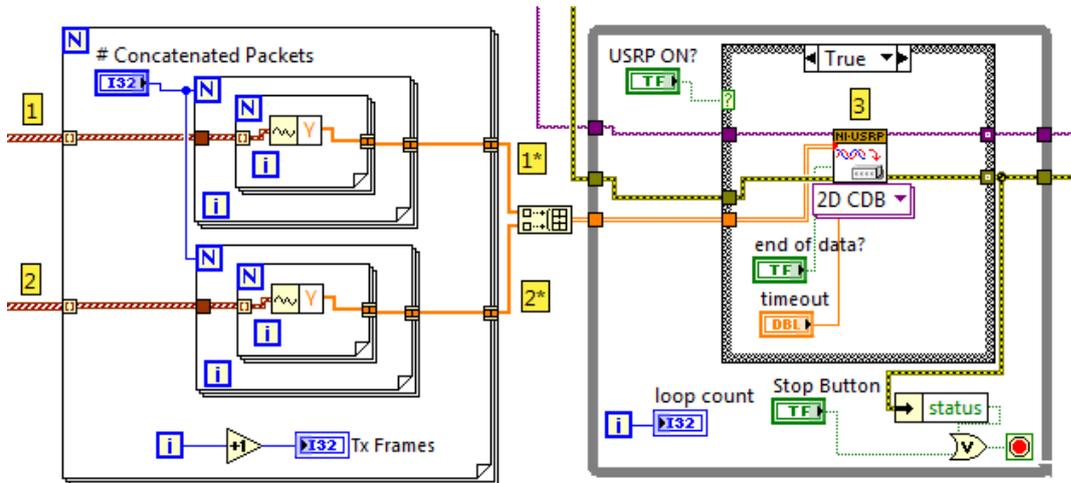


Figure 3.38: USRP MIMO Transmitter Design.

Besides the pilot implementation, we were able to generate frames of 1 pilot and several data packets and send them through the two MIMO antennas. In this case, the same modulated data was transmitted over the two antennas while the pilots were interleaved. However, for our testbed objectives the MIMO transmitter was restricted to pilots frame.

In order to synchronize the devices for MIMO transmission the USRP had to be configured specially for that purpose. The main parameters that made possible the synchronism are presented in Figure 3.39, where two main modules were essential. First, the *Start trigger time* module and the *Configure Clock* module. In addition, the parameters *Reference Frequency Source* and *Timebase Clock Source* had to be set to *Internal* clock for one device and *MIMO* clock for the other device. In this way, one device set the clock internally and the other using the MIMO cable followed the timing imposed by the first device. Only in this way was possible to transmit synchronized over the two antennas.

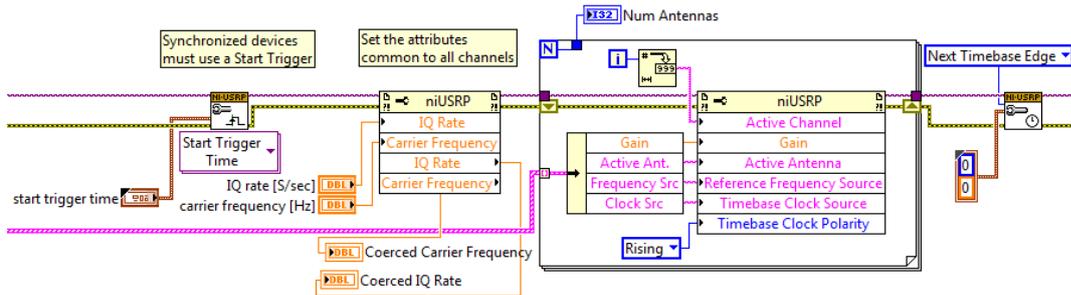


Figure 3.39: USRP SISO Transmitter Design.

As we described, the main parameters of the USRP devices were set for MIMO transmission in the GUI, as shown in Figure 3.40.

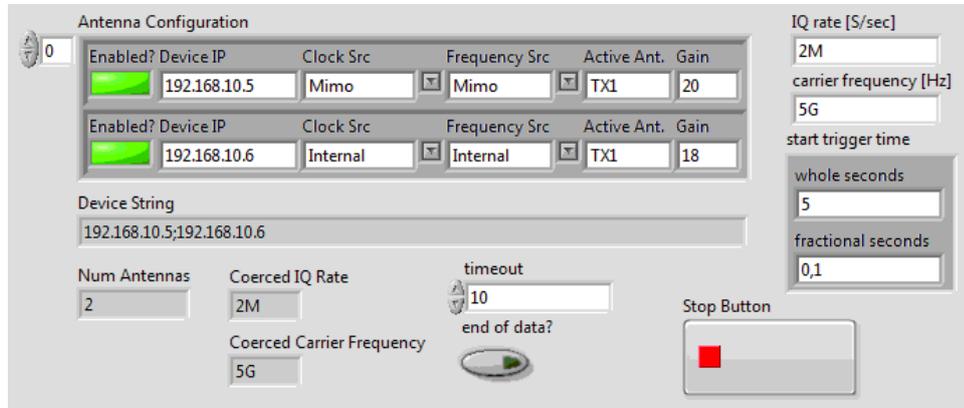


Figure 3.40: USRP SISO Transmitter Design.

Receiver Design

The 2x2 MIMO LabVIEW receiver design differed from the complete SISO-SIMO designs. Instead of adapting these designs to the USRP devices, we realized a basic receiver with an extra feature, the spectrum analyzer. Since it was important to obtain information about the transmission over the channel, we built inside the design a spectrum monitor, which gave us evidence of the transmission of the OFDM signal at the desired carried frequency and subcarrier spacing.

We designed the spectrum analyzer in the upper part of our 2x2 MIMO USRP GUI, illustrated in Figure C.47 at Appendix C. In this Figure the plotted spectrum referred to a 2x2 MIMO transmission-reception, where one symbol was represented in the spectrum analyzer so that we distinguished the guard band and the 300 OFDM subcarriers.

Having developed the spectrum analyzer, we implemented the receivers Rx1 and Rx2, which processed the obtained 2D array from the USRP fetch data module. We incorporated the essential modules of the SISO-SIMO LV schemes designed previously into our MIMO design. Afterwards, knowing the transmitter parameters such as the pilots pattern and position, subcarrier mapping and number of frames transmitted, we processed each signal independently using the pilots to obtain the respective channel estimation values. At the Rx1 we used the pilots pattern of the Tx1 and Tx2 to calculate the channel estimation H11 and H21, respectively. Then, we extract from the Rx2 the respective channel estimation parameters.

As a result, we established the channel matrix which was updated for each pilot symbol. We show the LV design of the Rx1 in the Figure 3.41. The same scheme was used for the Rx2, which was implemented next to this block. We depict the complete USRP MIMO Rx Labview design in Figure C.48 at Appendix C.

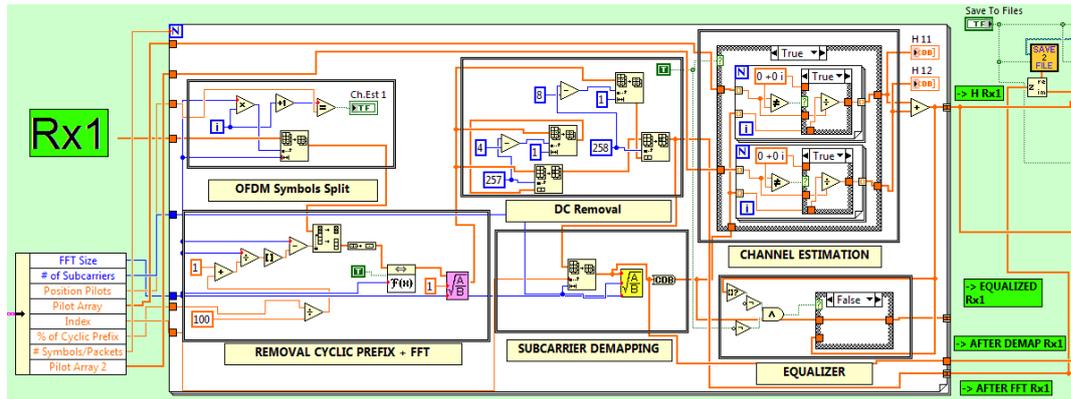


Figure 3.41: USRP MIMO Rx1 LV Design.

We conclude by reviewing the configuration setup of the USRP receiver devices in Figure C.49 at Appendix C. We point out the importance of the *Start Trigger Time* module and the *Reference Frequency* and *Timebase Clock* parameters, which enabled the MIMO synchronization between devices. That was important in order to match the occupied subcarriers with the pilots pattern and then calculate the channel estimation accordingly to the position of the CAZAC transmitted pilots pattern.

To summarize, we presented in this chapter the LabVIEW implementation of a digital communication system based on OFDM frames over a fading and noisy channel. We described the communication blocks designed from the transmitter to the receiver part, including the wireless channel. Furthermore, we developed in LV the performance analysis tools to validate our simulations. Then, we adapted our design for the NI USRP devices, transmitting and receiving successfully OFDM frames over the wireless channel within a SISO and 2x2 MIMO environment.

Chapter 4

SDR and USRP Real-World Scenario Testbed

In this chapter we present and describe our testbed scenario where we employed our LabVIEW SDR designs and USRP devices to test some of the MIMO techniques capable of boosting the network throughput. First, we describe the real-world testbed scenario where we set different physical configurations in order to estimate the channel.

We conclude analyzing the performance of the MIMO links for the different scenarios. We also illustrate the LV graphical tools designed in order to observe and evaluate the transmission characteristics such as spectrum and the receiver performance such as frame synchronism, subcarrier demapping and channel estimation.

4.1 Scenario Description

We realized our MIMO testbed in an indoor office scenario, where the main elements were tables, chairs, small fridge and other small elements such as notebooks. Moreover, a big window connected to the outside and a thick wooden door gave access to the corridor. Then, we establish 2x2 MIMO transmissions between two nodes (Tx and Rx) for different test setups in Line-of-Sight (LOS) and Non-Line-of-Sight (NLOS). We depict the indoor office testbed scenario in Figure 4.1.

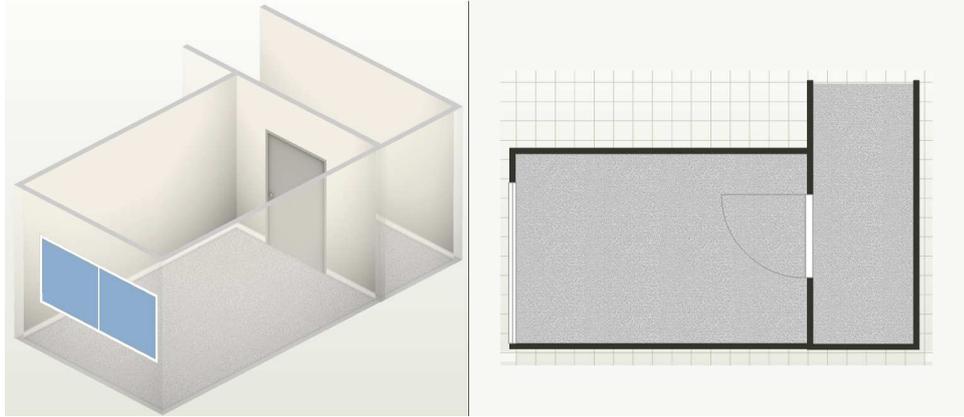


Figure 4.1: MIMO Indoor Office Testbed Environment.

In order to obtain uncorrelated realizations of the channel estimation, we defined three main node scenarios. Each node was configured for transmission or reception using two antennas synchronized for MIMO mode.

In the first scenario, the two nodes were placed inside the room in direct LOS without any obstacle between them. We set also three different positions for the transmitter node inside the room, which consisted in translations of 45 degrees to the left and right side from the point view of the receiver.

The second scenario was defined with the receiver in the center of the room and the transmitter in the corridor, while the door was closed. We categorized this configuration as NLOS. For the last scenario, we keep the position of the nodes but we opened the door.

We illustrate the three testbed MIMO scenarios in Figure 4.2.

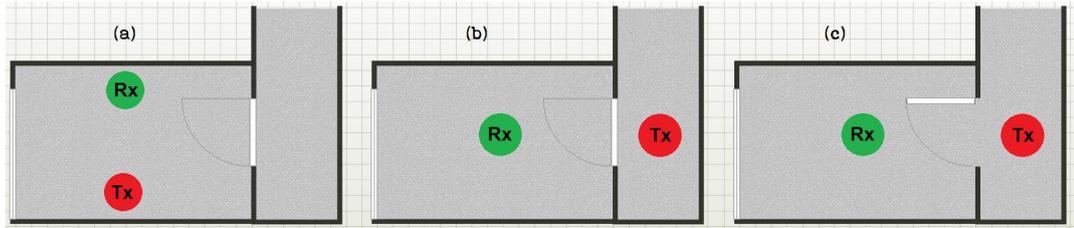


Figure 4.2: Testbed MIMO Scenarios. (a) LOS, (b) NLOS (door closed), (c) LOS (door open).

Regarding the nodes, each of them was utilized as transmitter or receiver. The NI USRP devices were mounted on a trolley, which had the two antennas fixed on its top corners. Since the configuration of the USRP devices was done within the SDR, the installation of them on the trolley consisted basically on the power supply, the connection to the antennas and the LAN cable, which connected the USRPs with the computer using Ethernet. We show the 2x2 MIMO node in Figure 4.3.

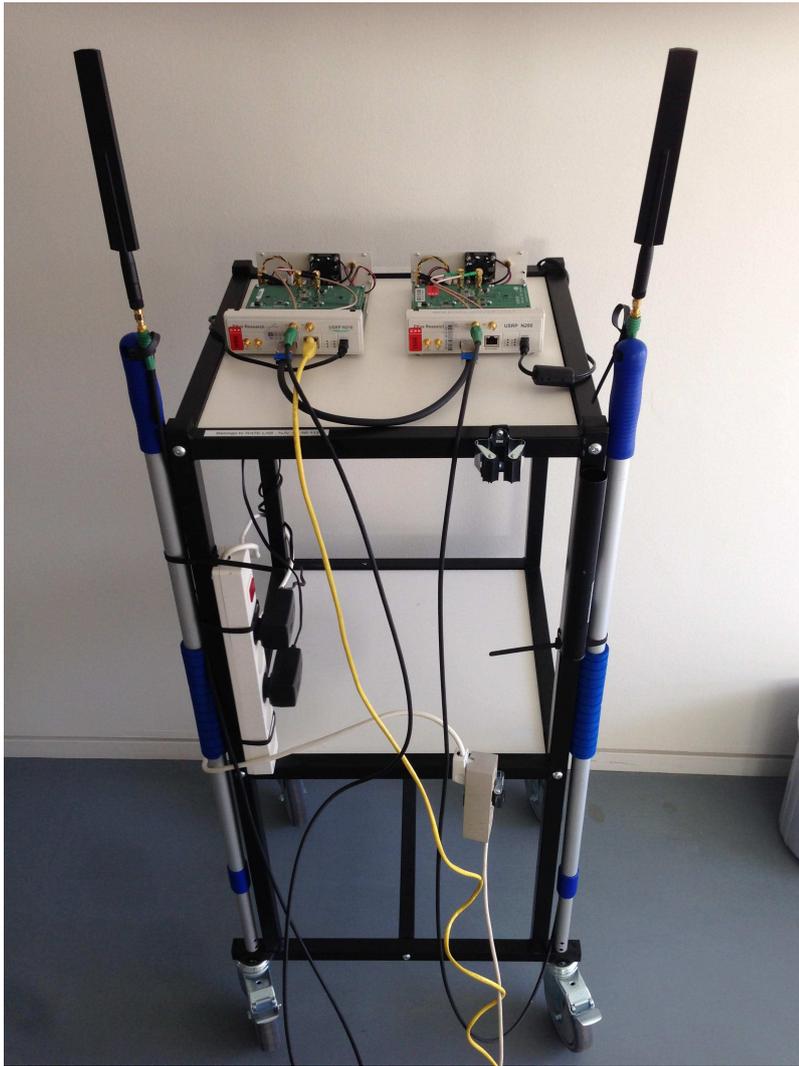


Figure 4.3: USRP MIMO Transceiver Tesbed Node.

Since the USRP SISO transmission employed one device in order to transmit or receive one signal, there was no need of the MIMO cable. However, for the 2x2 MIMO testbed the computer was only connected to one device since it has only one Ethernet board. Therefore, the MIMO cable made possible the control of both USRPs and established the synchronism as explained before. We present the 2x2 MIMO USRP devices and its setup in Figure 4.4.

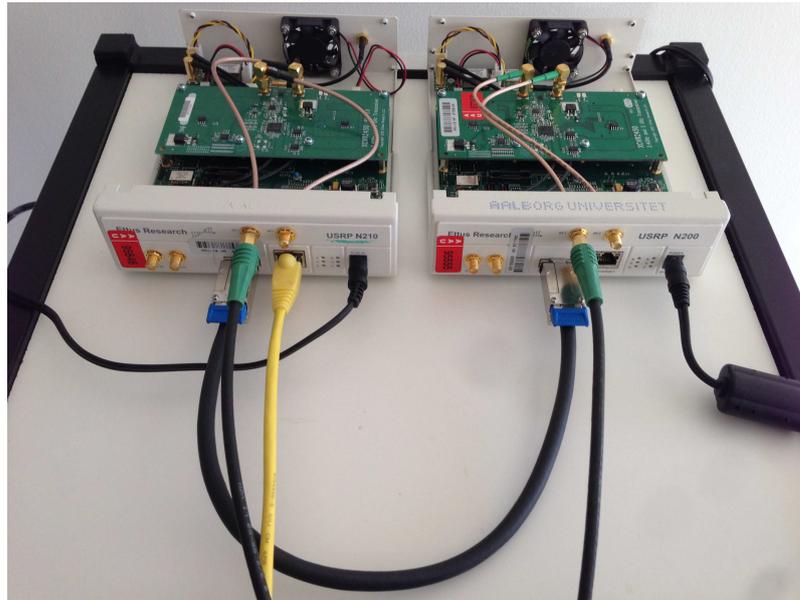


Figure 4.4: 2x2 MIMO Configured USRP Transceivers.

We observe that the LAN cable (yellow) was connected to one USRP device, which had to be configured with the reference clock as *Internal*. The second device, which had no LAN cable was setup with the reference clock as *MIMO*. Then, the MIMO cable (thick and black), connected both devices. These values were introduced in the USRP GUI of the LV implementation. Finally, each USRP was connected to the power supply and the cables with green end were the input/output to their respective antennas. In particular, the USRP N200 boards operated over 2.4 GHz and 5 GHz band. We selected as carrier frequency 5 GHz, with a bandwidth of 2 MHz and transmit power of 20 dBm. Note that we attached the USRP N200 Datasheet in the Appendix B.

To summarize, within the different scenarios we connected two independent computers to the two nodes, which transmitted or received in MIMO mode. Then, at the receiver side, we analyzed the captured data, processed it and we calculated the channel estimation using the pilots. In the next section we describe the parameters defined for our testbed, such as carrier frequency and IQ rate. Furthermore, we present the MATLAB calculations realized in order to evaluate the performance of the channel precoding diversity technique, which was the goal of the testbed.

4.2 Performance Analysis

The main parameters which defined the USRP testbed are shown in the Table 4.1. These values were chosen considering the range of the USRP devices and the performance limitations of the computers, which stopped due to overflows or underflows. Some of them were defined by default. We selected the carrier frequency within the range of the USRP devices. For the MIMO system, we selected a IQ rate of 2 MHz, which avoided overflows errors. We note that for the USRP SISO transmissions we achieved IQ rates of 12,5 MHz.

In addition, we transmitted symbols of 512 subcarriers, which were concatenated into arrays of 1000 OFDM symbols. At the receiver we configured the number of samples according to the defined array size at the transmission, which was made of 1000 symbols of 512 subcarriers.

Parameters	USRP Tx	USRP Rx
	Value	Value
IQ Sampling rate [S/sec]	2M	2M
Carrier Frequency [Hz]	5G	5G
Number of Samples	-	512000
Active Antenna	TX1	RX1
Number of OFDM Symbols	1000	1000
Start Trigger Time		
Whole Seconds	5	2
Fractional Seconds	0,1	0

Table 4.1: Main USRP Configuration parameters

The transmitter LV implementation did not have any analysis tool, since it transmitted continuously the frames over the selected carrier frequency. In order to validate the transmission and check that the signal exhibited an OFDM scheme we employed an external spectrum analyzer. Afterwards, we exploited the capabilities of the SDR LabVIEW in order to reuse some schematics that recreated a spectrum monitor. As a result, we enhanced our transmitter with the spectrum analyzer which offered us a direct feedback about the transmitted signal and its immediate reception at the receiver. We illustrate this feature in the Figure 4.5 where we observe the two signals received over the MIMO channel, in particular the guard band and the 300 subcarriers centered.

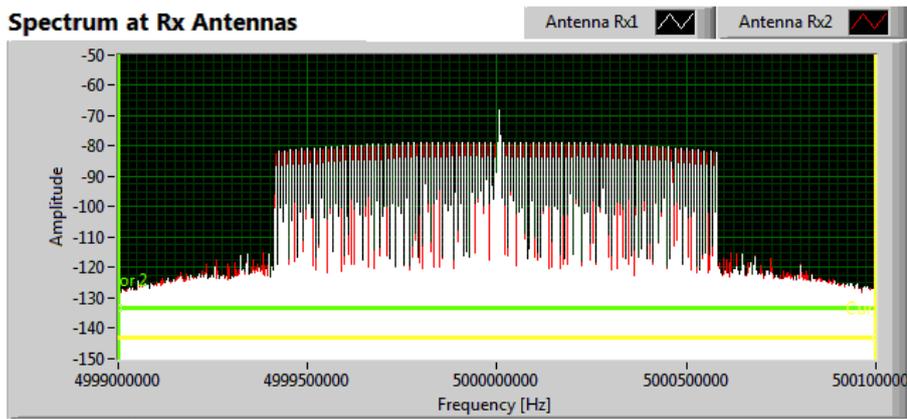


Figure 4.5: Spectrum Analyzer LV implementation at the receiver.

Defining the 1000 symbols of 512 subcarriers, what we obtained after the FFT of the received array is plotted in Figure 4.6. We observe a zoomed section of the 1000 concatenated symbols, in this case 4 OFDM symbols. Moreover, we implemented our receiver in a way that the long array was split into its respective symbols, and afterwards demapped as illustrated in Figure 4.7, where we distinguish the subcarriers without guard band, and in this case carrying interleaved pilots.

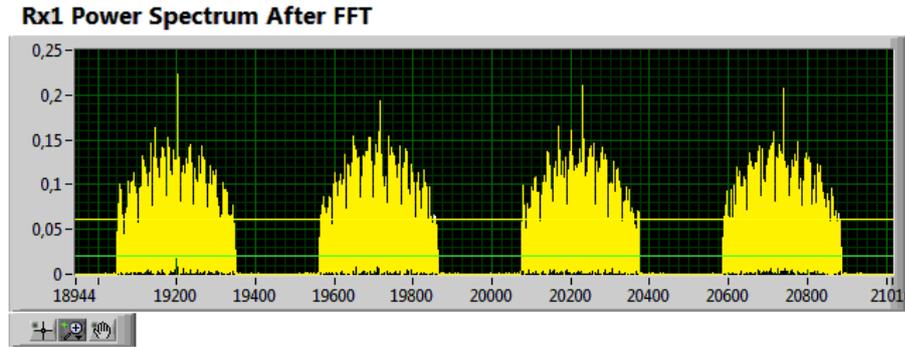


Figure 4.6: Power Spectrum after FFT of the received array.

We implemented the interleaving pilot sequence so that at the receiver we noticed both signals mixed together. That interleaving was the reason because in our channel capacity calculations we did not consider any signal interference.

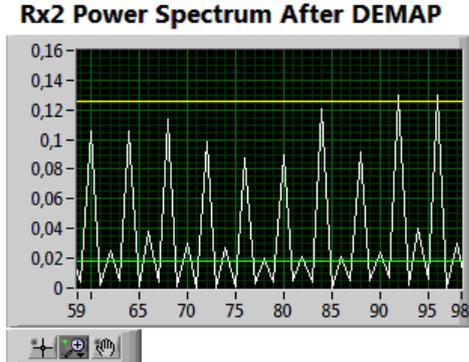


Figure 4.7: Power Spectrum of the OFDM symbol without guard band.

After extracting the guard band, we processed the symbols and estimated the channel matrix, which was saved into a file. With the channel matrix we computed the throughput of the channel for the precoding matrices. As explained in Chapter 2, we exploited the MIMO diversity by obtaining the information of the channel and then coding our transmitted signal in order to make it more robust to the channel distortions. Since our objective is to evaluate theoretically the performance of the different precoding matrices, we realized our channel capacity calculations in MATLAB with the channel matrix obtained in our real-world testbed scenarios.

For the channel capacity calculation, we were provided with an script that applied for each subcarrier a fixed precoding matrix to our channel matrix. Then, we calculated the SINR which was used to obtain the throughput. We adapted this script for our testbed and incorporated new features, such as the selection of the precoding matrix that offered the highest SNR, maximizing the throughput. We presented the edited scripts for the MATLAB throughput calculations in the Appendix D.

To conclude, for our performance evaluation boosting the network throughput we set a fixed matrix for all the testbed scenarios and calculated the channel capacity. Then, we utilized the script feature that selected the precoding matrix that maximized the throughput. At the end, we contrast both results on a cumulative distribution function for our 1000 transmitted symbols within the real-world testbed scenarios.

Chapter 5

Results

In this chapter we present the most important results of our simulations and real-world testbed scenario, realized with LabVIEW and the USRP Hardwares.

5.1 BER Performance Analysis

In this section we present first the SISO LV implementation results, such as modulations, fading and AWGN channel, and BER calculations. Finally, we depict the results for our SIMO LV simulation design.

5.1.1 SISO

In order to validate our LabVIEW implementation, we obtained in our GUI several plots at key points of our communication system. The most relevant were related to the available modulations and the channel realization. Furthermore, the addition of the cyclic prefix was essential in order to avoid ISI in our OFDM system. Therefore, we provide first some of the results of the performance of our implementation. Then, we contrast the BER results with the theoretical, over AWNG and fading channel.

QAM modulations over fading and AWGN channel

Having designed the transmitter, channel and receiver we evaluated the performance of the complete system by using different modulations and representing its Tx and Rx constellation. We illustrate the effects of the AWGN channel on a QPSK modulation in in Figure 5.1. Since we captured the constellation for a E_b/N_0 of 15 dB, we identify at the receiver side the same constellation.

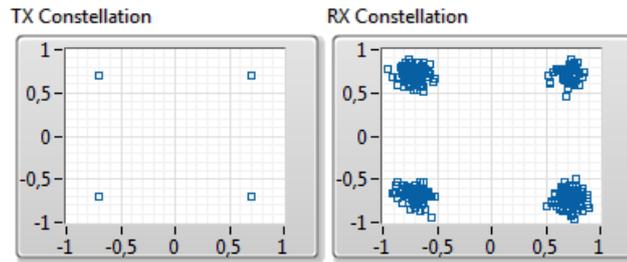


Figure 5.1: Tx and Rx QSPK constellations over a AWGN channel.

While the modulation increased, the received constellation suffered more from the distortions of the channel generating more errors in detection. We depict as example a 64-QAM modulation over AWGN channel in Figure 5.2 .

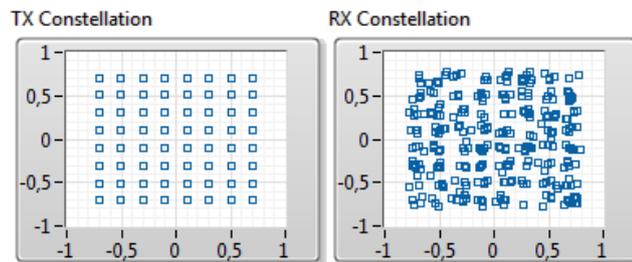


Figure 5.2: Tx and Rx 64-QAM constellations over a AWGN channel.

On the other hand, after applying the fading to our transmitted signal, we tested our design by varying the cyclic prefix length and observing the equalization results. In Figure 5.3 we show how the ISI corrupted the received constellation on a free noise channel.

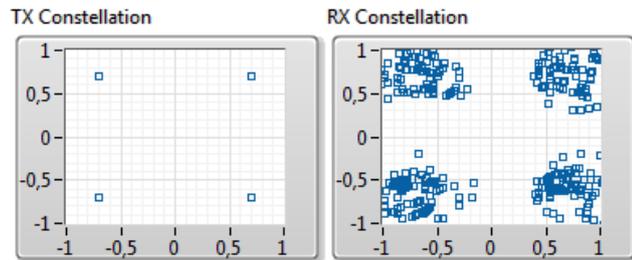


Figure 5.3: QPSK constellations over a fading channel and CP=0%.

By increasing the cyclic prefix, the equalized received signal matched the originally transmitted, as Figure 5.4 corroborates. As a result, we consolidate our LV design testing different modulations over the channel and representing the results.

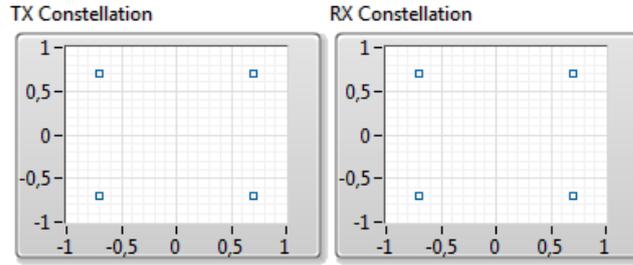


Figure 5.4: Equalized QPSK constellation over a fading channel and CP=10%.

BER Performance Analysis over AWGN channel

We defined a AWGN channel model without fading and simulated different modulation schemes. Then, we plotted the BER vs. SNR graphs in contrast to the theoretical values. Thus, we verified the behavior of the design of our LV digital communication system.

For our available M-QAM modulations ($M=4, 16, 64, 256$) we generated random bits and transmitted them over the noisy channel. Then, we reversed the transmitter steps at the receiver obtaining the estimated bits. Finally, we calculated the bit error rate for a defined SNR. We repeated the process for different SNR values and plotted the results for each of the modulations. We group all the calculations and plots in contrast to the theoretical values in Figure 5.5.

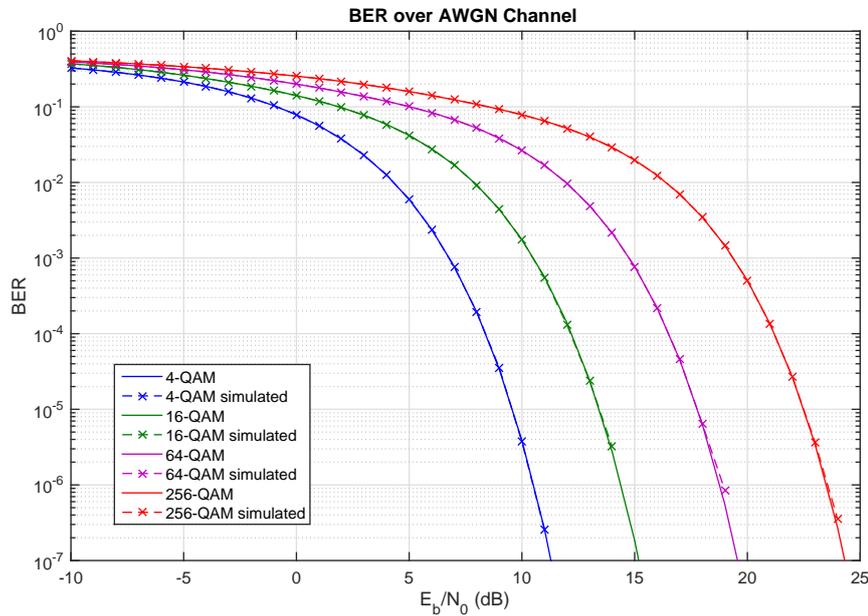


Figure 5.5: BER over AWGN channel for QAM modulations.

BER Performance Analysis over Fading channel

We applied the fading profile using the taps generated from the power delay profile presented in Table C.1. Then, we utilized the ideal estimator to equalize the channel. We performed the same procedure to plot the BER vs. SNR graphs in contrast to the theoretical Rayleigh fading profile. Those results for the M-QAM modulations are illustrated in Figure 5.6.

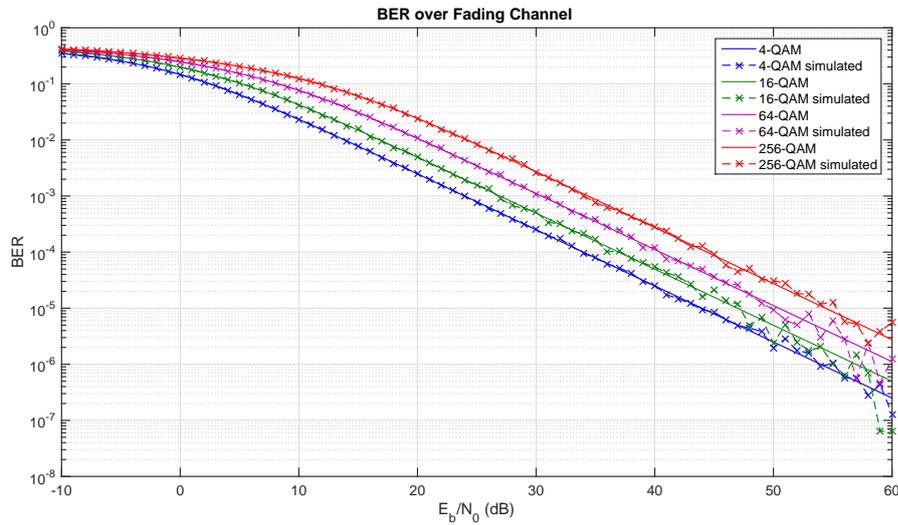


Figure 5.6: BER over Fading channel for QAM modulations.

5.1.2 SIMO

The SIMO LV implementation was characterized by the addition of more than one receive antennas. As we explained before, the internal design of the receiver was enhanced to multiple Rx antennas and maximal-ratio-combining (MRC). Since this implementation was a step-in-between our SISO and MIMO system, we selected a QPSK modulation and calculated its BER for different number of receiver antennas (n_{Rx}). Then, we compared them with the theoretical results for validation purposes, as Figure 5.7 presents.

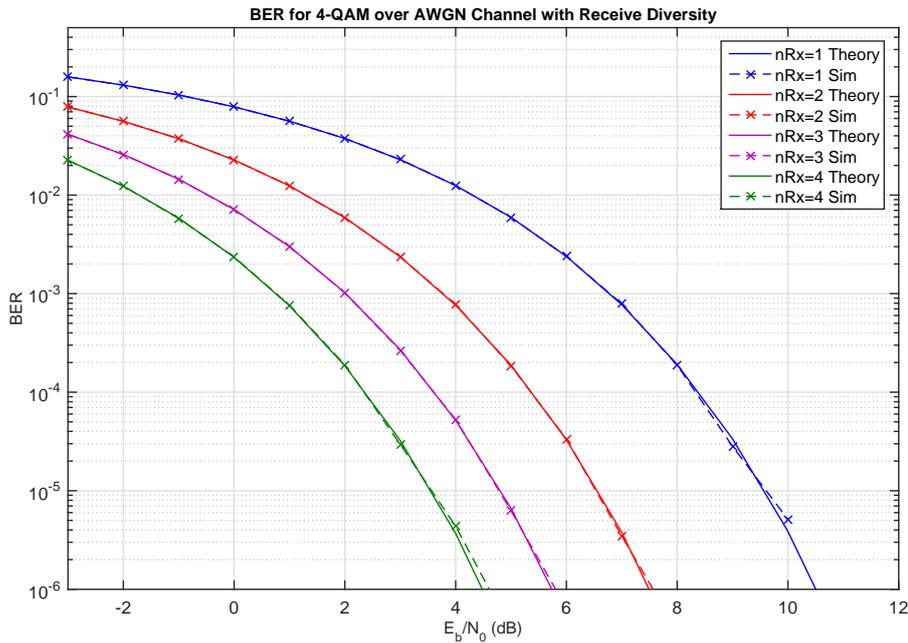


Figure 5.7: BER over AWGN channel for 4-QAM modulation and Receiver Diversity.

5.2 Real World Testbed Performance Analysis

In this section we examine the results of our LabVIEW MIMO design and the MIMO diversity techniques to boost the throughput in our real-world scenario testbed. First, we present important validation results that determined the correct configuration of the MIMO USRP devices in order to transmit and receive in synchronism. Finally, we present the theoretical results that addressed our goal of the channel capacity increase while using precoding techniques which exploited the diversity of the system.

5.2.1 2x2 MIMO Scenario

We illustrated how our spectrum monitor at the receiver captured the transmitted signal. Furthermore, we observed in the external and also embedded spectrum analyzer, that the transmitted signal had an OFDM scheme. Finally, we validated our 2x2 MIMO system by presenting the following results that justified a correct performance.

After defining a number of transmission symbols of 1000 and a FFT size of 512 with 300 centered subcarriers, we assumed that our concatenated symbols were synchronized if the guard band and the position of the occupied subcarriers were preserved. In addition, in order to verify the MIMO transmission and reception, Rx1 and Rx2 had to fulfill these requirements simultaneously.

First, we obtained an array of 512.000 samples, representing the 512 subcarriers concatenated for 1000 transmitted symbols. We saved it after the FFT and plotted the result. As observed in Figure 5.8, the symbols were concatenated keeping the synchronism between symbols from the Rx1 and Rx2.

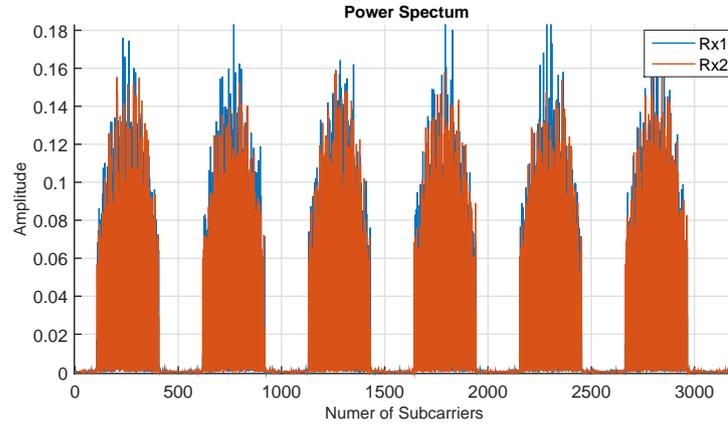


Figure 5.8: Received Signals at 2x2 MIMO receiver.

In order to check the subcarrier synchronism we identified the start position of our first pilot subcarrier, the last position of our 300 subcarriers and the position that marked the end of the symbol. Adding again the guard band, we proved that the guard band was preserved. We illustrate these verification results in Figure 5.9.

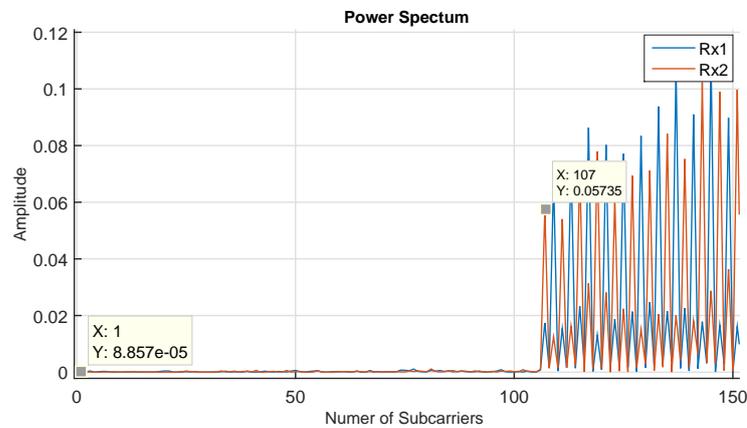


Figure 5.9: Subcarriers of the Received Signals at 2x2 MIMO receiver.

At the transmitter, we added 106 guard band subcarriers at the beginning of the 300 subcarriers and 106 more at the end. As a result, we obtained $106+300+106=512$ subcarriers, which formed one symbol. We identify in Figure 5.10 the concatenated symbols and its subcarrier spacing, for 3 symbols of 512 subcarriers, which 300 centered subcarriers were occupied by the pilots.

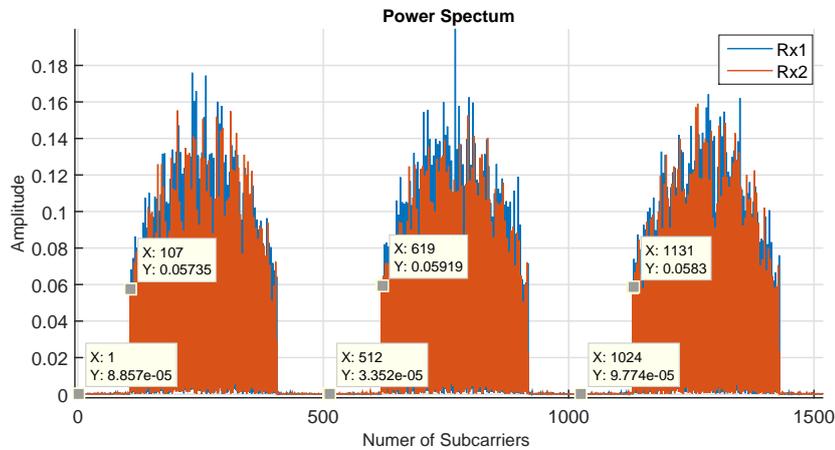


Figure 5.10: Subcarriers of the Received Signals at 2x2 MIMO receiver.

The last result addressed the interleaving of the pilots. Following the MIMO pilot interleaved sequence presented in Chapter 2, Tx2 transmitted pilots spaced with 3 zero samples, while Tx1 the same pattern but with 2 samples offset. Then, the expected received signal was a merge of Tx1 and Tx2, giving a pilot sequence spaced only with 1 zero sample. We illustrate Rx1 and Rx2 in Figure 5.11, particularly the beginning of the pilot subcarriers of the first received symbol. We notice that the first pilot from both antennas started at the sample number 107. Then, the next pilot that followed was numbered as 109 and then 111 and consecutively. As a result, the interleaving of pilots was recognized at both received signals. After these verifications, the demapping process was done correctly since the guard band was identified and deleted. Finally, the received pilot sequence for each symbol was extracted for the computation of the channel matrix, which was saved into a file for the further calculations in MATLAB.

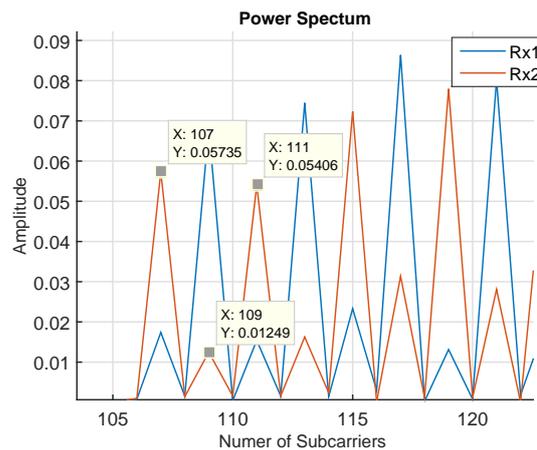


Figure 5.11: Subcarriers of the Received Signals at 2x2 MIMO receiver.

5.2.2 MIMO Channel Precoding

In this section, we present the performance results of our testbed experimentation. The calculated throughput was obtained by Shannon mapping, assuming no interfering channels. Furthermore, the highest order modulation of 256-QAM with a maximum spectral efficiency of 8 bits/s/Hz was considered for the computation of the maximum achievable throughput using the Shannon-Hartley formula.

We applied the precoding matrices of the Table 2.2 (see Chapter 2 Section 2.9.3) to the channel matrix. Then, by means of the MATLAB scripts presented in Appendix D we computed the SINR of the subcarriers for each of the precoding matrices of the codebook, which were multiplied by the estimated channel matrix. Particularly, we defined a fixed precoding matrix for all the scenarios and a automatic selection of the matrix that maximized the SNR.

We illustrate the results in Figure 5.12, with the empirical cumulative distribution function (CDF) for our different testbed scenarios. We note that the throughput is related to the 2 MHz bandwidth used in our testbed.

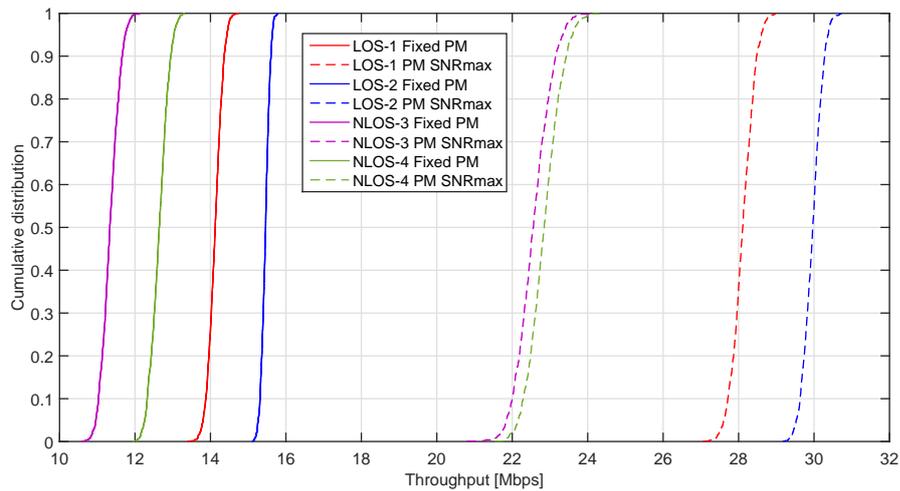


Figure 5.12: CDF of the throughput for a 2x2 MIMO Antenna Configuration using Precoding Matrix (PM) and 2 MHz bandwidth on different testbed scenarios (LOS and NLOS).

Chapter 6

Discussion

In this Chapter we summarize and interpret the most important results derived from our simulations and experimental testbed. Moreover, we discuss the relationships, trends and generalizations among the results.

The implementation of a basic digital communication system using the SDR LabVIEW has demonstrated design flexibility in order to simulate diverse scenarios. We modeled the communication by changing easily parameters before and during simulations. As we presented in Figures 5.1 to 5.4, we appreciated directly the effect of our simulation parameters on our transmitted data. Viewing these representations, we corroborated the background theory for our OFDM transmission by observing the modulation of bits data, its OFDM implementation and the wireless channel emulation.

We designed four modulation schemes, namely 4-QAM, 16-QAM, 64-QAM and 256-QAM. We confirmed that the cyclic prefix removed the ISI of our subcarriers, while the equalizer corrected errors of the received signal due to the channel distortions. While the transmission was running, we compared the generated constellation with the received in embedded graphs inside our LV desing. That was of important relevance for the validation of our LV implementation during the development process.

The SISO LV design resulted in long debugging time and trial-error simulations. One of the main reasons, were the black boxes of LV. We understand by black boxes, the communication modules available in the SDR that were easily implemented following its setup specifications. However, we ended in two kind of errors that delayed the developing time.

The first type of errors was related to the simulation work-flow. Since LabVIEW employed modules connected with wires, the data flowed from block to block through these wires but the computation processes inside the modules were unknown. Easily, we assumed that each module realized the basic operations we wanted to implement. However, we ended replacing these modules by our own basic ones with elemental computation processes that were previously tested as independent units. As a result, we verified our implementation in spite of a higher design time.

The second type of errors referred to the integration of all the blocks together and the addition of new features. Since each module and process was adapted to its previous ones,

the use of new variables and communication alternatives, such as type of QAM modulation, number of subcarriers or frame generation derived in simulation interruptions or non expected output results. To solve that, after implementing the complete communication system we observed every input and output of the modules in order to verify them and also to observe how the signal was processed in a real wireless communication system based in OFDM symbols over fading and AWGN channel.

We implemented a LV design that simulated a transmission of random bits modulated in QAM and mapped into an OFDM symbol. The OFDM symbols were packed into frames that were sent through the channel. Then, at the receiver we reversed all the processes performed in the transmitter and we obtained an estimation of the transmitted random bits. Finally, we enhanced our LV design in order to calculate the BER and represent it versus the SNR.

We infer from our SISO LV design results in Figure 5.5 that for our implemented modulations the simulated BER vs. SNR matched the theoretical one. By increasing the number of transmitted bits, we noted that the resolution of our simulations increased. On the other hand, for short frames we appreciated that the simulation trend followed the theoretical results. We recognize this trend in Figure 5.6, where we illustrated the BER over fading channel. After the design of the fading taps following the PDP for an urban scenario, we observe in the results that simulated and theoretical plots matched for all the modulations.

Having the SISO verified, we appreciated how the receive diversity of our SIMO LV design improved the system performance by decreasing the BER. Furthermore, we observed that the highest SNR gain was obtained from the 1x1 SISO to the 1x2 SIMO channel, with approximately 2 dB of gain. For 3 and 4 receive antennas, the achieved gain was 1 dB and 0.5 dB, respectively. We proved that the diversity improved the performance in relation to the diversity order. Furthermore, we confirmed that the gain trend decreased for higher number of receive antennas, as we theoretical exposed in Chapter 2 Figure 2.39. Contrasting the simulated and theoretical results for receive diversity on a 4-QAM (QPSK) modulation we interpreted that the SIMO LV design using MRC receivers was correctly implemented.

The 2x2 MIMO was implemented within the testbed scenario. After adapting our LV design for the transmitter/receiver USRPs over a SISO channel, we attached directly two devices more with the MIMO cable to the nodes. Firstly, we did not configure the transmission for MIMO mode. That meant that there was no reference clock source defined and each device transmitted/received with its own internal clock. As a result, with our spectrum analyzer we observed fluctuations among our pilot subcarriers. Moreover, the position of the subcarriers within the received array was random.

We deduced that we had to synchronize all the devices in order to transmit and receive in a real 2x2 MIMO scenario. We achieved it by defining the internal clocks in the configuration parameters of the USRPs, as described in the Chapter 5. In Figure 5.10 we identify 3 concatenated symbols within a frame. With a defined guard band of 106 zero subcarriers at both sides of the pilot subcarriers, we located that when the devices were synchronized for MIMO, the subcarrier positions was preserved. Furthermore, we located the first pilot at the position 107, as expected. Then, for 512 subcarriers symbols, the first pilot of the second symbol started at 619, namely $107+512$. We observe the same from the third symbol first pilot at position 1131. We confirmed then that both receiver antennas were receiving in MIMO mode and the recovered frames preserved the symbol subcarrier structure.

On the other hand, in order to validate the MIMO transmission of interleaved pilots (see Chapter 2 Section 2.9.3), in Figure 5.11 we recognize the received signals at Rx1 and Rx2. We appreciate how pilots from Tx1 (P1) and from Tx2 (P2) were transmitted in MIMO mode so that at each receiver antenna the pilots were interleaved as expected. Furthermore, we observe an amplitude correlation between the P1/P2 and the receive antennas Rx1/Rx2. In other words, one of the transmitter antennas had better directivity with a receive antenna than with the other. The same occurred with the other transmitter antenna. As a result, we identify how P1 and P2 which were obtained together in the same signal have almost five fold of amplitude difference. While moving the nodes we influenced this relationship and obtained different testbed realization in order to simulate uncorrelated channel estimation scenarios. In spite of that, the testbed scenarios were very similar and the distance between nodes small.

In the different scenarios of our real-world testbed, we transmitted 1000 pilots symbols to estimate the channel. Since the processing of the received data was realized within the *while loops*, the reception of so long frames derived in overflow errors in our LV implementation. However, we solved that issue by reducing the IQ sampling rate to 2 MHz.

In Figure 5.12 we observed that the throughput from the precoding matrix that maximized the SNR increased approximately two-fold in comparison to the fixed precoding matrix. We determined the same improvement for the LOS and NLOS scenarios. On the one hand, for the NLOS scenarios, the SNRmax precoding matrix boosted the channel throughput to values around 23 Mbps. We deduced that the channel estimation with the Tx and Rx nodes in different rooms with a door and wall in between offered similar channel estimation matrices. We inferred also that the conditions of the channels did not vary enough to appreciate changes in the throughput. That was a desirable scenario where unmovable objects define the main characteristics of the channel. Consequently, the selection of a precoding matrix according to this channel estimation boosted the network capacity.

On other hand, the LOS scenarios offered higher capacity but its channel conditions varied between realizations. We identify this behavior from our LOS-1 and LOS-2 realizations, where the SNRmax precoding matrix throughput had a difference of 2 Mbps. We speculate that the movement of persons and the position of objects inside the room influenced the channel characteristics.

To summarize, we verified that the use of precoding techniques such as codebook based precoding matrices, exploited the diversity of the channel by boosting its capacity.

Chapter 7

Conclusion

The predicted growth of the data traffic for the next years challenges the engineers to design and implement a new mobile broadband generation able to cope with this demand. Not only high peak data rates are addressed in the proposed designs, but also other key elements that are essential for the successful development of the novel transmission techniques that will be part of the coming next generation, the 5G. In order to develop these techniques, the simulations have to be validated within real-world scenario testbeds.

On the one hand, the software defined radio has proved to be capable of simulating a digital communication system saving hardware costs, development time and computational complexity. Moreover, the software implementation simplicity has contributed to the design of sophisticated techniques that are currently used in our mobile devices. On the other hand, we examined the benefits of transmission techniques that enhanced the system performance over fading channels in a real-world scenario using the USRPs within our SDR. We conclude that most of project objectives have been fulfilled as we examine in the following paragraphs.

As we designed the SISO communication system, we realized that the implementation of each block had to be tested as an independent unit and then carefully integrated into the general diagram. As the experience demonstrated us, the generation of boxes or modules from our designed schemes made the diagram more attractive to the sight. However, while simulating or integrating new features to our system, these modules became black boxes, namely the signal processes realized inside them became untraceable. As a result, we generated errors that stopped the simulation or were added to the signal and passed to the next module. Hence, the final performance results were incoherent and the probability to find the error within all the wired modules decreased in proportion to the number of implemented modules. We solved this time expender error hunting by designing a flat hierarchic design in order to track our information data through all the modules from the transmitter over the channel to the receiver.

We agree that LabVIEW is a powerful tool, capable of simulating many communication modules and also offers the tools to create new ones. However, we also recognize that for a beginner user LabVIEW suppose a really complex tool were all is programed by modules connected with wires, which basically are implemented by mouse clicks. Consequently, any previous programming skills based on code lines, variable definition and object oriented rules are almost useless. For this reason among others, LabVIEW offers training courses with

different levels in order to master the basics and progressively acquire a professional LV software background, being able to understand each module and design more efficiently.

With these argumentations, we want to clarify that the majority of the time invested in the project was dedicated to the implementation of the SISO system in LV. As a consequence, we perceived that it was of great importance to design a transmission and reception system that did not introduce distortions and errors in the signal. In this way, we justified that our testbed results were based only on the wireless channel characteristics and not in black box modules which imperceptibly distorted the signal. Furthermore, the verification and validation was carried out within weeks since it only required the computation of the BER vs. SNR and its comparison with the theoretical expectations.

Per contra, the slow development design offered us the opportunity to understand each communication module purpose in detail. In addition to that, we broadened our knowledge by emulating most of the digital communication blocks that are found in the transmitter base stations and in the receiver user equipments. Apart from the basics features, we implemented some of the techniques that are present in the mobile communications. For example, CAZAC sequences for the pilot symbols, source coding interleaver, M-QAM modulation (for $M=4, 16, 64$ and 256) and maximal ratio combining (MRC) receivers. Moreover, other modules such as the convolutional encoder were implemented but not used for our simulations.

As we validated our SISO design, we enhanced our system to the SIMO configuration, which increased our receiver design complexity. Then, we managed to simulate different receive antennas acquiring results that matched the theoretical BER vs. SNR graphs. As a result, we obtained a first insight into the benefits of diversity techniques, in this case space diversity at the receiver side.

Before our transition to the MIMO system, we decided to adapt our current LV design to the USRPs, an important part of our project objectives in order to develop a real-world scenario testbed. First, we verified with an external spectrum analyzer that at the transmitter node our single antenna was sending to the wireless channel OFDM frames with the respective subcarrier spacing. Afterward, we implemented the receiver in order to simultaneously analyze the spectrum and capture the signal transmitted over the defined carrier frequency. As a result, we established a SISO connection between two USRP devices connected to independent computers.

For the 2x2 MIMO design, we started by using two devices for transmission and reception. Then, we ran our LV SDR and we observed that the transmitted signal with interleaved pilots fluctuated. Our preliminary results revealed that we received the subcarriers with the pilots pattern P1 or P2, alternatively. After spending time finding the error in the LV implementation, we realized that the USRP devices were not synchronized, such that every device used its own internal clock to transmit and the same for the USRP receivers. With that observations we studied the USRP configuration in more detail discovering that in order to transmit in MIMO mode, there was a configuration parameter that set the reference clock of the devices. Finally, we defined the clocks for MIMO transmission and reception respectively, obtaining the results presented that validated our USRP 2x2 MIMO system.

Having the MIMO real-world scenario testbed, we accomplish to transmit a long array of OFDM interleaved pilots symbols and at the receiver extract them in order to estimate the

channel for each of the symbols and its subcarriers. Consequently, we obtained the channel matrix for different scenario testbeds, particularly two indoor office nodes with line-of-sight (LOS) and non-line-of-sight (NLOS).

Our last goal was to verify that implementing precoding techniques that exploited the MIMO diversity we increased the channel capacity and thus boosted the network throughput. Hence, we collected the estimated channel matrix and stored it into a file. Then, by means of MATLAB scripts, first we generated the precoding matrices extracted from the codebook based on LTE specifications. Second, we set a fixed matrix and we implemented also an algorithm metric-based which selected the precoding matrix that maximized the SNR. Third, we extract our channel matrix from the file and applied the precoding method. Finally, we represented the CDF of the calculated throughput for our different scenario testbed realizations. The results demonstrated that; 1) precoding techniques maximized the channel capacity in scenarios with space diversity, 2) the SNRmax based algorithm to select the precoding matrix offered the highest throughput, and 3) it is possible to boost the network performance having some information about the channel characteristics.

To summarize, we implemented from scratch a SDR in LabVIEW and adapted it to transmit over the wireless channel using the USRP radio hardwares. Moreover, we designed an OFDM system and utilized interleaved pilot symbols to estimate the channel. Then, we enhanced our SDR from the SISO to the MIMO case with its respective 2x2 MIMO USRP configuration. Finally, we exploited the space diversity of our channel in a real world scenario testbed by using precoding methods such as codebook based precoding matrices, and consequently we theoretically validated its potential in boosting the network throughput.

Bibliography

- [1] P. Mogensen, K. Pajukoski, B. Raaf, E. Tiirola, L. Eva, I. Kovacs, G. Berardinelli, L. G. U. Garcia, L. Hu, and A. F. Cattoni, “B4g local area: high level requirements and system design,” *IEEE Globecom 2012*, 2012.
- [2] B. Raaf, W. Zirwas, K.-J. Friederichs, E. Tiirola, M. Laitila, P. Marsch, and R. Wichman, “Vision for beyond 4g broadband radio systems,” in *Personal Indoor and Mobile Radio Communications (PIMRC), 2011 IEEE 22nd International Symposium on*, pp. 2369–2373, IEEE, 2011.
- [3] O. Tonelli, G. Berardinelli, F. M. Tavares, A. F. Cattoni, I. Z. Kovacs, T. B. Sorensen, P. Popovski, and P. E. Mogensen, “Experimental validation of a distributed algorithm for dynamic spectrum access in local area networks,” in *Vehicular Technology Conference (VTC Spring), 2013 IEEE 77th*, pp. 1–5, IEEE, 2013.
- [4] C. V. Forecast, “Cisco visual networking index: Global mobile data traffic forecast update 2013-2018,” *Cisco Public Information, February*, vol. 1, 2014.
- [5] P. Mogensen, K. Pajukoski, E. Tiirola, J. Vihriälä, E. Lähetkangas, G. Berardinelli, F. M. L. Tavares, N. H. Mahmood, M. Lauridsen, D. Catania, *et al.*, “Centimeter-wave concept for 5g ultra-dense small cells,”
- [6] P. Mogensen, K. Pajukoski, E. Tiirola, E. Lähetkangas, J. Vihriälä, S. Vesterinen, M. Laitila, G. Berardinelli, G. W. Da Costa, L. G. Garcia, *et al.*, “5g small cell optimized radio design,” in *International workshop on Emerging Technologies for LTE-Advanced and Beyond G*, vol. 4, 2013.
- [7] G. W. Costa, A. F. Cattoni, I. Z. Kovacs, and P. E. Mogensen, “A fully distributed method for dynamic spectrum sharing in femtocells,” in *Wireless Communications and Networking Conference Workshops (WCNCW), 2012 IEEE*, pp. 87–92, IEEE, 2012.
- [8] F. M. Tavares, G. Berardinelli, N. H. Mahmood, T. B. Sorensen, and P. Mogensen, “On the potential of interference rejection combining in b4g networks,” in *Vehicular Technology Conference (VTC Fall), 2013 IEEE 78th*, pp. 1–5, IEEE, 2013.
- [9] O. Tonelli, I. Rodriguez, G. Berardinelli, A. F. Cattoni, J. L. Buthler, T. B. Sørensen, and P. Mogensen, “Validation of an inter-cell interference coordination solution in real-world deployment conditions,”
- [10] K. Wesolowski, *Introduction to digital communication systems*. John Wiley & Sons, 2009.
- [11] J. Proakis and M. Salehi, *Digital Communications*. McGraw-Hill International Edition, McGraw-Hill, 2008.

- [12] S. Benedetto, *Principles of digital transmission: with wireless applications*. Springer, 1999.
- [13] L. Andrews, *Special Functions of Mathematics for Engineers*. Oxford science publications, SPIE Optical Engineering Press, 1992.
- [14] M. Pätzold, M. Patzold, and M. Paetzold, *Mobile fading channels*. John Wiley England, 2002.
- [15] M. K. Simon and M.-S. Alouini, *Digital communication over fading channels*, vol. 95. John Wiley & Sons, 2005.
- [16] A. R. Bahai, B. R. Saltzberg, and M. Ergen, *Multi-carrier digital communications: theory and applications of OFDM*. Springer, 2004.
- [17] L.T.E, “Evolved universal terrestrial radio access network (e-utran): Overall description; stage 2, 3gpp std,” *TS*, vol. 36, p. V8, 2008.
- [18] F. Khan, *LTE for 4G mobile broadband: air interface technologies and performance*. Cambridge University Press, 2009.
- [19] D. Chu, “Polyphase codes with good periodic correlation properties (corresp.),” *Information Theory, IEEE Transactions on*, vol. 18, pp. 531–532, Jul 1972.
- [20] National Instruments, *Manual LabVIEW 2014 Help*, labview 2014 ed., 2014. <http://zone.ni.com/reference/en-XX/help/371361L-01/>.
- [21] U. T. R. A. N. (UTRAN), “Physical layer aspects for evolved universal terrestrial radio access (release 8),” tr 25.814 v7.1.0, European Telecommunications Standards Institute (ETSI), September 2006.
- [22] J. Erskine-Murray, *A handbook of wireless telegraphy*. Van Nostrand, 1909.
- [23] R. Simons, “Guglielmo marconi and early systems of wireless communication,” *GEC Review*, vol. 11, no. 1, pp. 37–55, 1996.
- [24] E. Stone, *Elements of Radiotelegraphy*. D. Van Nostrand Company, 1919.
- [25] Qualcomm, “The evolution of mobile technologies: 1g,2g,3g,4g lte,” tech. rep., 2014.
- [26] C. Shannon and W. Weaver, *The Mathematical Theory of Communication*. No. pt. 11 in Illini books, University of Illinois Press, 1963.
- [27] C. Clark, *LabVIEW Digital Signal Processing and Digital Communications*. McGraw-Hill Education, 2005.

Appendix A

120 Years of Radio Communication Engineering

Although it could surprise to some readers, the first radio communication was realized with a wireless telegraph [22] and it can be seen as a digital communication system where ‘0’ and ‘1’ were sent using the Morse code. Afterwards, it came the wireless telephony (analog signal) which monopolized the communications. Fifty years later the digital era started until nowadays that all is digitalized even the telephony.

Hence, it is worth to mention that the era of the wireless communications began with the simplest digital communication system. However, not until 120 years later we state that we live in the digital era. In this first radio transmission, all the digital processing was done by humans who encoded/decoded the message to dots and lines which per analogy we associate to the ‘1’s of our digital systems. The truly discovering at that moment was the transmission and reception of an analog signal through a non-wired medium, the air. But before going on into more technical details, let us start from the beginning.

The Italian inventor Guglielmo Marconi [23] in 1896 patented the first commercially wireless telegraph system [24] based on airborne Hertzian waves, more known as electromagnetic waves. From that point, the radio communications revolution started and nowadays he is credited as the inventor of radio despite the early contributions of many other inventors ¹. We illustrate in the Figure A.1 the receiver circuit of Marconi’s invention.

¹Julio Cervera Baviera, a Spanish engineer, was pioneer in development of radio. He collaborated with Marconi on resolving the problems of the wireless communication system. Baviera developed his own system and patented the radio in 1902 in England, Germany, Belgium, and Spain. On the other hand, Marconi demonstrated the effectiveness of the wireless telegraph at the end of 1901, but he did not start the production of radios until 1913. Therefore, some researchers postulate that the inventor of the radio was not Marconi, but the Spaniard Julio Cervera Baviera.

In a more practical point of view, if we had to reproduce all the events to build up the first successful wireless transmission we would need to follow the next steps. Firstly, we would have to find the Danish physicist and chemist Hans Ørsted who discovered that electric currents create magnetic fields. Then, the Scottish mathematician James Clerk Maxwell could hypothesized that a changing electric field induces a magnetic field.

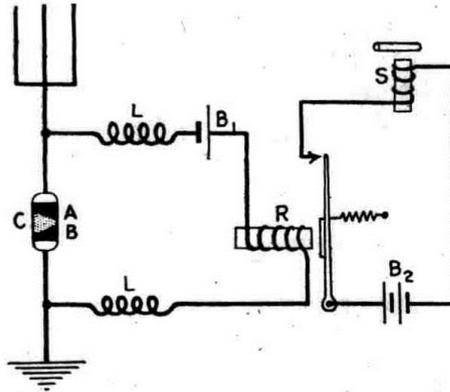


Figure A.1: Marconi 1896 Receiver [24].

At the same time, Michael Faraday would have found that a changing magnetic field induces an electric field. With all together, Maxwell could then produce a pair of equations whose results forecasted electromagnetic waves propagating at the light's speed. Years later, the German physicist Heinrich Hertz would prove with success the Maxwell's equations for electromagnetism. However, because of the fact that at low frequencies waves could not propagate efficiently, Hertz assumed that radio would not be a reliable instrument for telegraph signals. Nevertheless, we all know that a high-frequency radiowave is used as a carrier of low-frequency information signals, named up-conversion technique. Therefore, thanks to all the previous experimental work of Hans Christian Ørsted, Michael Faraday, James Clerk Maxwell and Julio Cervera Baviera, among others, Marconi could exploit the possibilities of radio and succeed.

At this point, in an elemental way we are able to generate electromagnetic waves, transmit and receive them. Then, with the properly method (modulation) to modify these waves according to our source of information (e.g. audio), we send an analog signal and receive it wireless. At the receiver, we reverse these steps (demodulation) and we extract our useful information (e.g. audio). As a result, we have our elemental radio communication system. We depict it in the Figure A.2 with a block diagram.

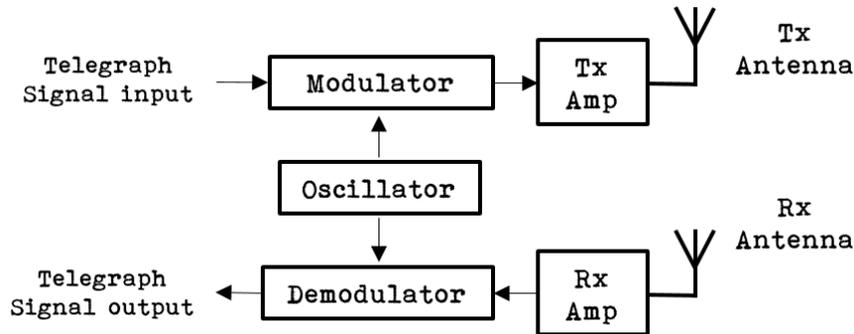


Figure A.2: Block Diagram of a Wireless Telegraph System.

In this case, the signal is generated with a telegraph, the oscillator generates the carrier frequency and the transmission (Tx) and receiver (Rx) amplifier modules supply more power to the signal for a successful propagation through the air.

From that rudimentary structure, we have improved the techniques and developed new ones in order to supply a fast, reliable, secure and efficient radio communication system. Even now, the modern and cutting-edge communication systems are based on these same principles mentioned above. Indeed, they process the information in the digital domain while the modulation, which changes the wave characteristics according to our information, and the up-conversion, which uses a high-frequency signal that carries the low-frequency information signal, are supported only in the analog ² domain.

As the revolution of radio started within the 20th century, the communications between distant stations began to emerge. One of the major advancements on radio systems occurred when German engineers erected radio towers between New York and Germany, establishing the only wireless communication from North America to Europe.

It was the Canadian inventor Reginald Aubrey Fessenden who used continuous waves to send the first radio transmissions of voice and music. His invention of the amplitude modulated (AM) radio allowed more stations to send signals without covering the entire bandwidth spectrum. Apart from that, thanks to his invention of the heterodyning technique, it was possible to create new frequencies by combining or mixing two others, process that is needed for the merging of the low-frequency signal (information) with the high-frequency signal (carrier).

The next three important radio features invented in the following years by the American electrical engineer Edwin Howard Armstrong were the regeneration circuit, the superheterodyne circuit and the frequency modulation (FM). The regeneration circuit, now called positive feedback, allows to amplify an electronic signal many times through a feedback loop. That was of important relevance for the receiver design which could improve the received

² We understand by analog or analogue signal, any continuous signal which varies in time and its features are related to physical phenomena such as sound, temperature, light, pressure or position. At the contrary, a digital signal is represented by a non-continuous signal with finite values and it is also used to represent analog signals. For instance, this conversion is used in the analog-to-digital (A/D) module in our digital communication systems.

signal in order to obtain its original characteristics (e.g. transition from headphones to speakers).

The superheterodyne added an extra stage to the original radio carrier frequency mixer. The convenience to use an additional intermediate frequency (IF) was the improvement on selectivity and sensitivity, the most relevant characteristics on ideal receivers. The third invention was the frequency modulator. As the name suggests, the information is encoded in a carrier wave by varying the frequency of the wave while the amplitude remains constant. Contrarily, the amplitude modulator (AM) keeps the frequency constant and varies the signal's amplitude.

We illustrate in the Figure A.3 an example signal carried using these modulation techniques.

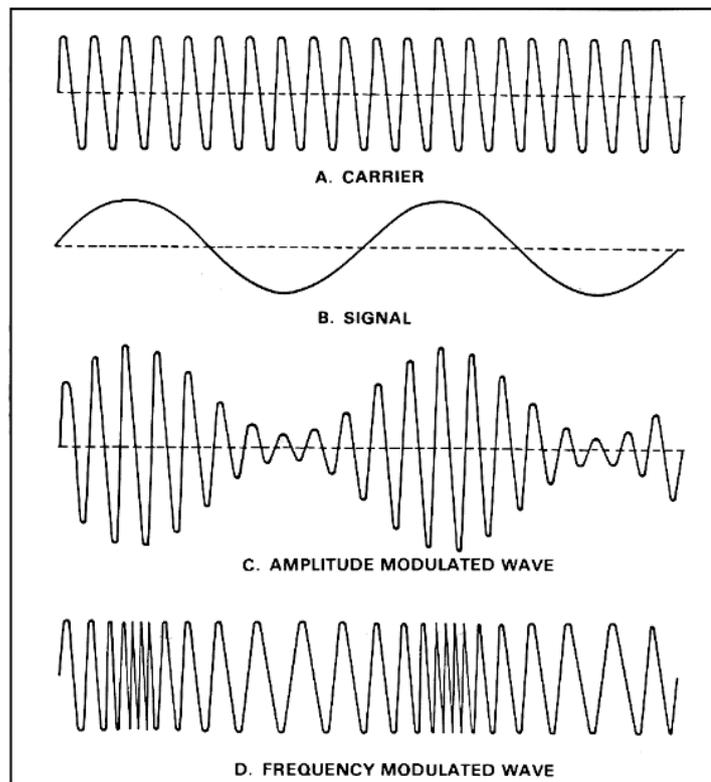


Figure A.3: Wave Shapes.

Competing with Marconi, the American inventor Lee de Forest invented the Audion, also called de Forest valve and nowadays as triode. The triode was one of the milestones for the development of transcontinental telephone communications, radio, and radar until 1948 with the invention of the transistor.

As we mentioned before, the equipment, techniques and applications evolved during the years but the principle of radio communications remained the same. It was the Mobile

Telephone Service (MTS) the first pre-cellular radio system and one of the earliest mobile telephone standards. It granted to the users their own telephone number and the calls were routed to a mobile operator. Years later, the A-Netz (A-Net) in West Germany offered the first public commercial mobile phone network. The automatic direct-dial feature was developed with the Improved Mobile Telephone Service (IMTS).

The Altai mobile telephone system turned into the first automatic mobile phone system in Europe. It is considered the 0-G generation of the radiotelephone service and from it all the successive generations evolved until the fifth generation (5G), which would be introduced around 2020. The first generation (1G) of mobile telecommunications was based on analog telecommunications standards and it was replaced by the 2G generation with the only difference that the radio signals used by the 2G networks were digital. That means that the phone conversations were digitally encrypted and only the intended receiver could read it.

Let us go back to the 1G which established the foundations of mobile broadband. Base stations (BS) were deployed by the operator to grant access for subscribers. However, the spectrum is a limited resource and the analog voice services required a complete channel for one phone call. Therefore, the network had limited capacity and the interference between proximity channels was the main problem regarding efficiency. Moreover, analog devices are heavier, power inefficient and expensive. Hence, 2G provided the digital technology that delivered voice services to more people in more places due to its improved capacity. Furthermore, the spectrum efficiency was improved while the channels could support more users simultaneously. 2G also introduced data services such as text messages (SMS). In many parts of the world 2G networks are still being used.

The improved capacity was achieved compressing the voice signal into small packets. That processing technique is done in the encoder block where the information (signal) is encoded in such a way that less bits are needed to transmit. That enabled more users per radio channel using the same frequency but at different times, what is called time division multiple access (TDMA). Another benefit of the second generation is that digital devices components were cheaper and offered a more secure signal, which contributed to expand the mobile market with new operators and subscribers.

Apart from the TDMA, the code division multiple access (CDMA) was developed and different users shared the same frequency at the same time while they were using different unique languages ("codes"). At the receiver, the signals with different codes as the expected were discarded and the one related to the desired communication link (same code for Tx and Rx) was reconstructed.

While the 2G technology improved the capacity three times compared to 1G, the CDMA technique provided a capacity up to 14-fold more [25]. Hence, CDMA was the foundation for the mobile 3G technologies which introduced for the first time high-speed internet access.

The third generation boosted the data services and subscribers wanted to use the same capabilities of the broadband Internet at home in their phones, causing the raise of the smart-phones. Some of the advancements on data communications were the increase of the peak data rate using high order modulations and aggregating spectrum (carrier aggregation), and the introduction of a data optimized channel (data only) CDMA2000/EV-DO, which later would be embedded within the voice channel (WCDMA/HSPA).

Nowadays, the 4G Long Term Evolution (LTE) is being spread among more devices and essentially complements 3G boosting the data capacity. An enhancement of this standard named LTE Advanced is being commercially deployed at the end of the 2014s in several countries and it takes advantage of advanced topology networks with a mix of low power nodes (e.g. picocells, femtocells). Most of the novel technologies used in the 4th generation would be enhanced also in the 5th generation. For instance, some of these improved features are wider channels enabled with Orthogonal Frequency Division Multiple Access (OFDMA), advanced multiple-input-multiple-output (MIMO) techniques (more antennas) and carrier aggregations. The 4G LTE is the first global standard for mobile broadband which operates in two modes, frequency division duplex (FDD) and time division duplex (TDD).

In order to offer more data capacity, our targets are related to the Shannon's Law of Capacity [26]. As the equation in Figure A.4 shows, by improving the spectrum, increasing the number of antennas and reducing interference we will increase the capacity. Mainly, we aim to use all the spectrum more efficiently:

- Creating spatially separated data paths with advanced multiple antenna techniques (4x4 MIMO).
- Reducing the interference using advanced receivers and techniques.

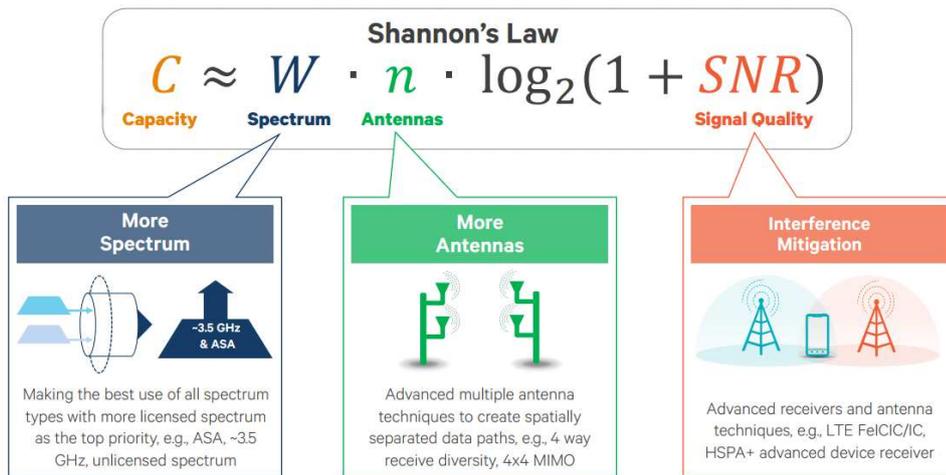


Figure A.4: Mobile 3G/4G Technologies evolving for more Data Capacity [25].

Every new mobile generation introduced a wider spectral bandwidth and a higher peak bitrate. The new generation of 5G standards may be available approximately around 2020s and apart from the key features mentioned above it would have to offer more simultaneous device connections, low battery consumption, better coverage, lower latencies, lower deployment costs and higher reliability, among others. As we recall from the objectives of this project, these are our key design elements for the fifth mobile generation and we aim to analyze the performance of our 5G techniques with the SDR testbed proposed in this thesis.

Appendix B

Software Defined Radio and USRP Radio Hardware

The implementation of the basic elements such as mixers, filters, modulators, amplifiers, detectors, etc. is expensive and takes time if its purpose is only for tests. Therefore, *software-defined radio* is a software or embedded system which by mean of computer processing replaces all the hardware elements of a radio communication system. Furthermore, as a software implementation its capabilities are beyond the practical approach. That means that theoretical processes and algorithms can be simulated and tested, while due to complexity its real implementation is not possible.

The Software Defined Radio (SDR) not only allows to prototype and test communication systems, but also to combine them with antennas in order to use spread spectrum techniques where several transmitters could use the same frequency with very little interference. That also provides cognitive radio techniques which measures the spectrum usage and communicates that information with other radios. Another advantage of SDRs are their libraries and embedded tools. Not all has to be designed from scratch since the main communications blocks are available to use and only its initial values and setup variables has to be defined. By this way, the prototyping time decreases and the system performance is characterized faster. Also the costs are reduced if the hardware components are co-simulated before its real implementation.

While MATLAB and C are also powerful tools for SDRs, LabVIEW do not require coding such as memory allocation or variable declarations, no compiling, and it has a powerful built-in display utilities for representing the signals at various points of our communication system. However, it is far to be a real-time environment because its system is too slow for real-time communications, but is suitable for instrument-type measurements [27]. Despite its limited speed, LabVIEW offers flexibility, easy coding and outstanding data display and user interface.

The *Universal Software Radio Peripheral* (USRP) is a hardware platform designed by National Instruments. It transmits and receives data based on modules that process base-band signals and serve as the RF front end. Then, using a SDR, for example *LabVIEW*, it is possible to design, implement and test algorithms and techniques for wireless communications.



USRP™ N200/N210 NETWORKED SERIES



FEATURES:

- Use with GNU Radio, LabVIEW™ and Simulink™
- Modular Architecture: DC-6 GHz
- Dual 100 MS/s, 14-bit ADC
- Dual 400 MS/s, 16-bit DAC
- DDC/DUC with 25 mHz Resolution
- Up to 50 MS/s Gigabit Ethernet Streaming
- Fully-Coherent MIMO Capability
- Gigabit Ethernet Interface to Host
- 2 Gbps Expansion Interface
- Spartan 3A-DSP 1800 FPGA (N200)
- Spartan 3A-DSP 3400 FPGA (N210)
- 1 MB High-Speed SRAM
- Auxiliary Analog and Digital I/O
- 2.5 ppm TCXO Frequency Reference
- 0.01 ppm w/ GPSDO Option

N200/N210 PRODUCT OVERVIEW:

The Ettus Research™ USRP™ N200 and N210 are the highest performing class of hardware of the USRP™ (Universal Software Radio Peripheral) family of products, which enables engineers to rapidly design and implement powerful, flexible software radio systems. The N200 and N210 hardware is ideally suited for applications requiring high RF performance and great bandwidth. Such applications include physical layer prototyping, dynamic spectrum access and cognitive radio, spectrum monitoring, record and playback, and even networked sensor deployment.

The Networked Series products offers MIMO capability with high bandwidth and dynamic range. The Gigabit Ethernet interface serves as the connection between the N200/N210 and the host computer. This enables the user to realize 50 MS/s of real-time bandwidth in the receive and transmit directions, simultaneously (full duplex).

The Networked Series MIMO connection is located on the front panel of each unit. Two Networked Series units may be connected to realize a complete 2x2 MIMO configuration using the optional MIMO cable. External PPS and reference inputs can also be used to create larger multi-channel systems. The N200 and N210 are largely the same, except that the N210 features a larger FPGA for customers that intend to integrate custom FPGA functionality.

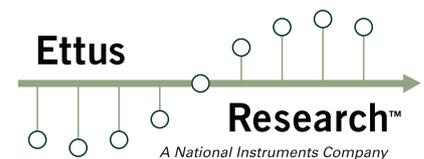
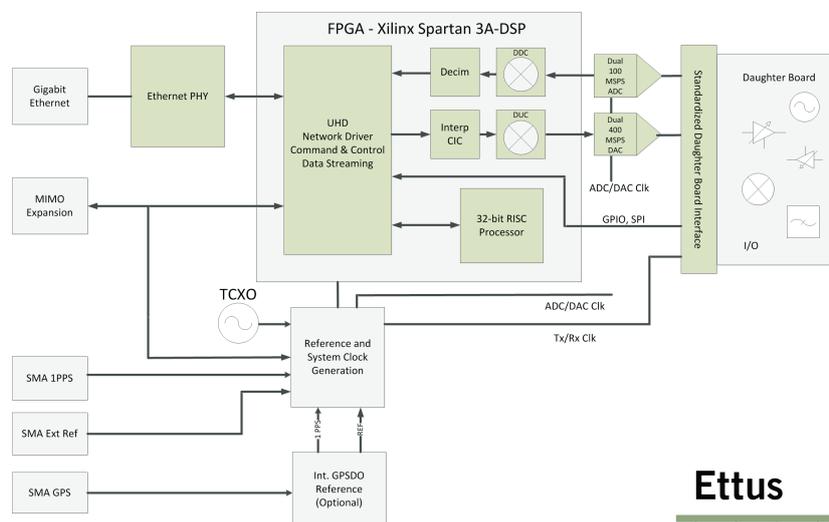
The USRP Hardware Driver™ is the official driver for all Ettus Research products. The USRP Hardware Driver supports Linux, Mac OSX, Windows.

USRP™ N200/N210 NETWORKED SERIES

SPECIFICATIONS

Spec	Typ.	Unit	Spec	Typ.	Unit
POWER			RF PERFORMANCE (W/ WBX)		
DC Input	6	V	SSB/LO Suppression	35/50	dBc
Current Consumption	1.3	A	Phase Noise (1.8 GHz)		
w/ WBX Daughterboard	2.3	A	10 kHz	-80	dBc/Hz
CONVERSION PERFORMANCE AND CLOCKS			100 kHz	-100	dBc/Hz
ADC Sample Rate	100	MS/s	1 MHz	-137	dBc/Hz
ADC Resolution	14	bits	Power Output	15	dBm
ADC Wideband SFDR	88	dBc	IIP3	0	dBm
DAC Sample Rate	400	MS/s	Receive Noise Figure	5	dB
DAC Resolution	16	bits	PHYSICAL		
DAC Wideband SFDR	80	dBc	Operating Temperature	0 to 55°	C
Host Sample Rate (8b/16b)	50/25	MS/s	Dimensions (l x w x h)	22x16x5	cm
Frequency Accuracy	2.5	ppm	Weight	1.2	kg
w/ GPSDO Reference	0.01	ppm			

* All specifications are subject to change without notice.



ABOUT ETTUS RESEARCH:

Ettus Research is an innovative provider of software defined radio hardware, including the original Universal Software Radio Peripheral (USRP) family of products. Ettus Research products maintain support from a variety of software frameworks, including GNU Radio. Ettus Research is a leader in the GNU Radio open-source community, and enables users worldwide to address a wide range of research, industry and defense applications. The company was founded in 2004 and is based in Mountain View, California. As of 2010, Ettus Research is a wholly owned subsidiary of National Instruments.

1043 North Shoreline Blvd
Suite 100
Mountain View, CA 94043

P 650.967.2870 www.ettus.com
F 866.807.9801

Appendix C

LabVIEW Additional Implementations

C.1 SISO System

C.1.1 Global Parameters

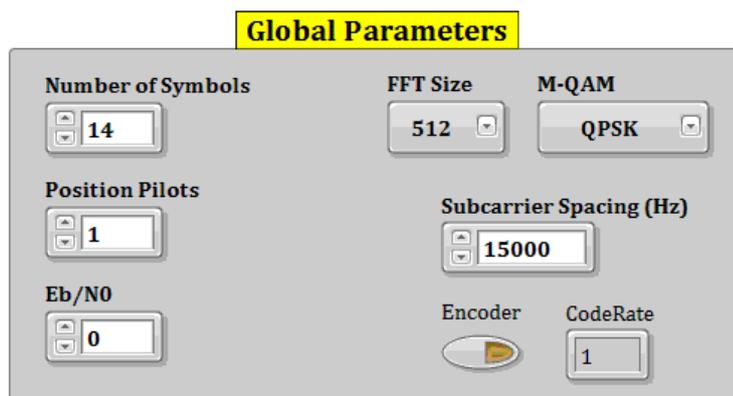


Figure C.1: Graphical User Interface (GUI) for the main global parameters selection.

C.1.2 Transmitter

Random Bits Generator

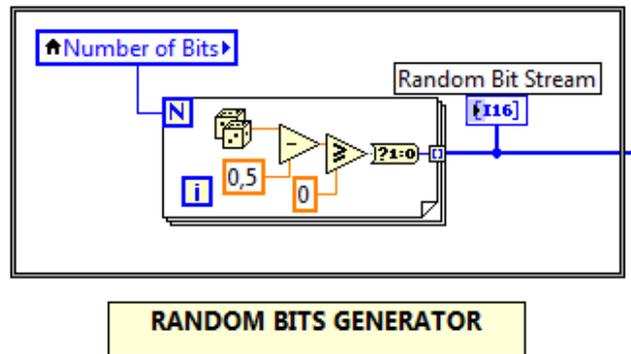
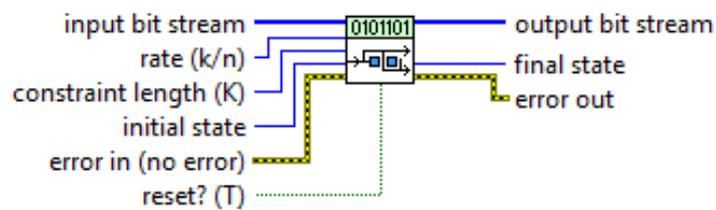


Figure C.2: Random Bits Generator in LabVIEW.

Coding/Encoder

niDMT.lvlib:MT Convolutional Encode.vi

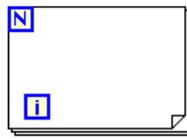


Generates an encoded bit stream based on a specified code rate. The code rate is equal to the ratio of the data word length to the code word length. This VI allows you to choose a code rate of 1/2, 1/3, 1/4, 2/3, or 3/4 with preconfigured generator matrices. Use the Generator Matrix instance to specify custom values.

Figure C.3: Convolutional Encoder LabVIEW VI [20].

DATA Split

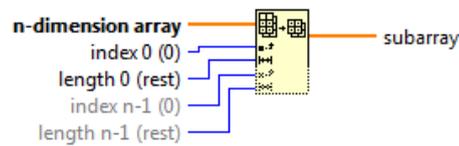
For Loop



Executes its subdiagram n times, where n is the value wired to the count (**N**) terminal. The iteration (**i**) terminal provides the current loop iteration count, which ranges from 0 to $n-1$.

Figure C.4: FOR loop LabVIEW [20].

Array Subset

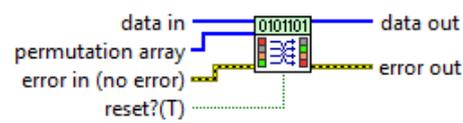


Returns a portion of **array** starting at **index** and containing **length** elements.

Figure C.5: Array Subset LabVIEW VI [20].

Channel Coding/Encoder

niDMT.lvlib:MT Permute.vi



Changes the position of input data elements based on values specified in the **permutation array** parameter.

Figure C.6: Channel Coding Interleaver LabVIEW VI [20].

Pulse Shaping

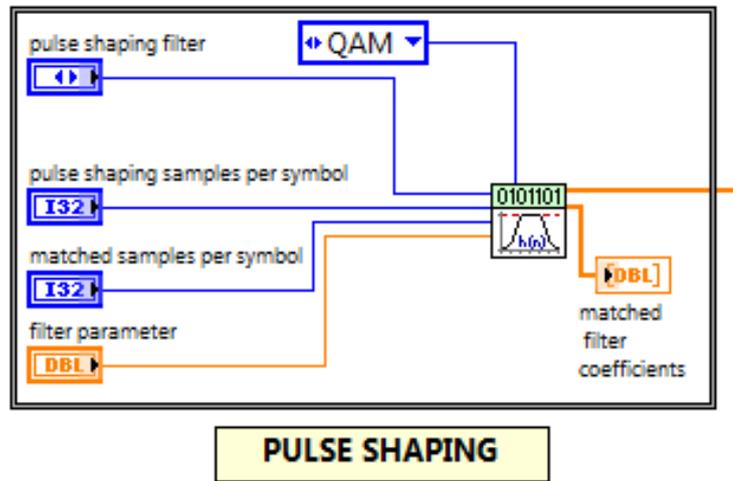
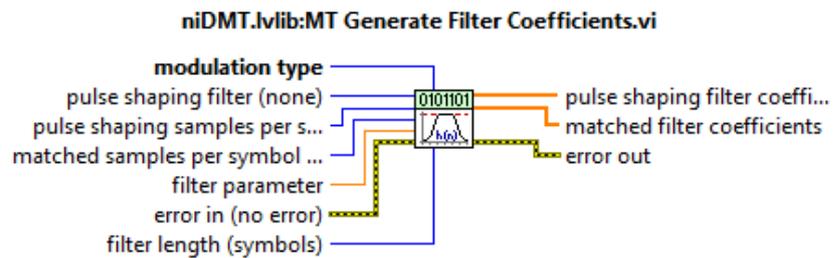


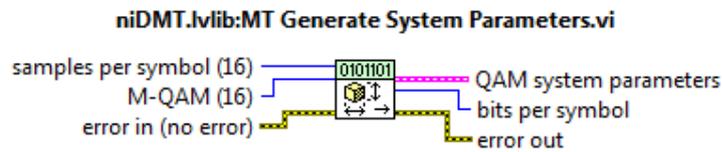
Figure C.7: Pulse Shaping in LabVIEW.



Calculates filter coefficients for pulse-shaping and matched filters applied by the digital modulation VIs and demodulation VIs.

Figure C.8: Generate Filter Coefficients LabVIEW VI [20].

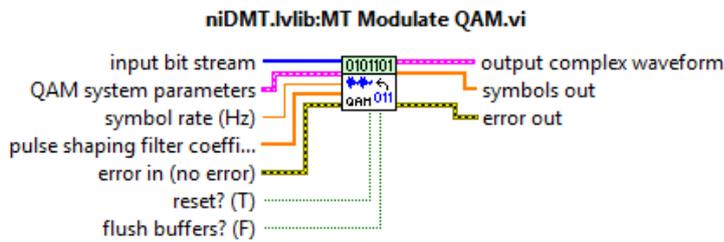
QAM System Parameters



Calculates parameters for use with modulation and demodulation VIs. Wire the **system parameters** cluster from this VI to the corresponding parameter of the appropriate demodulation or modulation VI.

Figure C.9: Generate QAM System parameters LabVIEW VI [20].

QAM Modulator and Pilots Insertion



Receives a sequence of data bits, performs QAM modulation, and returns the modulated complex baseband waveform in the **output complex waveform** parameter.

Figure C.10: Modulate QAM LabVIEW VI [20].

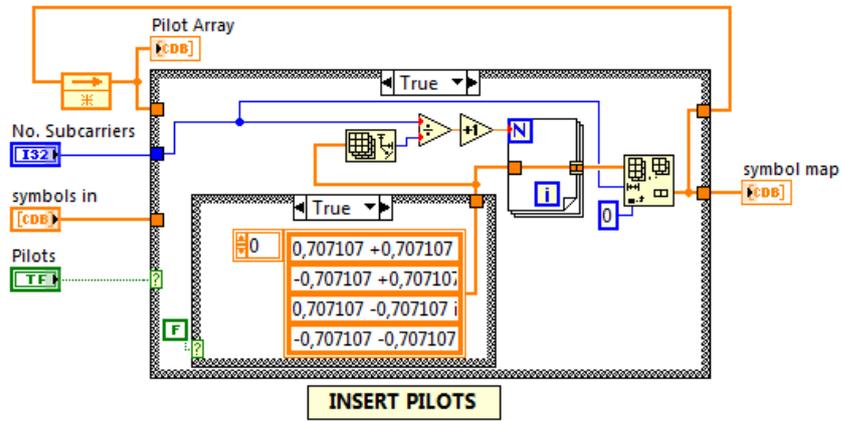


Figure C.11: Design of the Pilots Insertion VI for the 'True' case.

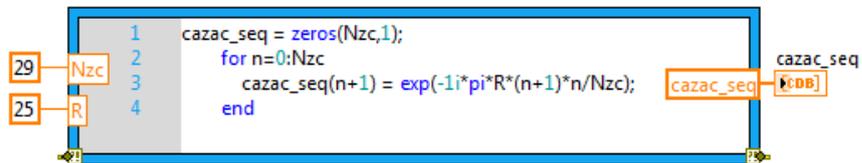


Figure C.12: CAZAC sequence generation script of the Pilots Insertion VI for the 'True' case.

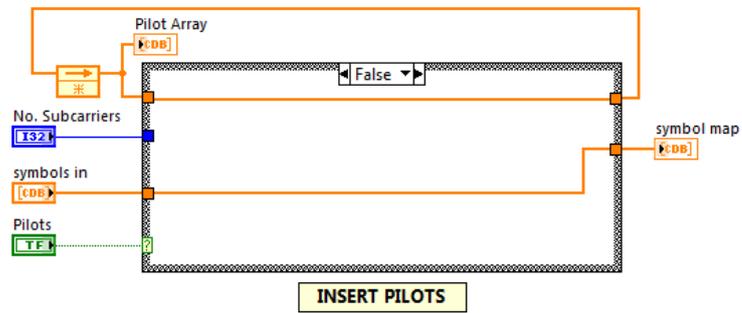


Figure C.13: Design of the Pilots Insertion VI for the 'False' case.

IFFT and Cyclic Prefix

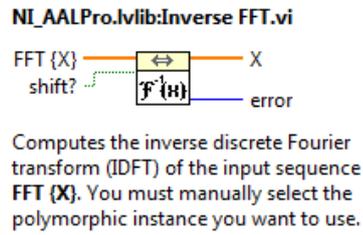


Figure C.14: Inverse FFT LabVIEW VI [20].

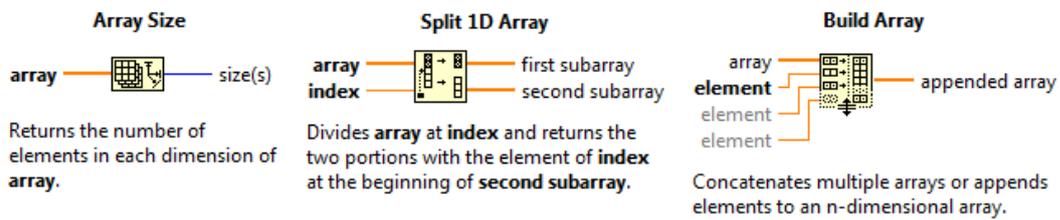


Figure C.15: Basic VIs for the Cyclic prefix generation [20].

Subcarrier Spacing

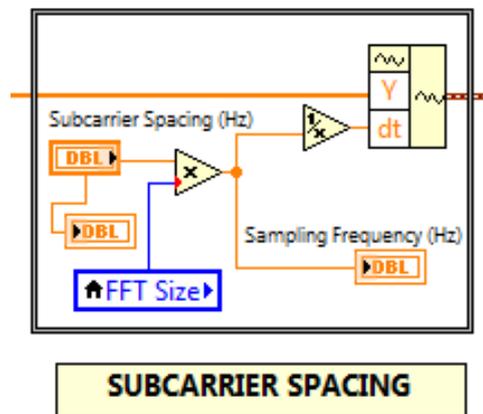


Figure C.16: Subcarrier spacing in LabVIEW.

C.1.3 Channel

Rayleigh Fading Taps

Table C.1: PDP of the used channel models

N _p	Typical Urban		Indoor	
	Delay(ns)	Power(dB)	Delay(ns)	Power(dB)
1	0	-5.7	0	-2.2
2	217	-7.6	5	-6.6
3	512	-10.1	5	-2.1
4	514	-10.2	5	-5.8
5	517	-10.2	15	-3.3
6	674	-11.5	15	-4.7
7	882	-13.4	15	-4.1
8	1230	-16.3	20	-8.2
9	1287	-16.9	20	-3.0
10	1311	-17.1	25	-5.2
11	1349	-17.4	30	-7.0
12	1533	-19.0	35	-4.6
13	1535	-19.0	40	-6.8
14	1622	-19.8	45	-8.6
15	1818	-21.5	80	-10.0
16	1836	-21.6	85	-12.1
17	1884	-22.1	110	-12.4
18	1943	-22.6	115	-11.8
19	2048	-23.5	150	-20.4
20	2140	-24.3	175	-16.6

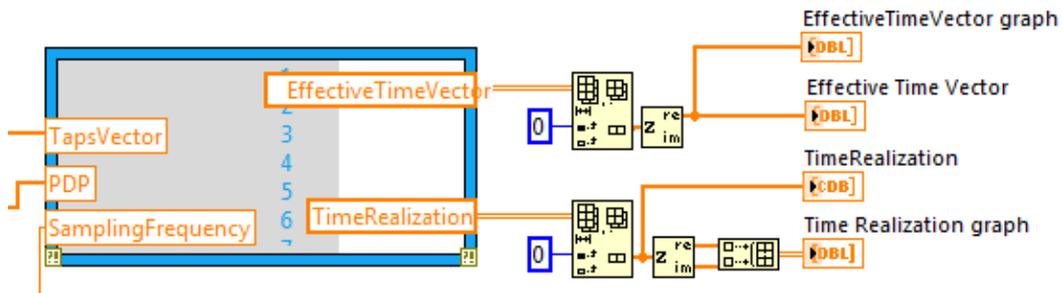


Figure C.17: Fading Taps module.

```

1 NumberTaps=length(TapsVector);
2 SamplingTime=1/SamplingFrequency;
3 DelayVec=round(TapsVector/SamplingTime);
4 EffectiveTimeVector=zeros(1,DelayVec(end)+1);
5 for count_path=1:NumberTaps
6     EffectiveTimeVector(1,DelayVec(count_path)+1)=EffectiveTimeVector(1,DelayVec(count_path)+1)+PDP(1,count_path);
7 end
8 EffectiveTimeVector=EffectiveTimeVector./sqrt(sum(abs(EffectiveTimeVector.^2)));
9 EffectiveTimeVectorReal=real(EffectiveTimeVector);
10 TimeRealization=(1/sqrt(2))*(EffectiveTimeVectorReal).* (randn(1,length(EffectiveTimeVectorReal))+j*randn(1,length(EffectiveTimeVectorReal)));
    
```

Figure C.18: Script for the generation of the time realization of the PDP taps.

AWGN

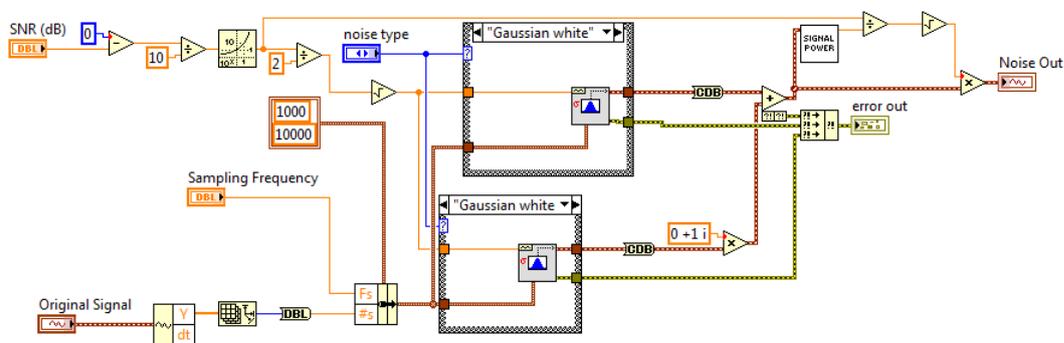


Figure C.19: NOISE generator block.

C.1.4 Receiver

OFDM Symbol Split

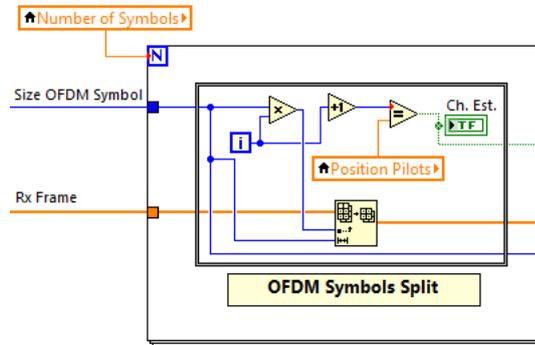
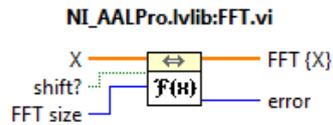


Figure C.20: OFDM symbol split block within the FOR loop.

Removal Cyclic Prefix and FFT



Computes the fast Fourier transform (FFT) of the input sequence X . Wire data to the X input to determine the polymorphic instance to use or manually select the instance.

Figure C.21: FFT LabVIEW VI [20].

Channel Decoding

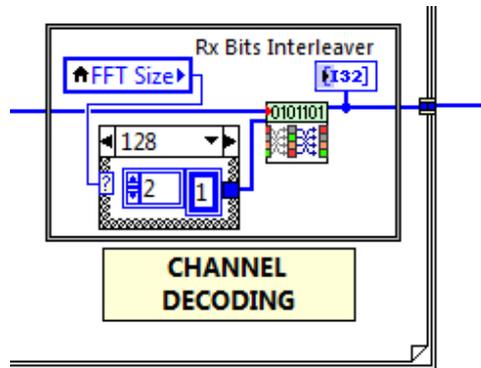


Figure C.25: Channel Decoding block in LabVIEW.

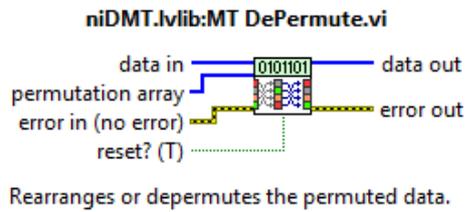


Figure C.26: Channel Decoder De-Interleaver LabVIEW VI [20].

Delete Pilots

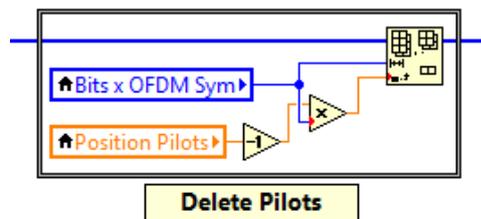


Figure C.27: Pilots Bits Deletion Block in LabVIEW.

Decoder

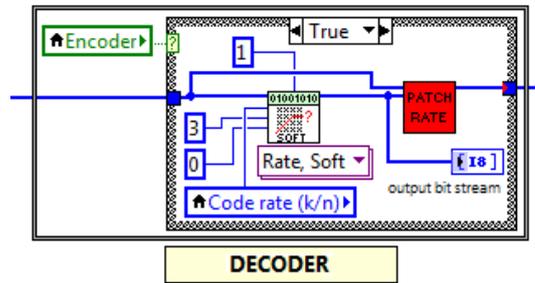
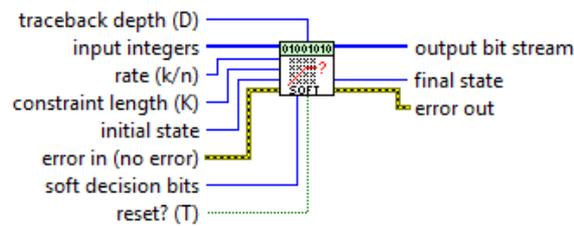


Figure C.28: Decoder Block in LabVIEW.

niDMT.lvlib:MT Convolutional Decoder (Viterbi Soft Decision, Rate).vi



This polymorphic instance decodes a convolutionally encoded bit stream, using a specified code rate and Viterbi soft decision decoding. The code rate is equal to the ratio of the data word length to the code word length.

Figure C.29: Convolutional Decoder LabVIEW VI [20].

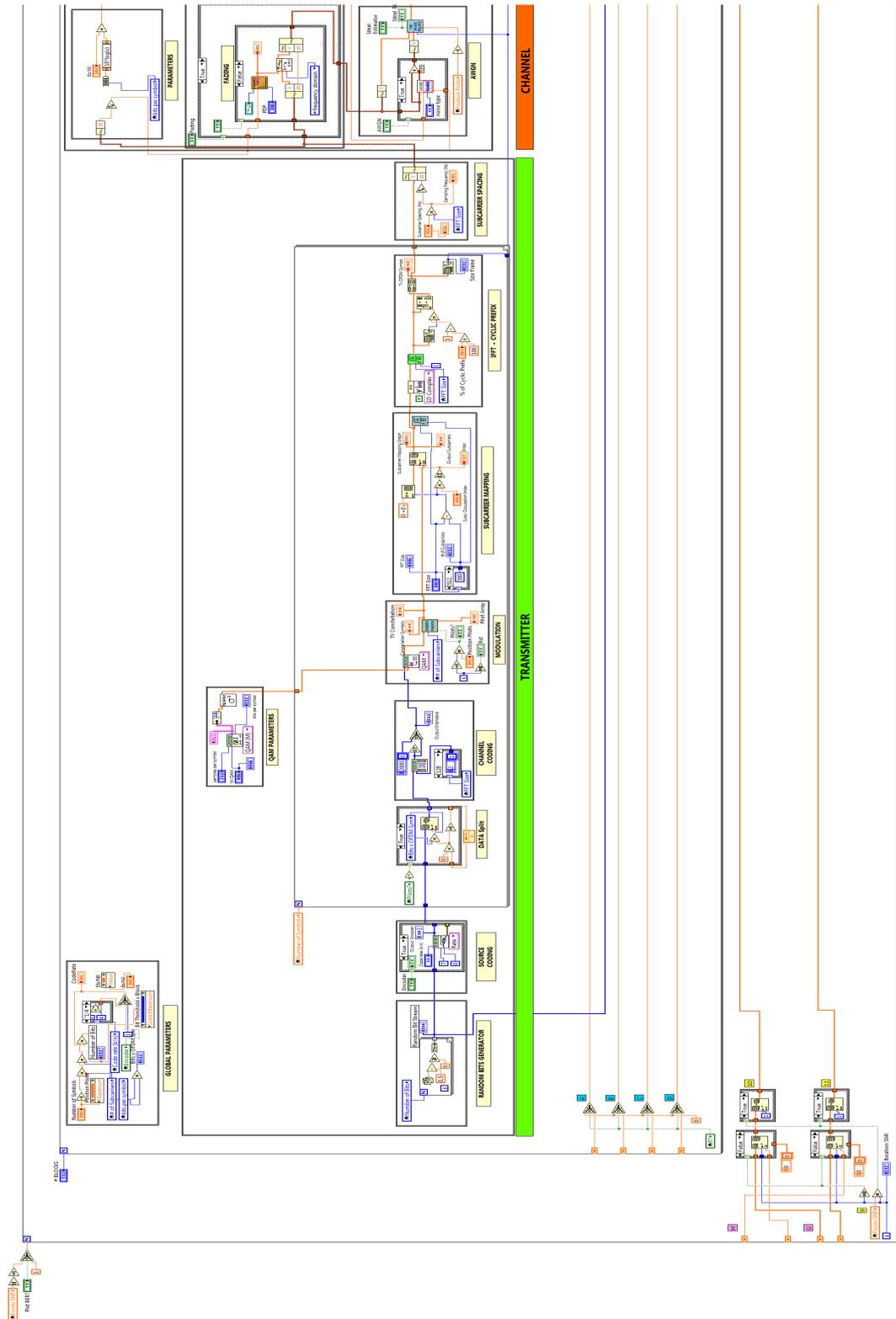


Figure C.30: (First part) Complete Digital Communication SISO system in LabVIEW.

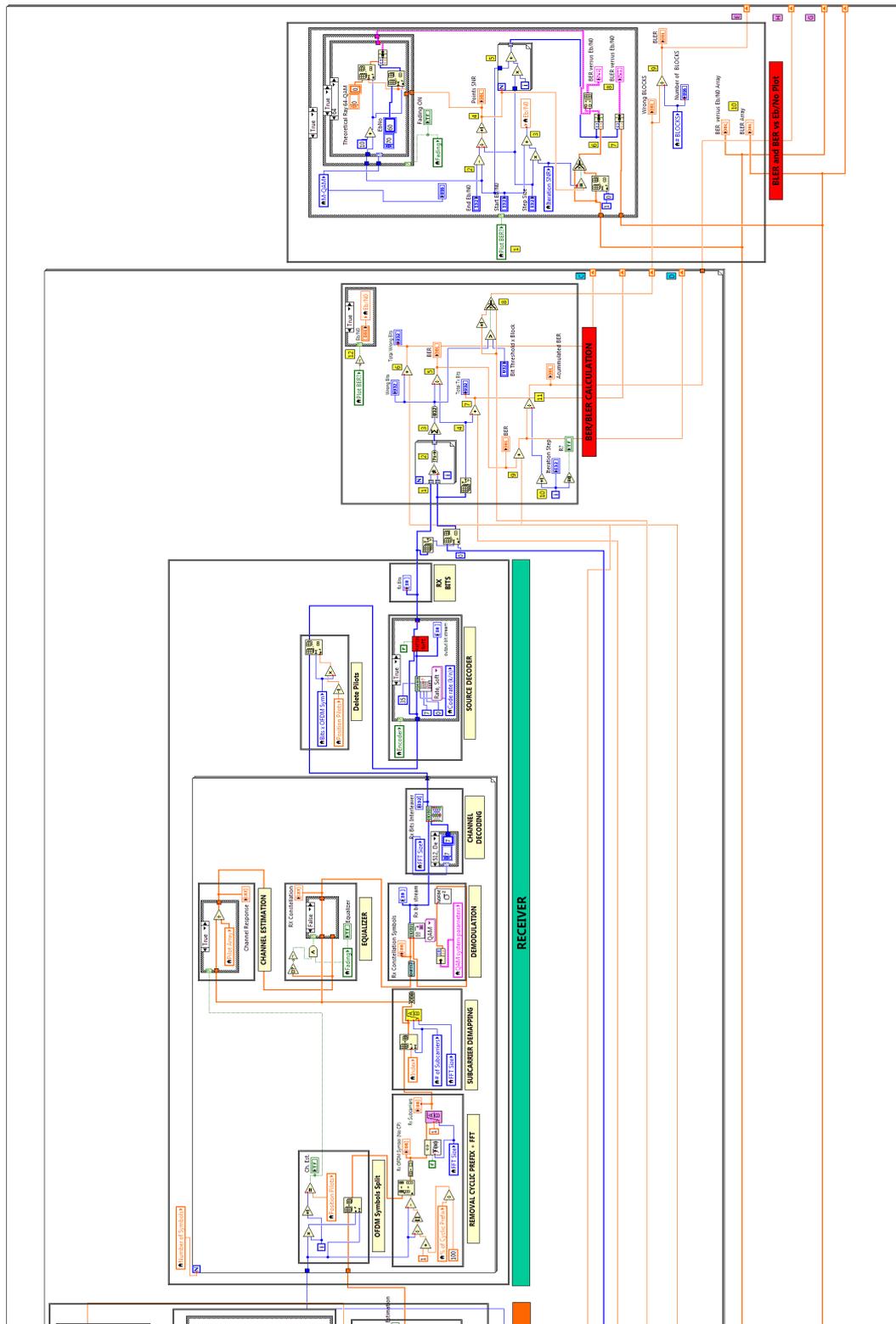


Figure C.31: (Second part) Complete Digital Communication SISO system in LabVIEW.

C.1.5 Graphical User Interface (GUI)

First of all, we selected the desired parameters of the digital communication at the global parameters GUI shown before in Figure C.1. Then, as we ran the simulation at least once, the rest of the parameters were automatically calculated. We represented some of these parameters in order to check that the variables were well defined. We illustrate in Figure C.32 the first three transmitter blocks GUIs. In our example, by default we sent 8 symbols within a frame, for a QPSK modulation and FFT 512 with 300 occupied subcarriers. As a result, 600 random bits are generated per symbol, and 7 data symbols gave us a total of 4200 bits transmitted per frame. The convolutional encoder was disabled in the figure as we see from the green led unlighted.

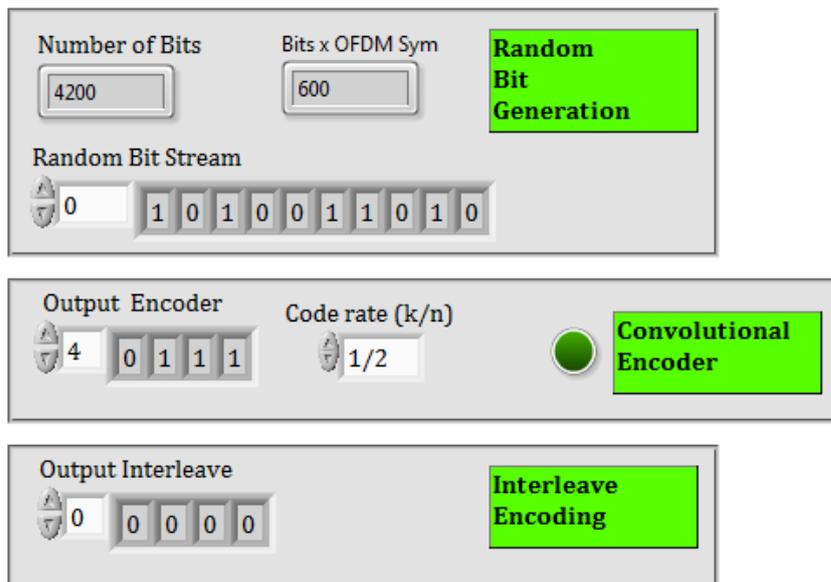


Figure C.32: Random bit generation, encoder and channel encoding block GUIs of the transmitter in LabVIEW.

In the following blocks we modulated the bits and mapped them into occupied subcarriers within a front and back band guard. We illustrate the GUIs of both stages in Figure C.33. Moreover, the pulse shaping block parameters were set but not used in our simulations.

In the subcarrier mapping we implemented a flexible design that allowed us to insert the occupied subcarriers in any position of the symbol of size FFT size. We controlled it by using the horizontal bar named *Subcarrier Occupation Index*, which '0' value represented the position [0] and '1' value the last position possible, namely [FFT size - number of occupied subcarriers]. We appreciate in figure that a factor of 0,5 establish an equal guard band at the front and back part of the subcarriers. Furthermore, the *Index* variable represents the position where the first occupied subcarrier was inserted.

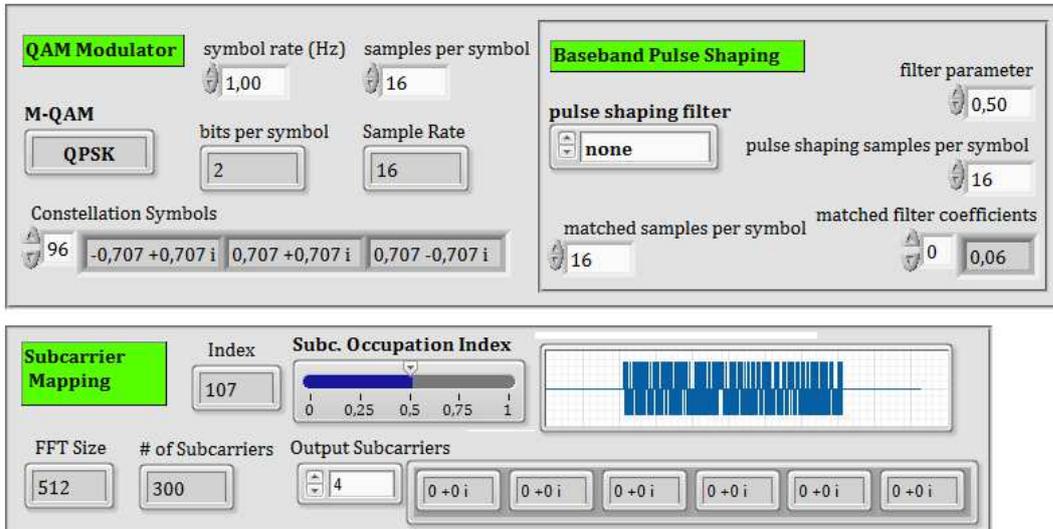


Figure C.33: Modulation and subcarrier mapping block GUIs of the transmitter in LabVIEW.

Figure C.34 illustrates an example where the subcarriers are placed at the beginning (Index 0) of the symbol and the guard band only at the end.

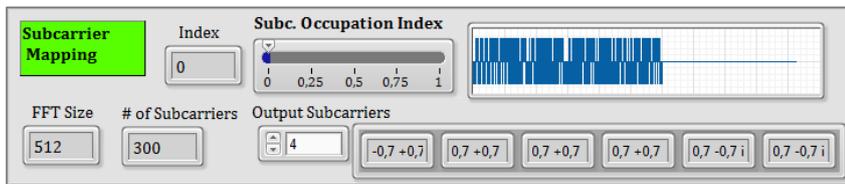


Figure C.34: Subcarrier mapping block GUIs with back guard band.

Figure C.35 illustrates an example where the subcarriers are placed at the end (Index 212) of the symbol and the guard band only at the beginning. Let us note that the last index value possible is 212 because if FFT size is 512 and number of subcarriers 300, we deduce that $212 + 300 = 512$ FFT symbol size.

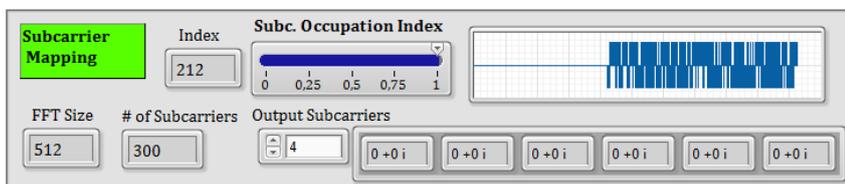


Figure C.35: Subcarrier mapping blocks GUI with front guard band.

We implemented several filter shapes for the pulse shaping block. These were selected in the GUI and the available choices are shown in the Figure C.36.

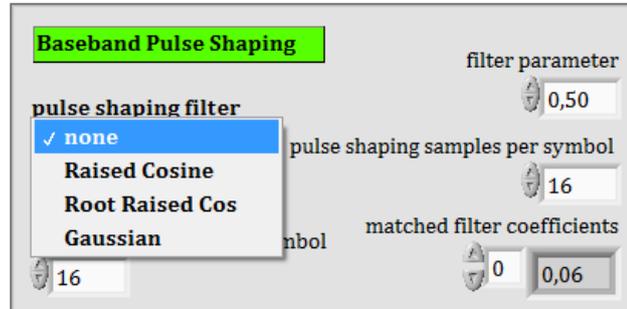


Figure C.36: Pulse shaping filter options GUI of the pulse shaping block.

The last to GUI blocks were the IFFT + cyclic prefix and subcarrier spacing, see Figure C.37. Only in these blocks was possible to select a value for the cyclic prefix length. The other parameters were representative.

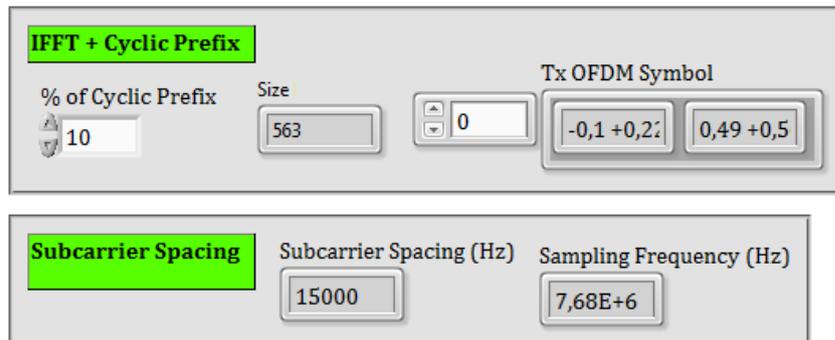


Figure C.37: IFFT, cyclic prefix and subcarrier spacing block GUIs of the transmitter in LabVIEW.

In the channel propagation model we designed the boolean switches that enabled/disabled the AWGN and Rayleigh fading. Furthermore, we defined in this GUI the fading parameters *delay spread*, *doppler spread* and *fading variance*. Apart from the parameters we plotted graphs for the fading profile generated and the constellation symbols at the transmitter and another for the constellation symbols received. In that way we monitored the effects of the channel distortions. All these elements are part of the channel model GUI depicted in Figure C.38.

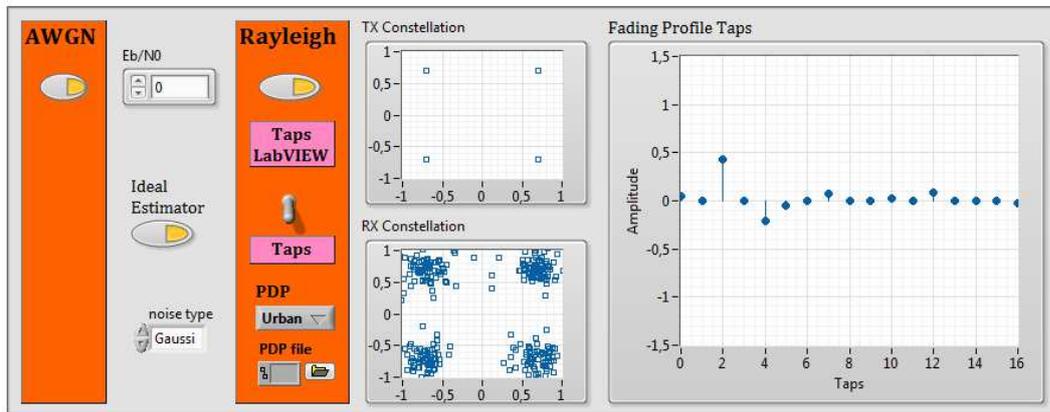


Figure C.38: Channel Model GUI of the Digital Communication system in LabVIEW.

At the receiver side, we monitored the first three blocks where its main variables were represented, see Figure C.39. At the channel estimation block we represented the channel response (H) array and also the three LEDs that represented if the pilots were processed (*Pilots?*), if the channel was estimated (*Ch. Est.*) and if the *Equalizer* was enabled. Let us mention that these boolean variables were important in the design and implementation of several blocks at the receiver.

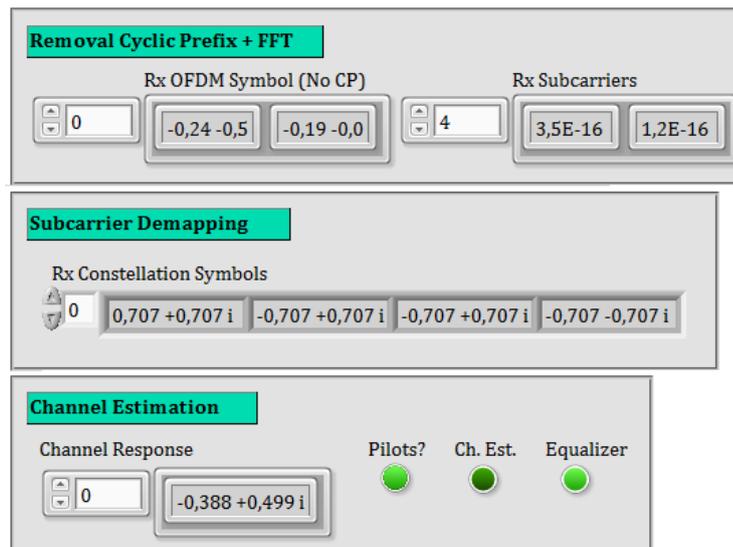


Figure C.39: Removal cyclic prefix and FFT, subcarrier demapping, channel estimation block GUIs of the receiver in LabVIEW.

Following with the last receiver modules, we represented the output variables of the blocks for monitoring purposes. We show the last three blocks of the receiver, demodulation, channel decoding and convolutional decoding in the Figure C.40. The received bits array Rx Bits, are the estimated bits that should be equal to the bits sent at the transmitter, if the communication system was ideal, no errors.

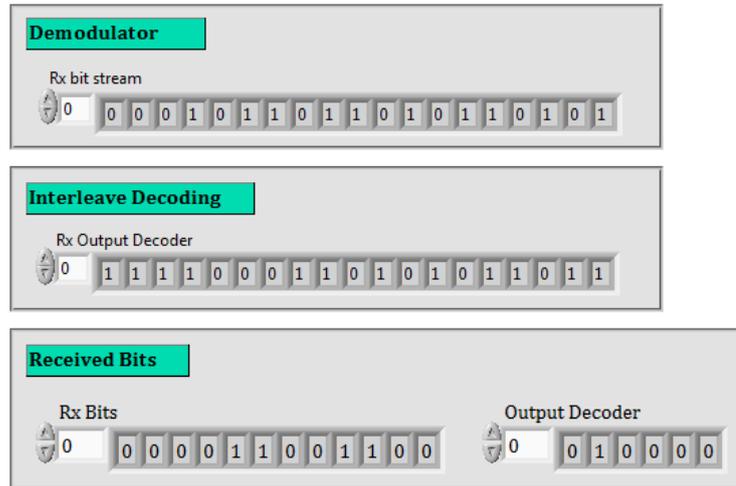


Figure C.40: Demodulation, channel decoding and received bits block GUIs of the receiver in LabVIEW.

We present in Figure C.41 the GUI of the block that calculated the BER and BLER. As we see in the BER interface, we monitored the number of blocks/frames sent, the E_b/N_0 selected and in the left side the instantaneous BER and wrong bits per frame. The *iteration step* counter represents number of the block/frame being processed, where the first one is the iteration 0, and thus the 10th block the 9th iteration. On the right side, the calculus of the total wrong bits, total transmitted bits and accumulated BER were represented in relation to the number of blocks sent.

While the simulation ran and the frames were sent and received, these performance measurements were updated on time. On the other hand, we monitored the BLER measuring the number of blocks sent, the number of wrong blocks. The *Bit threshold x Block* horizontal control bar defined the allowed error bits per frame, where in our case was set to 0 meaning that for a single bit error all the block was considered wrong.

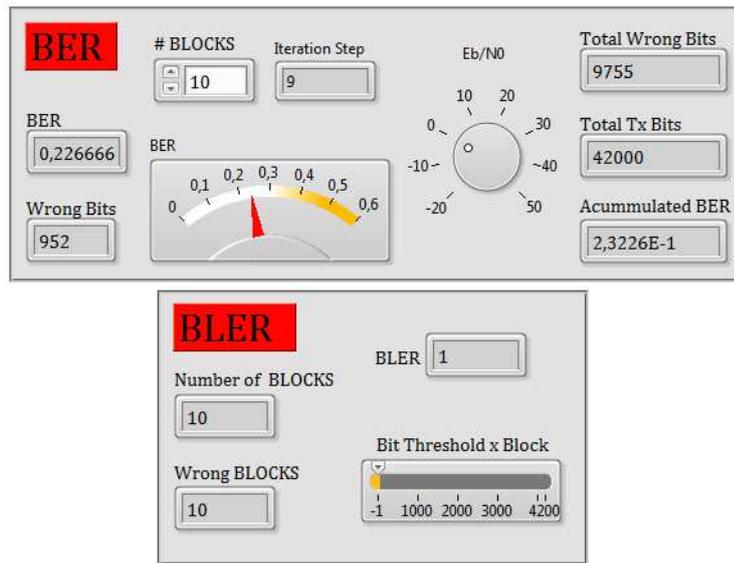


Figure C.41: BER and BLER calculation GUI in LabVIEW.

The last GUI block designed was the BER/BLER vs E_b/N_0 plots. The Figure C.42 illustrates the control panel or parameter definition interface where the range of the BER and SNR were set. As the plots needed several iterations per every different E_b/N_0 value, we implemented an ON/OFF switch button *Plot BER?*. In this way, we plotted when necessary. We observe that the accessible variables are only the *Start E_b/N_0* value, the *Step Size* and the *End E_b/N_0* value. The step size is the increment of E_b/N_0 desired from the start to the end value of SNR. Figure C.42 illustrates a simulation example were 11 SNR points were calculated and represented in the respective variable. Then, the iteration SNR increments while the simulation runs for each SNR point. The calculated BER and BLER arrays are stored and represented in the variables shown in the interface.

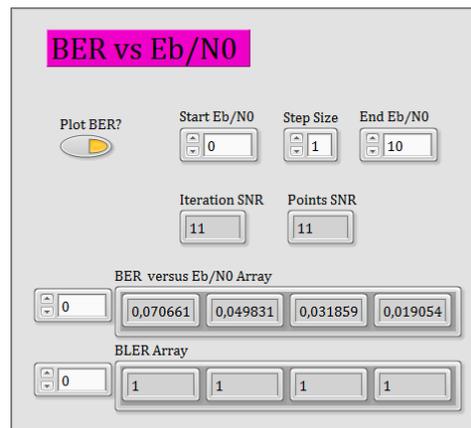


Figure C.42: BER/BLER vs. SNR plot parameters GUI in LabVIEW.

While the simulation ran and the arrays were updated, the plot of the BER vs. SNR was also updated. The final plot after the complete simulation for an example case of a QPSK modulation over an AWGN channel is shown in Figure C.43. We defined in the control panel of Figure C.42 a simulation from 0 to 10 Eb/N0 values and step size 1. As a result, 11 points SNR were obtained for the plot and the BER values of each SNR value is finally plotted as in Figure C.43.

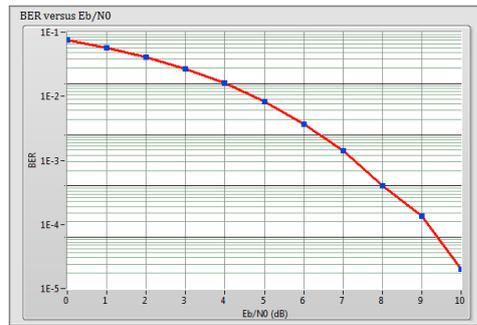


Figure C.43: BER vs. SNR plot GUI in LabVIEW.

Having the same parameters defined for simulations for BER and BLER, the last plot was obtained with the BLER array values. Figure C.44 depicts the results of the same example case of the previous BER graph.

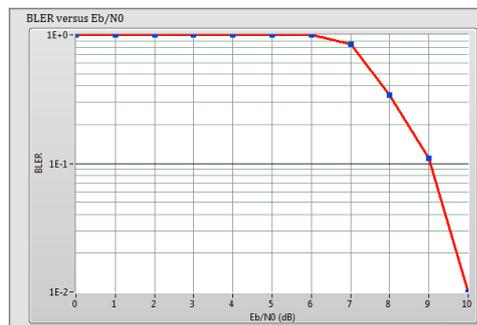


Figure C.44: BLER vs. SNR plot GUI in LabVIEW.

We described our complete GUI of the LabVIEW design. In order to give an overall view of how all was designed together, we presented in Figure 3.31 the general graphic user interface designed in our SDR at labVIEW. All the blocks were sequentially designed from the top to the bottom following the transmitter-channel-receiver structure.

While simulations were running, all the represented variables and arrays, constellation graphs, channel profiles, BER/BLER calculations and plots were updated in time. However, as explained before due to the registers the FOR loops iterated one time more in order to gather all the previous values and represent the performance measurements and plots, such as accumulated BER and BLER and its respective BER/BLER vs. SNR graphs.

C.2 SIMO System

C.2.1 SIMO Channel

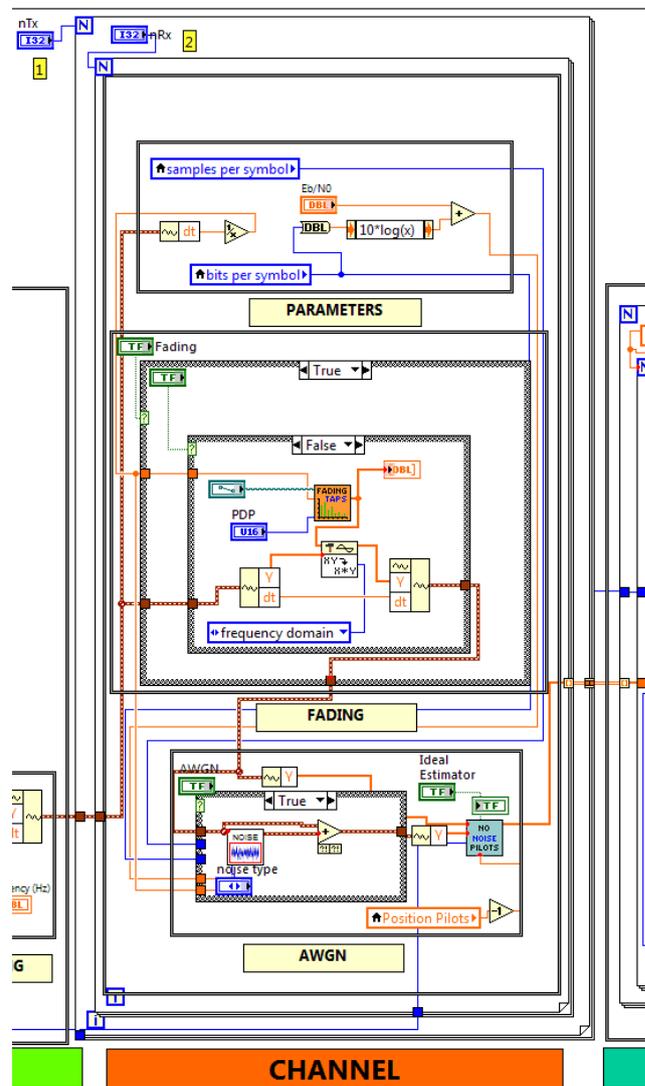


Figure C.45: SIMO Channel Design.

C.3 USRP Design

C.3.1 2x2 MIMO Transmitter

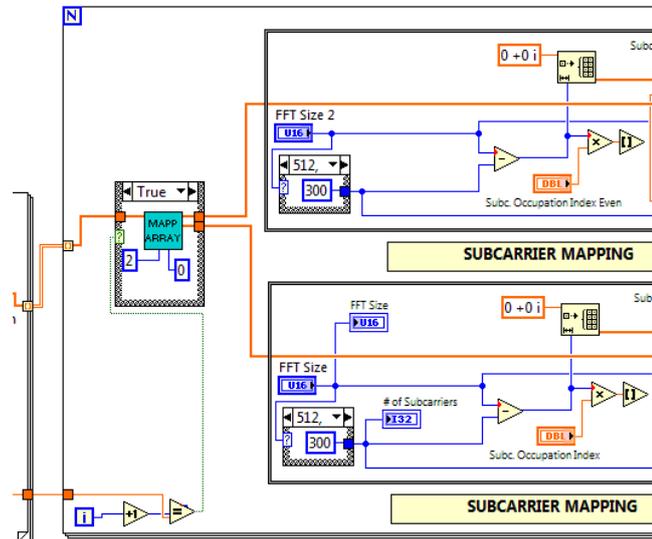


Figure C.46: USRP MIMO LV Transmitter Design.

C.3.2 2x2 MIMO Receiver

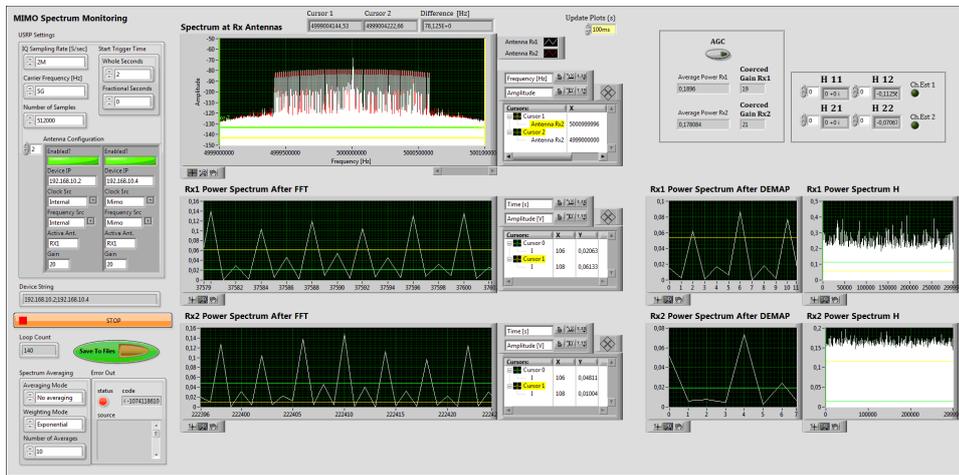


Figure C.47: USRP MIMO Receiver GUI.

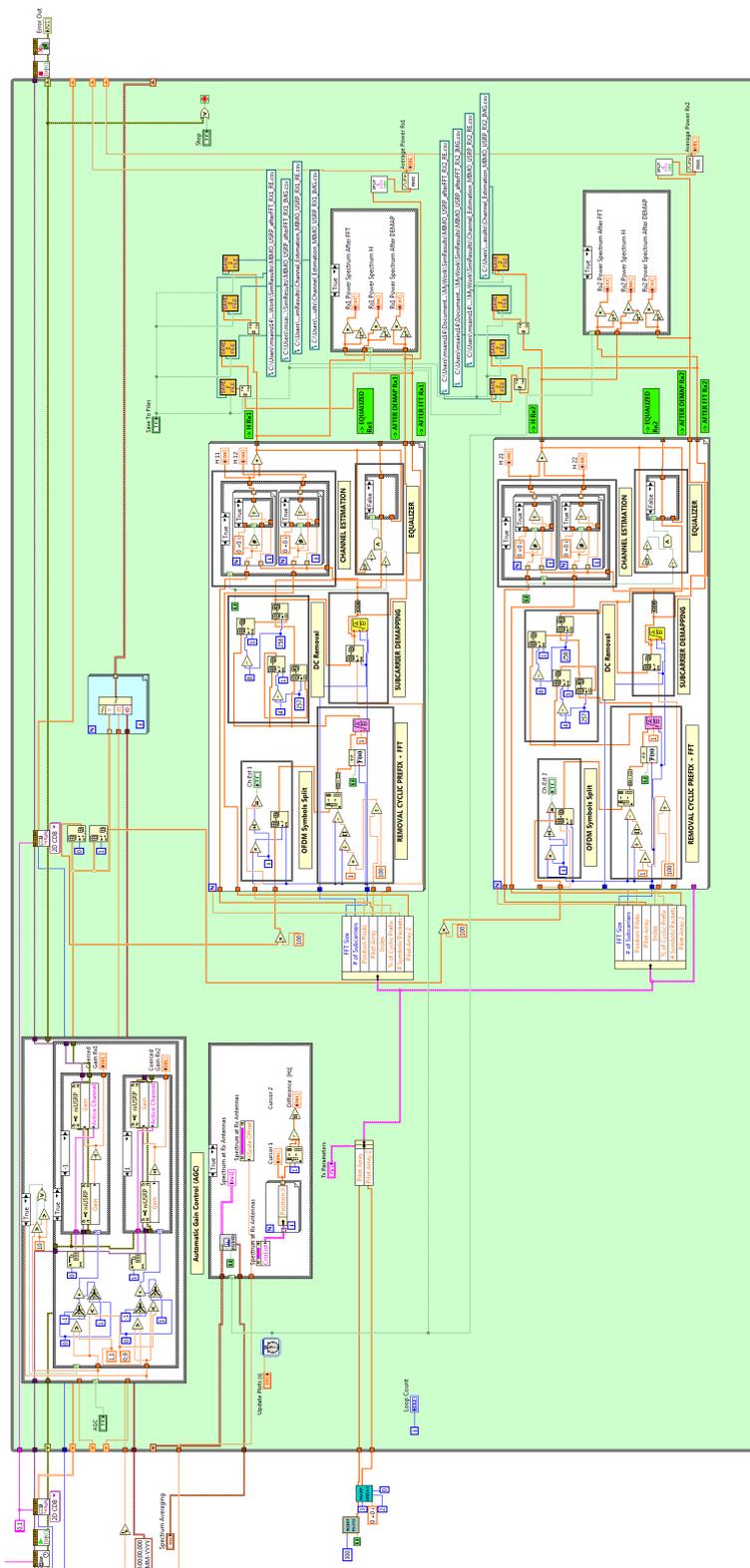


Figure C.48: USRP MIMO Receiver LV Design.

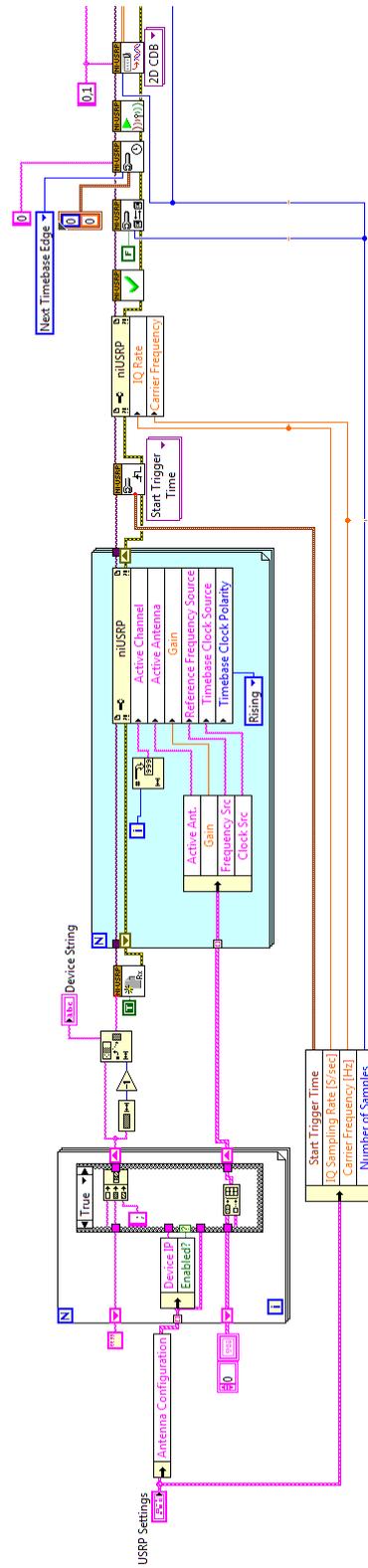


Figure C.49: USRP MIMO Receiver Configuration.

Appendix D

Additional Simulation Features

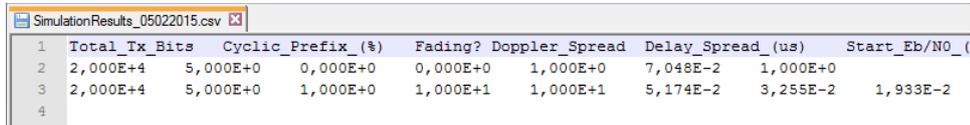
In this section we present and describe additional LabVIEW implementations which added useful features for the simulations. These features are not strictly related to digital communication systems, but with documentation, user interface flexibility and programming utilities, among others.

D.1 Save Results to File

This feature is called 'Save Results to File', and as it clearly defines offered to the user the option of recording the simulation results such as BER and BLER into a file. Before this feature, the calculated values of the BER and BLER and its plots were shown once and for each new simulation the old values were lost. Therefore, in order to keep a record and save the calculations of our simulations we decided to add this feature, which allowed us to document each test.

The format chosen for the data file was *.txt* and *.csv*, which by using TABs we could arrange all the parameters in a single row. At the header of the file, with an independent string row, we added the list or names of the variables that would be saved, in respective order. Then, we were able to decide whether we store one simulation results per file or append multiples simulation results in the same file but successive rows. Furthermore, apart from the BER and BLER calculated values we saved also the setup parameters of the simulation. As a result, a reliable simulation record is obtained while all the parameters are tracked. For example, multiple simulations in a single file could have different modulation scheme, transmitted bits or SNR points. We added most of these setup parameters which were listed in the header and inserted before the BER and BLER values. In this way, we obtained rows which independently were determined by these parameters. As a result, all the values were stored in the spreadsheet files, which could be used with other software tools such as MATLAB.

We illustrate in Figure D.1 an example of the output file using this feature for two simulation tests (two rows).



	Total_Tx_Bits	Cyclic_Prefix_(%)	Fading?	Doppler_Spread	Delay_Spread_(us)	Start_Eb/N0_(
1	2,000E+4	5,000E+0	0,000E+0	0,000E+0	1,000E+0	7,048E-2
2	2,000E+4	5,000E+0	1,000E+0	1,000E+1	1,000E+1	5,174E-2

Figure D.1: Simulation results file using the *Save Results to File* feature.

In order to implement this simulation record feature, we implemented it as we explain in the diagram design sub-section. On the other hand, the GUI to define the desired file name and path is presented in the GUI design sub-section.

D.1.1 Diagram Design

As explained before, LabVIEW runs all the blocks simultaneously a exception of FOR loops or similar structures. Therefore, it was necessary to implement a flat sequence structure as shown in Figure D.2. This structure executes each part sequentially. As we implemented the feature, we needed to sequentially obtain the parameters values, create the file, add the header to the file and add the values of the variables. In order to do that, each part had to run in a defined order



Figure D.2: Processes of the *Save Results to File* feature.

In the first step, all the simulation was done, the values calculated and with the parameters, all was grouped into a compact variable packet. The second step, the file path and name was defined and it was enabled the option of new file creation or re-editing existing file. After the file was created or existed, the third step was to add the header to the file, the list of all the variables. If the file was new generated, then the header was attached in the first line. If the file exist, then no new header was needed. The fourth step was to save all the values to the file, with the option of append them to previous simulations.

To summarize, all the sequence had to be computed independently, because for example if we added the values to a file, which did not exist then the simulation will break. Another possibility was to store values before the simulation was finished. Consequently, wrong variable values or empty variables could have been saved into the file. Using this flat sequence structure we assured reliability and that all the file verification processes and variables calculations were done before anything was saved into the file.

We illustrate in Figure D.3 the implementation of the packet variable which contained all the variables. This was computed in the first sequence structure.

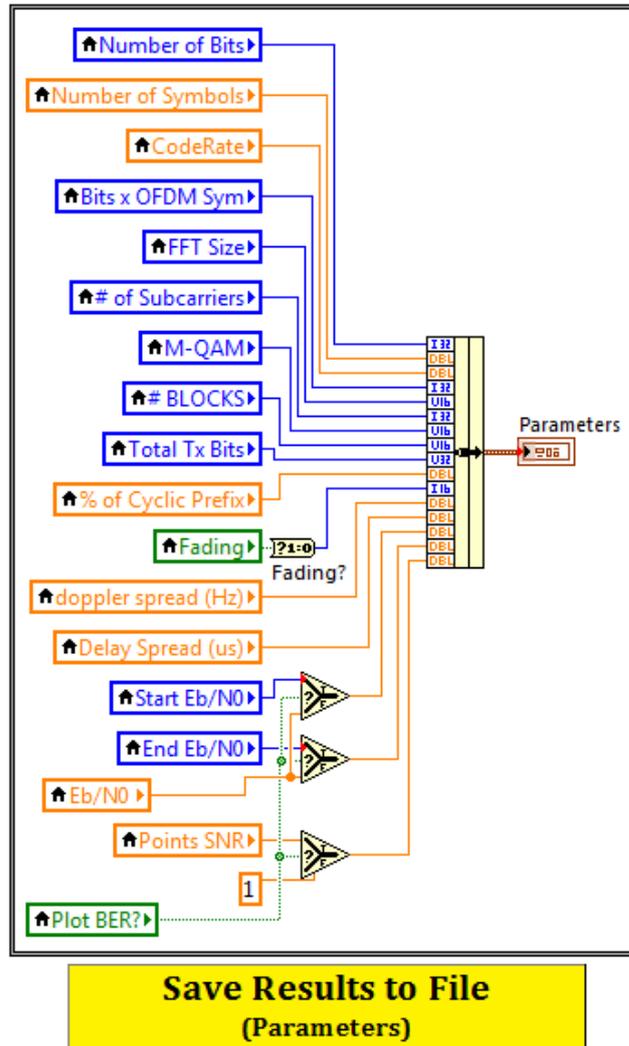


Figure D.3: Parameters of the *Save Results to File* feature.

The second sequence was mostly related to the user interface, where the file path and name was introduced and the file created or deleted. We depict its design implementation in Figure D.4. If the file existed but we decided to write new values we could append the new values to old ones or clear the file, which for us meant *reset* the file. In order to do that, the previous file was deleted and a new one with a new list of variables was created in the third sequence structure, illustrated in Figure D.5.

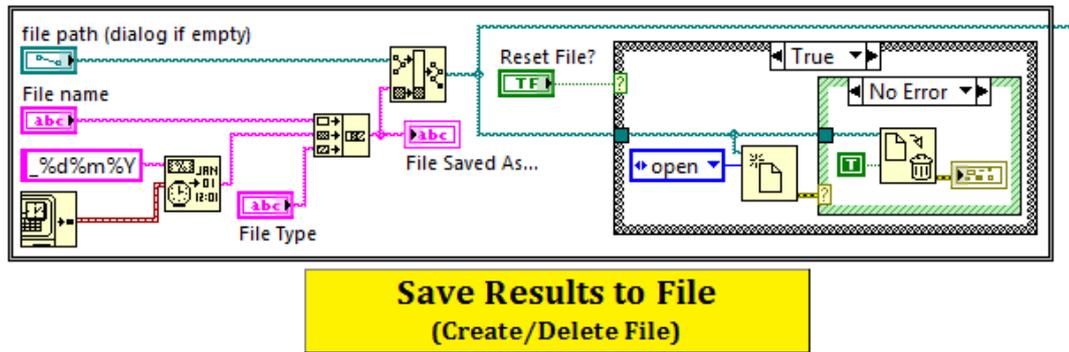


Figure D.4: File generation of the *Save Results to File* feature.

The implementation of this block, basically it tries to open an existing file and if it does not exist, the execution outputs an error which we used to generate a new file with the list of the variables name. If there was no error opening the specified file, then the file was closed again and no further operations were done. This case was for appending simulation values in the same file.

The last block regardless creation or edition of the file, it appended into it the current values of the simulation done in the first sequence. The parameters and the BER/BLER were saved into the file, each simulation in an independent file row. Since our simulation design calculated the BER/BLER for a fixed E_b/N_0 or plotted several BER/BLER vs. SNR, we created a box case which allowed us to store the BER/BLER results of a single E_b/N_0 value, or the BER/BLER arrays of the simulations that were computed to obtain the graphs of BER/BLER vs. SNR. In this way, we analyzed independent results for specific E_b/N_0 points in the stored files. We present the block design in Figure D.6.

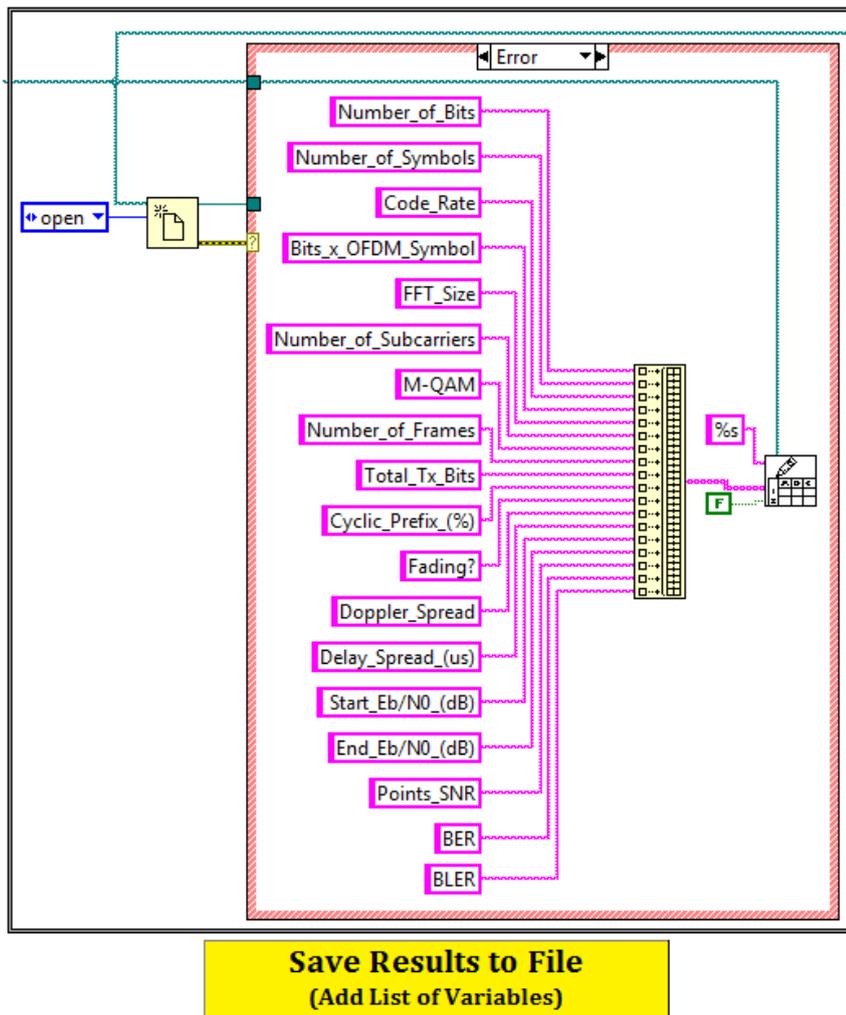


Figure D.5: List of variables of the *Save Results to File* feature.

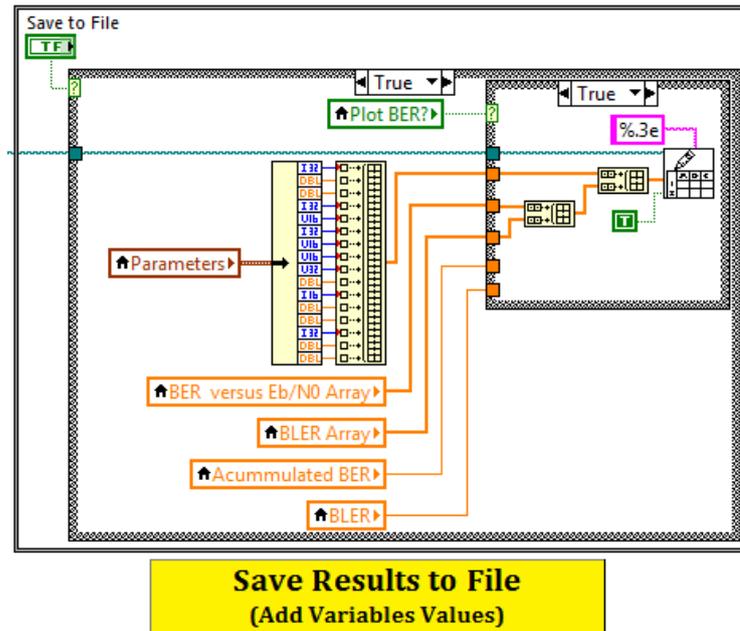


Figure D.6: Adding variables values of the *Save Results to File* feature.

D.1.2 GUI Design

The user interface is presented in Figure D.7, where the user chose the file name, file type (.csv or .txt) and the file path. Furthermore, the options of clear the file for each simulation was enabled with the switch button *Reset File?* and the step of saving the values into the file was enabled with the switch button *Save to File*.

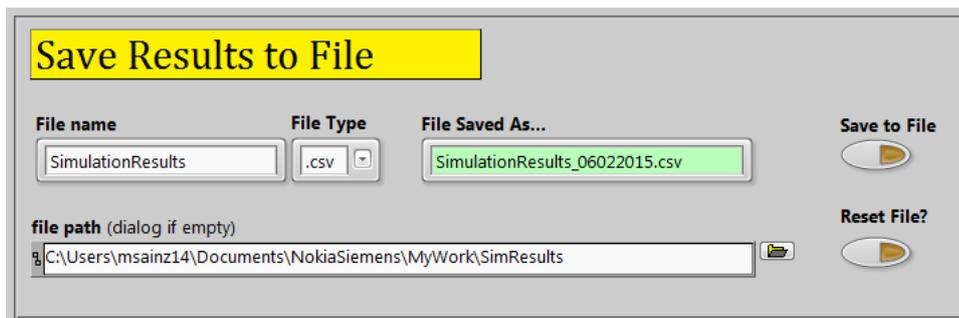


Figure D.7: GUI of the *Save Results to File* feature.

D.2 MATLAB Scripts

D.2.1 Get Throughput and Obtain the CDF

```

1 nSymbols=1000;
2 nSubcarriers=300;
3 % Layer 1 PRECODE_MATRIX=1,2,3,4
4 % Layer 2 PRECODE_MATRIX=5,6,7
5 PRECODE_MATRIX=2; %Selects fixed matrix
6
7 [ Z ] = generate_precoding_matrix_( );
8 throughput=zeros(nSymbols,7);
9 thbest=zeros(nSymbols,1);
10
11 for indxpacket=1:nSymbols %Calculates throughput of every packet/Symbol
12
13 nPacket=indxpacket;
14
15 % Calculate Noise from the guard bands of every Symbol only if subcarriers are
    centered
16 an=((nPacket-1)*512)+2;
17 bn=((nPacket-1)*512)+80;
18 Rn1=horzcat(RX1(an:bn),RX1(an+415:bn+415));
19 Rn2=horzcat(RX2(an:bn),RX2(an+415:bn+415));
20 avgRn1=sum(Rn1.*conj(Rn1))/size(Rn1);
21 avgRn2=sum(Rn2.*conj(Rn2))/size(Rn2);
22 avgRn=mean([avgRn1 avgRn2]);
23 Rn=[avgRn 0 ; 0 avgRn]; %Noise matrix
24 RndB=10*log10(Rn(1));
25
26 for iprec=1:7 %Calculates throughput with each precoding matrix
27
28 PRECODE_MATRIX=iprec;
29
30 if PRECODE_MATRIX>4
31     PRECODE_MATRIX=PRECODE_MATRIX-4;
32     nLayer=2;
33 else
34     nLayer=1;
35 end
36 PREC=Z{nLayer,PRECODE_MATRIX}; %Obtain the precoding matrix
37
38 a=((nPacket-1)*nSubcarriers)+1; %Calculates the position of the subcarriers
    within symbol
39 b=nPacket*nSubcarriers;
40
41 % Get the Channel Coefficients of a Symbol
42 Hd=[H11(a:b) H21(a:b) H12(a:b) H22(a:b)];
43 avgRn1=sum(Rn1.*conj(Rn1))/size(Rn1);

```

```

44 Hi=[0 ;0];
45
46 N=length(Hd)/4;
47 if nLayer==1
48     nSNR=0;
49     sinr1=zeros(1,N);
50     for i=1:4:nSubcarriers %Selects only the subcarriers with pilots
51         subHd=[Hd(i+2,1) Hd(i,3); Hd(i+2,2) Hd(i,4)];
52         PRECsubHd=subHd*PREC; %Applies the precoding matrix to the channel
53         nSNR=nSNR+1;
54         sinr1(nSNR) = calculate_sinr( PRECsubHd , Hi, Rn);%Calculates SINR of
                    each Subcarrier
55     end
56     sinr1_comp = calculate_SINR_comp( sinr1 , 1);%Calculate the averaged SNR
                    within the Symbol
57     FR_ratio=1;
58     throughput(indxpacket,iprec)=calculate_Throughput(sinr1_comp , FR_ratio);
59     if throughput(indxpacket,iprec)>thbest(indxpacket)
60         thbest(indxpacket)=throughput(indxpacket,iprec); %Identifies
                    the maximum throughput precoding matrix
61     end
62
63 elseif nLayer==2 %Calculation for the Layer 2 precoding matrices
64     nSNR=0;
65     sinr2=zeros(2,N);
66     for i=1:4:nSubcarriers %Calculates SINR each Subcarrier
67         subHd=[Hd(i+2,1) Hd(i,3); Hd(i+2,2) Hd(i,4)];
68         PRECsubHd=subHd*PREC; %Applies the precoding matrix to the channel
69         nSNR=nSNR+1;
70         calcsinr = calculate_sinr( PRECsubHd , Hi, Rn );%Calculates SINR of
                    each Subcarrier
71         sinr2(1,nSNR)=calcsinr(1);
72         sinr2(2,nSNR)=calcsinr(2);
73     end
74     sinr2_comp(1) = calculate_SINR_comp(sinr2(1,:) , 1);
75     sinr2_comp(2)= calculate_SINR_comp(sinr2(2,:) , 1);
76     FR_ratio=1;
77     ThLayer1=calculate_Throughput(sinr2_comp(1) , FR_ratio);
78     ThLayer2=calculate_Throughput(sinr2_comp(2) , FR_ratio);
79
80     throughput(indxpacket,iprec)=ThLayer1+ThLayer2;
81     if throughput(indxpacket,iprec)>thbest(indxpacket)
82         thbest(indxpacket)=throughput(indxpacket,iprec);%Identifies
                    the maximum throughput precoding matrix
83     end
84 end %elseif layer
85 end %for precoding matrix
86 end %for packets
87

```

```

88 %Plots and CDF Calculations
89 th2=throughput(:,2);
90 figure;
91 hold on;
92 ecdf(th2);
93 ecdf(thbest);
94 legend('Fixed Precoding Matrix','Precoding Matrix SNR max');
95 xlabel('Throughput [Mbps]');

```

D.2.2 Generate Precoding Matrix

```

1 function [ Z ] = generate_precoding_matrix_( )
2     Z = cell(2,4); % [ layers ( 1 or 2 layers ) x codes ( 0 to 3 ) ]
3
4     Z{ 1, 1} = [ 1; 1 ] / sqrt( 2 );
5     Z{ 1, 2} = [ 1; -1 ] / sqrt( 2 );
6     Z{ 1, 3} = [ 1; 1i ] / sqrt( 2 );
7     Z{ 1, 4} = [ 1; -1i ] / sqrt( 2 );
8
9     Z{ 2,1} = [ 1 0; 0 1 ] / sqrt( 2 );
10    Z{ 2,2} = [ 1 1; 1 -1 ] / 2;
11    Z{ 2,3} = [ 1 1; 1i -1i ] / 2;
12 end

```

D.2.3 Calculate SINR

```

1 function sinr = calculate_sinr( Hd, Hi, Rn)
2
3 Rr = Hd * Hd' + diag( diag( Hi * Hi' ) ) + Rn;
4 W = Rr \ Hd;
5
6 Rin = Hi * Hi' + Rn;
7
8 ns = size( Hd, 2 );
9 idx = 1 : ns;
10
11 sinr = nan( 1, ns );
12
13     for k = 1 : size( Hd, 2 )
14         ik = idx( idx ~= k );
15         sinr( k ) = real( ( W( :, k )' * ( Hd( :, k ) * Hd( :, k )' )
16             * W( :, k ) ) / ...
17             ( W( :, k )' * ( Hd( :, ik ) * Hd( :, ik )' + Rin ) * W(
18                 :, k ) ) );
19     end
20 end

```

D.2.4 Average SINR of All the Subcarriers

```

1 function sinr_comp = calculate_SINR_comp( sinr_vec , layers)
2
3     for i=1:length(sinr_vec)
4         sinr_evm(i) = (sinr_vec(layers,i) * 400) / ( sinr_vec(layers,i)+ 400);
5         I(i)=log2(1+ sinr_evm(i));
6     end
7
8     sinr_comp=(2 ^ mean(I)) -1 ;
9
10 end

```

D.2.5 Calculate Throughput

```

1 function throughput=calculate_Throughput ( sinr_ , FR_ratio)
2 max_se=8;
3 for i=1:length(sinr_)
4
5     I(i)=log2( 1+ sinr_(i));
6     if(I(i) > max_se)
7         I(i)=max_se;
8     end;
9     throughput(i)=(2 * I(i) )/FR_ratio; %I/Q Rate 2 Mhz
10
11 end
12 end

```