Development of a MATLAB application to generate NOMA-CAP 5G signals

A Degree Thesis

Submitted to the Faculty of the

Escola Tècnica d'Enginyeria de Telecomunicació de Barcelona

Universitat Politècnica de Catalunya

by

Marc Muro Barbé

In partial fulfilment

of the requirements for the degree in

TELECOMMUNICATION TECHNOLOGIES AND SERVICES ENGINEERING

MENTION IN TELEMATIC SYSTEMS

Advisor: José A. Lázaro

Co-Advidor: Samael Sarmiento

Barcelona, May 2020

Abstract

The demand over high speed fixed and wireless networks increases every day. NOMA-CAP is being studied as a way to improve the spectral efficiency of 5G signals. That is why the objective of this thesis is to develop an application that allows us to freely generate simulated 5G NOMA-CAP signals with the settings we desire making it easier its study. We will also use said application to do three tests with different configurations that make use of all the new possibilities that the application brings.

Resum

Actualment la demanda per 5G augmenta constantment. S'està estudiant NOMA-CAP com un mètode per incrementar la eficiència espectral de 5G. L'objectiu d'aquesta tesis és desenvolupar una aplicació que ens permeti generar senyals 5G NOMA-CAP amb les característiques que volem, fent així més fàcil el futur estudi de NOMA-CAP. Després utilitzarem aquesta aplicació per provar tres situacions diferents que utilitzin les diferents opcions que ofereix l'aplicació.

Resumen

Actualmente la demanda por una comunicación inalámbrica rápida aumenta cada día. NOMA-CAP se está estudiando como una manera de mejorar la eficiencia espectral de 5G. El objetivo de esta tesis es desarrollar una aplicación que nos permita generar señales 5g NOMA-CAP con las características que deseemos, haciendo así más fácil el estudio de NOMA-CAP. Luego usaremos la aplicación para estudiar tres casos distintos que usaran los distintos modos que ofrece la aplicación.

Acknowledgements

I want to thank my advisor, José A. Lázaro, and co-advisor, Samael Sarmiento, for all the help and support that I received from them. From the initial planning of the application to fixing some problems I had with the code as well as the ideas they gave me to make this project a success.

Revision history and approval record

Revision	Date	Purpose
0	14/04/2020	Document creation
1	21/04/2020	Document first revision
2	27/04/2020	Document second revision
3	08/05/2020	Final revision
4	15/05/2020	Mistake correction

DOCUMENT DISTRIBUTION LIST

Name	e-mail
Marc Muro Barbé	marc.muro@estudiant.upc.edu
José A. Lázaro	jose.lazaro@tsc.upc.edu
Samael Sarmiento	samael.sarmiento@tsc.upc.edu

Written by: Marc Muro Barbé		Reviewed and approved by:	
Date	08/05/2020	Date	15/05/2020
Name	Development of a MATLAB application to generate NOMA-CAP 5G signals	Name	José A. Lázaro
Position	Author	Position	Project Supervisor

Table of contents

List of Figures

List of Tables:

1. Introduction

1.1. Statement of purpose

This project plans to study NOMA-CAP 5G signals. To do so we will make a MATLAB application that will allow us to manage and simulate the transmission of NOMA-CAP signals.

Regarding the app, we will be able to choose to modify different aspects of the signal we are going to transmit. We will be able to choose, for example, the number of bands, bandwidth of the signal, modulation, power relation between users… Not only can we choose the number of bands, but we can also choose different settings between the different bands.

With the help of the application and the code of the transmitter/receiver, we will generate/recover different NOMA-CAP signals to study under different simulation settings. We will study the SNR/BER relation under these different simulations.

1.2. Requirements and specifications

On the first weeks of the project I had to learn about NOMA, multiCAP and NOMA-CAP. To do so I used different resources, specially the paper "*Nonorthogonal Multiple Access and Carrierless Amplitude Phase Modulation for Flexible Multiuser Provisioning in 5G Mobile Networks*" by Jose Antonio Altabas and Simon Rommel [1].

Following that, I had to review my MATLAB knowledge and learn from scratch how to develop MATLAB APPS. I used different online guides for that.

1.3. Methods and procedures

This work is a continuation of an already existing project where NOMA-CAP is proposed as a method of improving 5G efficiency. The main code of the transceiver and receiver were also provided to me. My job was then to develop an application that converted the transceiver code into a function where we could easily modify the different inputs of the code.

1.4. Work Plan

Project: Learning about NOMA-CAP	WP ref: WP 1
Reading and understanding different papers about NOMA-CAP. In order to have the basic knowledge of NOMA, multiCAP and NOMA-CAP to understand the given code.	Start date:16/09 End date: 23/09

Project: Refreshing knowledge about MATLAB	WP ref: WP2
Major constituent: Being able to write MATLAB code	Start date:23/09 End date: 07/10
While I have used MATLAB before, it is not a program I use frequently. My objective is to refresh my knowledge about how MATLAB work.	
Internal task T1: Read some guides about MATLAB Internal task T2: Read and understand the code	

Project: Learning about MATLAB APPS	WP ref: WP3
Major constituent: Being able to develop a MATLAB APP	Start date:07/10 End date: 21/10
MATLAB applications are a new world to me. In order to fulfill the objective of this project, we believe this is the best approach, as an application gives the user an easy and understandable UI. With the help of guides, I will then learn how to write a MATLAB APP.	

Project: Develop the application	WP ref: WP4
Major constituent: Develop a working application	Start date:21/10 End date: 30/11
Now that I have all the knowledge required, I will develop an application that allows the user to simulate a NOMA-CAP 5G signal with the configuration he desires for.	
Internal task T1: Develop an application for regular Bands Internal task T2: Develop a way to open N number of tabs Internal task T3: Develop an application for different Bands	

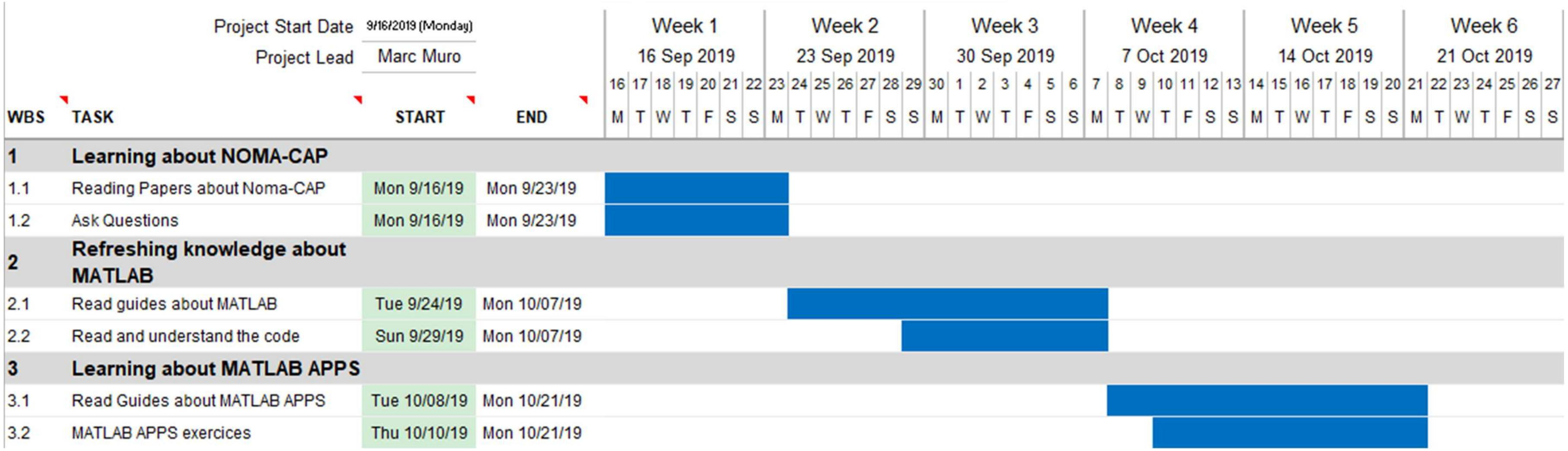
Project: Modify the code for multiple Bands	WP ref: WP5
Major constituent: Make the application work in any condition	Start date:01/12 End date: 22/12
The original given code only works if all the bands have the same settings, but one of the major points of the application is to let the user set each band differently. The original code must then be modified in order to be fully compatible with all of the application settings.	
Internal task T1: Check errors so that we can see what needs to be adapted.	

Project: Modify the Transciever	WP ref: WP6
Major constituent: Modify the transceiver so that we can study the simulated signals	Planned start date:22/12 Planned end date: 21/01
Modify the Receiver part of the code so that I can study the simulation I generated using the App	
Internal task T1: Modify RX	

Project: Obtain Results from experimentation	WP ref: WP7
Major constituent: Study results taken using the app	Planned start date:22/01 Planned end date: 14/03
Do different simulations with the application, then use the modified transceiver to obtain SNR/BER relations and scatter plots	
Internal task T1: Study the results obtained	

1.5. Gannt Diagram

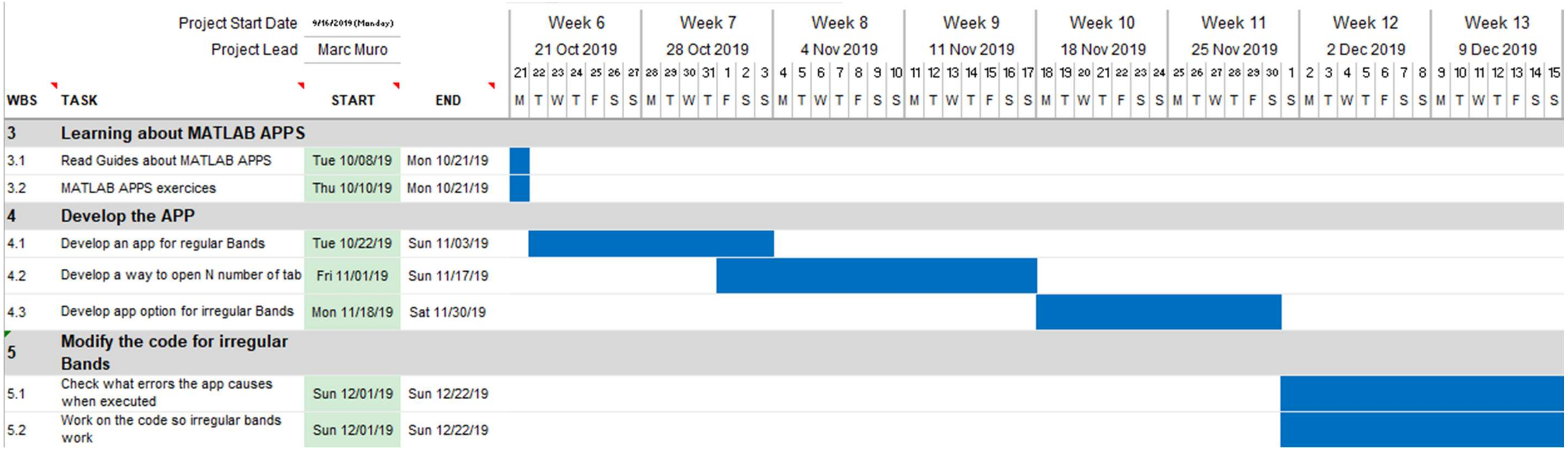
Gant diagram of the project will be posted in the next three pages:



Project Start Date 9/16/2019 (Monday)
Project Lead Marc Muro

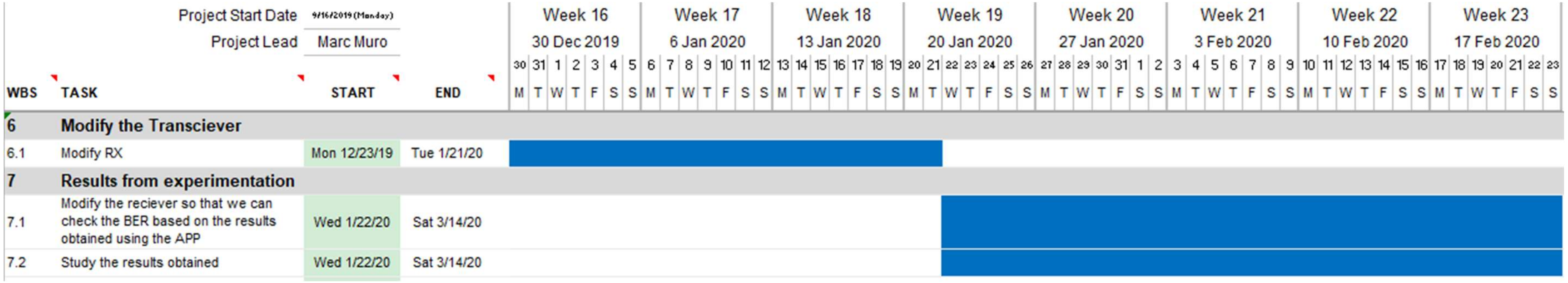
Week 1 16 Sep 2019
Week 2 23 Sep 2019
Week 3 30 Sep 2019
Week 4 7 Oct 2019
Week 5 14 Oct 2019
Week 6 21 Oct 2019

WBS	TASK	START	END
1	**Learning about NOMA-CAP**		
1.1	Reading Papers about Noma-CAP	Mon 9/16/19	Mon 9/23/19
1.2	Ask Questions	Mon 9/16/19	Mon 9/23/19
2	**Refreshing knowledge about MATLAB**		
2.1	Read guides about MATLAB	Tue 9/24/19	Mon 10/07/19
2.2	Read and understand the code	Sun 9/29/19	Mon 10/07/19
3	**Learning about MATLAB APPS**		
3.1	Read Guides about MATLAB APPS	Tue 10/08/19	Mon 10/21/19
3.2	MATLAB APPS exercices	Thu 10/10/19	Mon 10/21/19



	Project Start Date	9/16/2019 (Monday)		Week 6							Week 7							Week 8							Week 9							Week 10							Week 11							Week 12							Week 13						
	Project Lead	Marc Muro		21 Oct 2019							28 Oct 2019							4 Nov 2019							11 Nov 2019							18 Nov 2019							25 Nov 2019							2 Dec 2019							9 Dec 2019						
				21	22	23	24	25	26	27	28	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
WBS	TASK	START	END	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S
3	**Learning about MATLAB APPS**																																																										
3.1	Read Guides about MATLAB APPS	Tue 10/08/19	Mon 10/21/19	■																																																							
3.2	MATLAB APPS exercices	Thu 10/10/19	Mon 10/21/19	■																																																							
4	**Develop the APP**																																																										
4.1	Develop an app for regular Bands	Tue 10/22/19	Sun 11/03/19		■	■	■	■	■	■	■	■	■	■	■	■	■																																								
4.2	Develop a way to open N number of tab	Fri 11/01/19	Sun 11/17/19												■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■																												
4.3	Develop app option for irregular Bands	Mon 11/18/19	Sat 11/30/19																													■	■	■	■	■	■	■	■	■	■	■	■	■															
5	**Modify the code for irregular Bands**																																																										
5.1	Check what errors the app causes when executed	Sun 12/01/19	Sun 12/22/19																																						■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
5.2	Work on the code so irregular bands work	Sun 12/01/19	Sun 12/22/19																																						■	■	■	■	■	■	■	■	■	■	■	■	■	■	■

Task 7, Results from experimentation, was finished on March 3rd (week 26), with the two sub-tasks (7.1 and 7.2) running until that day. The final three weeks have been cut from the graph to avoid using additional space that would add little information.



Project Start Date 9/16/2019 (Monday)

Project Lead Marc Muro

WBS	TASK	START	END
6	**Modify the Transciever**		
6.1	Modify RX	Mon 12/23/19	Tue 1/21/20
7	**Results from experimentation**		
7.1	Modify the reciever so that we can check the BER based on the results obtained using the APP	Wed 1/22/20	Sat 3/14/20
7.2	Study the results obtained	Wed 1/22/20	Sat 3/14/20

1.6. Derivations

Once the main application was working, the next step was to make it work for bands with different specifications, this meant that the application should be able to have different tabs, one for each band. Not only that, but the information of each band had to be saved too, it also meant that the base code had to be changed too. This took far longer than expected, most of the application development and the main cause for the delay was found here.

2. State of the art of the technology used or applied in this thesis:

NOMA-CAP is a technique being studied for future 5G networks in order to address the increasing demand over high speed fixed and wireless networks [1][2][3][4][5][6][7]. As the name suggests it is a combination of nonorthogonal multiple access (NOMA) and multiband carrierless amplitude phase modulation (multiCAP).

In the following section we are going to review how NOMA, multiCAP function and how they work together with NOMA-CAP.

2.1. Non-Orthogonal Multiple Access

Non-orthogonal multiple access (NOMA) protocol can increase cell capacity making it a very promising multiple access technique between the different medium access control protocols for the fifth generation of mobile communication.

Multiple access (MA) techniques can be categorized into two types: as orthogonal multiple access (OMA) and nonorthogonal multiple access (NOMA).

First, we are going to review OMA in order to further understand NOMA. In OMA, in order to avoid interference from other users as done in OFDM [8] [9] [10], each user uses orthogonal communication resources those being either a different time slot, frequency band, or code. We can see the resource distribution of OMA in figure 1.

Previous generations of networks have used different OMA techniques. 1G uses frequency division multiple access (FDMA), 2G uses time division multiple access (TDMA), 3G uses code division multiple access (CDMA), and 4G uses orthogonal frequency division multiple access (OFDMA). However, these resources are in the end limited, and the demand for them increases every day.

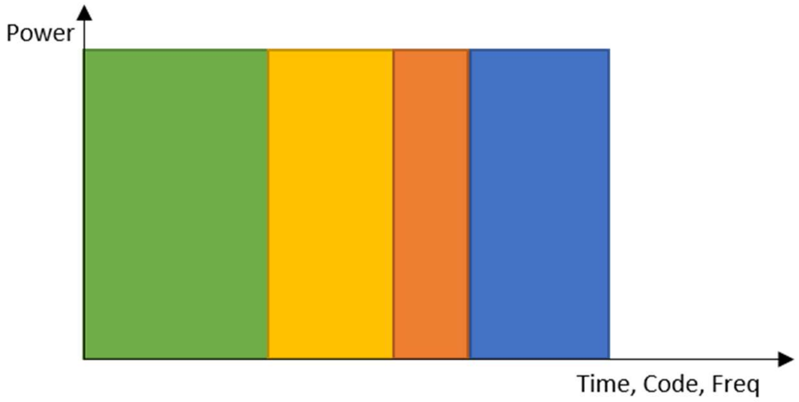


Figure 1 Orthogonal Multiple Access resource distribution

OMA performance makes it uncapable and unsuitable to provide for the needed services that future generations of wireless communication need. Researchers suggest that NOMA is a strong candidate for next generations. [11] [12][13]

In NOMA, multiple nonorthogonal resources are concurrently utilized by users, having this way a higher spectral efficiency compared to orthogonal methods while still allowing users multiple access at receivers. On Figure 2 we can see the resource distribution of NOMA. As seen in the picture it is much more efficient than OMA and multiple users can use the resources at the same time.

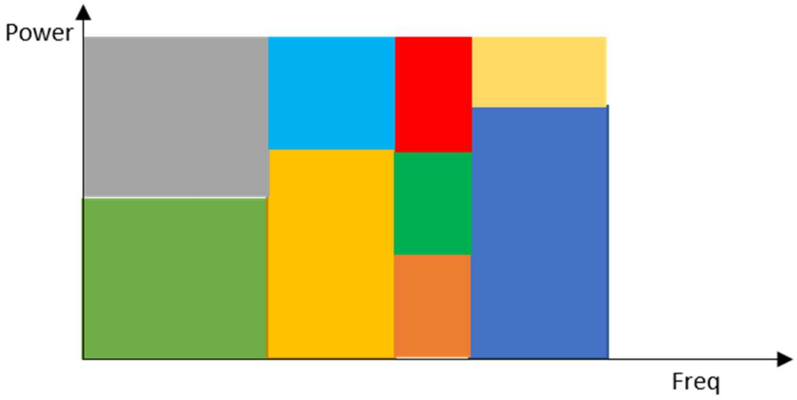


Figure 2 Non-Orthogonal Multiple Access resource distribution

In general, NOMA can also be classified into two types: power-domain multiplexing and code-domain multiplexing.

In power-domain multiplexing, depending on channel conditions, users are allocated different power coefficients so they can get the highest system performance possible. These different users' information is superimposed at the transmitter side. In order to decode the different signals side successive interference cancellation (SIC) is applied then at the receiver side. We can see how power-based NOMA downlink works in in figure 3.

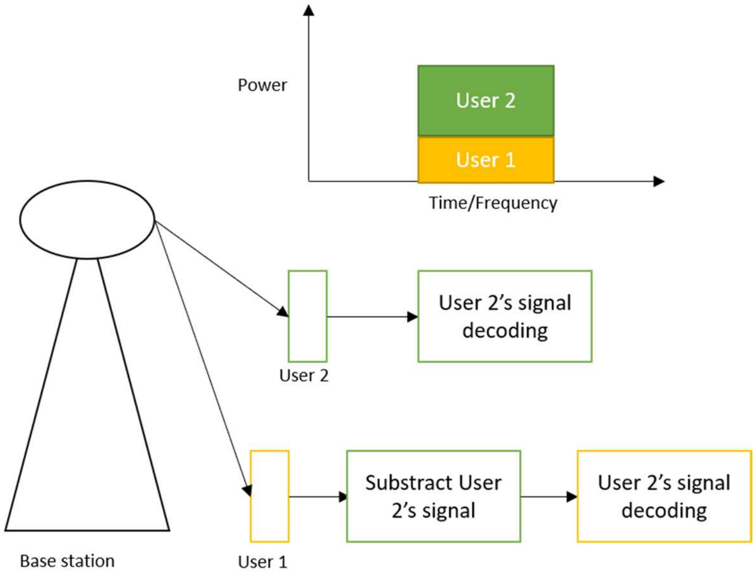


Figure 3 Downlink NOMA in a single cell with one base station and two users [18]

On the other hand, with code-domain multiplexing users use different codes and are multiplexed over the same time-frequency resources. Code-domain NOMA can also be divided depending on their multiple access techniques.

Besides these two types of NOMA schemes, there exist others such as pattern division multiple access (PDMA) and bit division multiplexing (BDM).

The main attractive advantage of power-domain multiplexing is that it has a simple implementation. Mainly due to it being already usable without the need of considerable changes on the existing networks. Also, it does not need additional bandwidth to improve spectral efficiency unlike code-domain multiplexing.

However, NOMA also has some limitations that should be addressed in order to be fully utilized for future generation communications.

The main one is that NOMA requires a higher computational complexity compared to OMA due to the need of users to decode signals of some users before decoding its own signal will lead to a longer delay.

2.2. Multiband carrierless amplitude phase modulation

Parallel optics appear as a valid short-term solution to the bandwidth limitation that comes with the growing data transmission demands over the limitations of short-range optical data link. However, its complex optical packaging makes it unviable on the long term.

That is why there is an increasing amount of research to achieve higher order modulation formats that can improve not only spectral efficiency but also reduce the transceiver complexity.

As an alternative carrierless amplitude phase (CAP) was proposed, but with a multiband approach (multiCAP). Thanks to multiCAP, we can achieve all three objectives: high spectral efficiency; increase the tolerance towards dispersion; reduce bandwidth limitations; and reduce the complexity at the transceiver. [14][15][16][17]

First of all, in order to explain multiCAP, we will make a short explanation of CAP and Discrete multitone transmission (DMT). CAP shares some similarities with QAM. Both are capable of transmitting parallelly two streams of data. However, the main difference is that, in contrast to QAM, CAP does not rely on a carrier. CAP uses orthogonal waveforms filters to separate the different data streams. This makes CAP receivers simpler than QAM receivers while achieving the same spectral performance and efficiency. We can see CAP's principle of operation in figure 4. And how the waveforms are separated at figure 5.

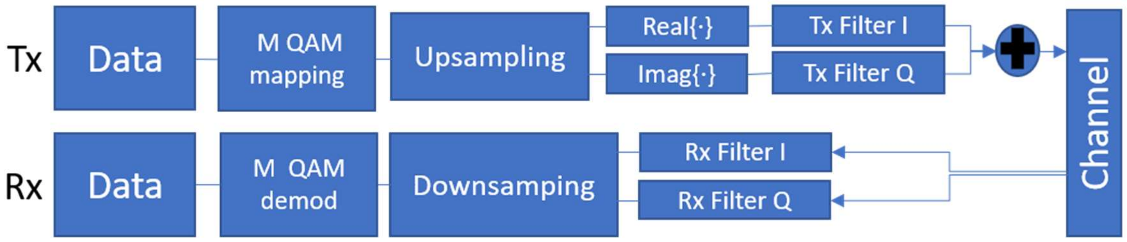


Figure 4 Principle of operation of a CAP system

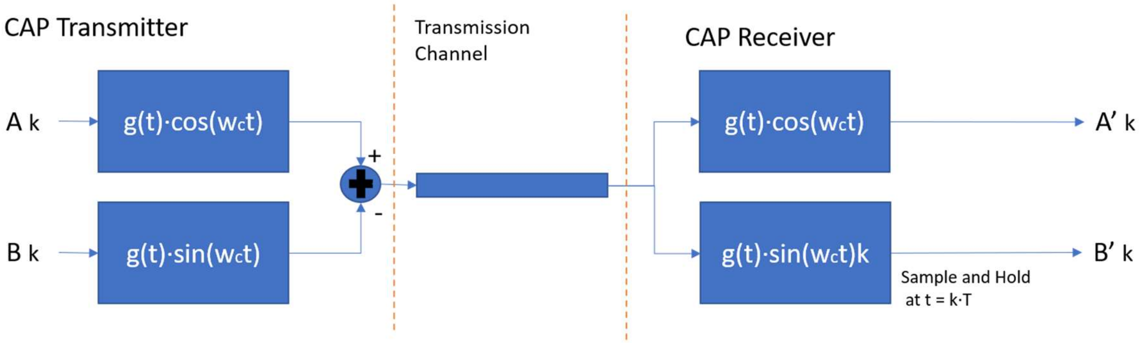


Figure 5 CAP orthogonal filters and how they separate the signals

In 1999 the international telecommunications union (ITU) chose the DMT scheme over CAP. DMT could increase total performance by dividing the available bandwidth into many subchannels. Even if the complexity was higher than the needed for CAP, the necessary electronics needed to make DMT were inexpensive. DMT still remains as the most widely used modulation format in most asynchronous digital subscriber lines (ADSLs).

CAP is still being investigated for short range optical data links. Its ability to use analogic filters to generate the CAP signal is one of its most attractive features, allowing for low power consumption. However, a practical implementation of wide-band analogic filters with linear and orthogonal phase response is very challenging.

That is why a multiband approach to CAP was proposed, signalling (MultiCAP), where the CAP signal is divided into subbands. This way, the advantages of CAP such as its simple implementation and the lower peak-to-average power ratio are combined with the advantages of DMT. We can see the principles of MultiCAP operation on figure 6.

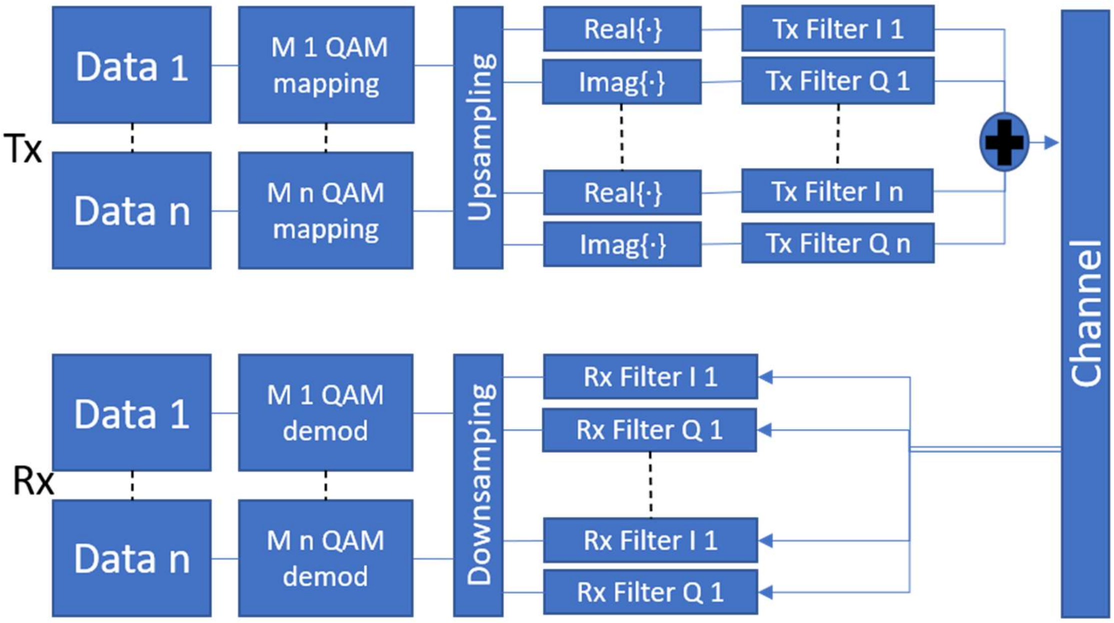


Figure 6 Principle of operation of a MultiCAP system

2.3. Non-Orthogonal Multiple Access With Multi-Band Carrierless Amplitude Phase Modulation

NOMA power multiplexing is combined with multiCAP modulation making NOMA-CAP, this way we manage to enhance the capacity and flexibility. Thanks to multiCAP, system capacity is increased, while both multiCAP and NOMA help to make a flexible and adaptative user provisioning.[1]

The combination of NOMA and multiCAP lets us make a dynamic assignment of the resources. For example, in a low-density user distribution scenario, the users can use all multiCAP bands simultaneously while their data is multiplexed by NOMA. If the number of users increases, NOMA multiplexing can be applied independently to each multiCAP band and bands can be assigned to different groups of users. This enables a flexible distribution of the RAN (Radio Access Network) capacity over many users and avoids blocking new users to a large degree. At figure 7 we can see diagram of NOMA-CAP working with multiCAP band separation and SIC receiver.

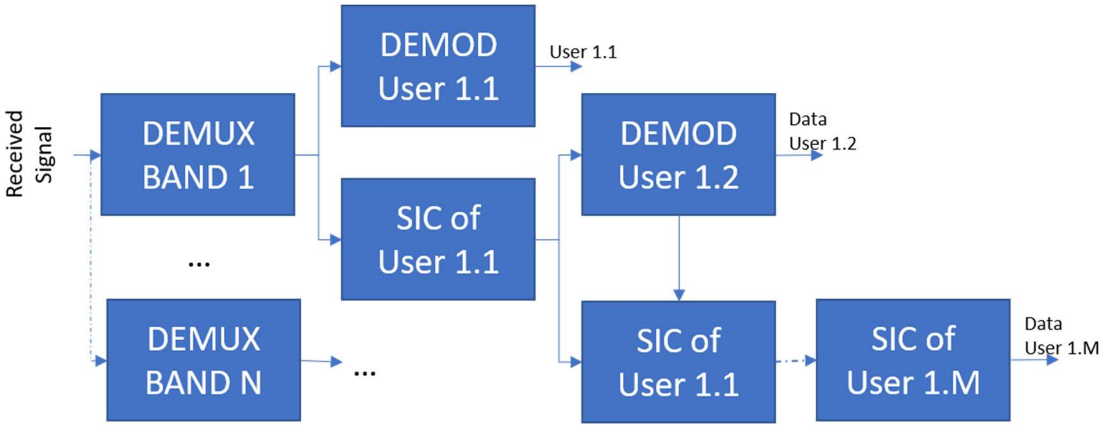


Figure 7 NOMA-CAP with multiCAP band separation

At the transmitter side, we use NOMA power multiplexing at the constellation level for each one of the multiCAP bands and before we generate the multiCAP signal. At figure 8 we can see the transmitter of NOMA-CAP.

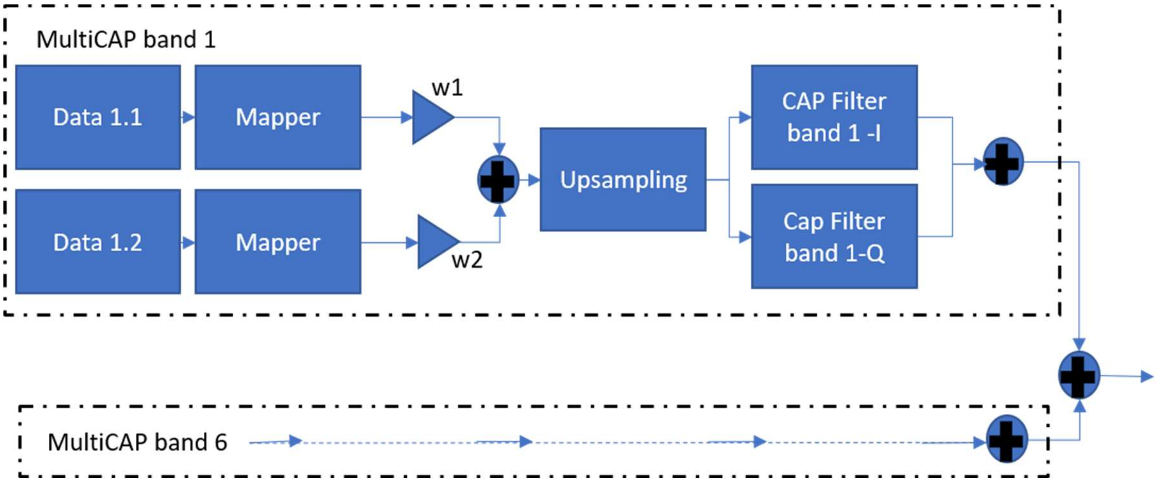


Figure 8 Transmitter block diagram for NOMA-CAP signal generation

The reception of the NOMA-CAP signal will require the extraction of each multiCAP band using its own filters and then we use successive interference cancellation to extract the different signals. If we have multiple NOMA users, we will repeat the SIC process after multiCAP demultiplexing until the signal of interest is demodulated. On figure 9 we can see a NOMA-CAP receiver

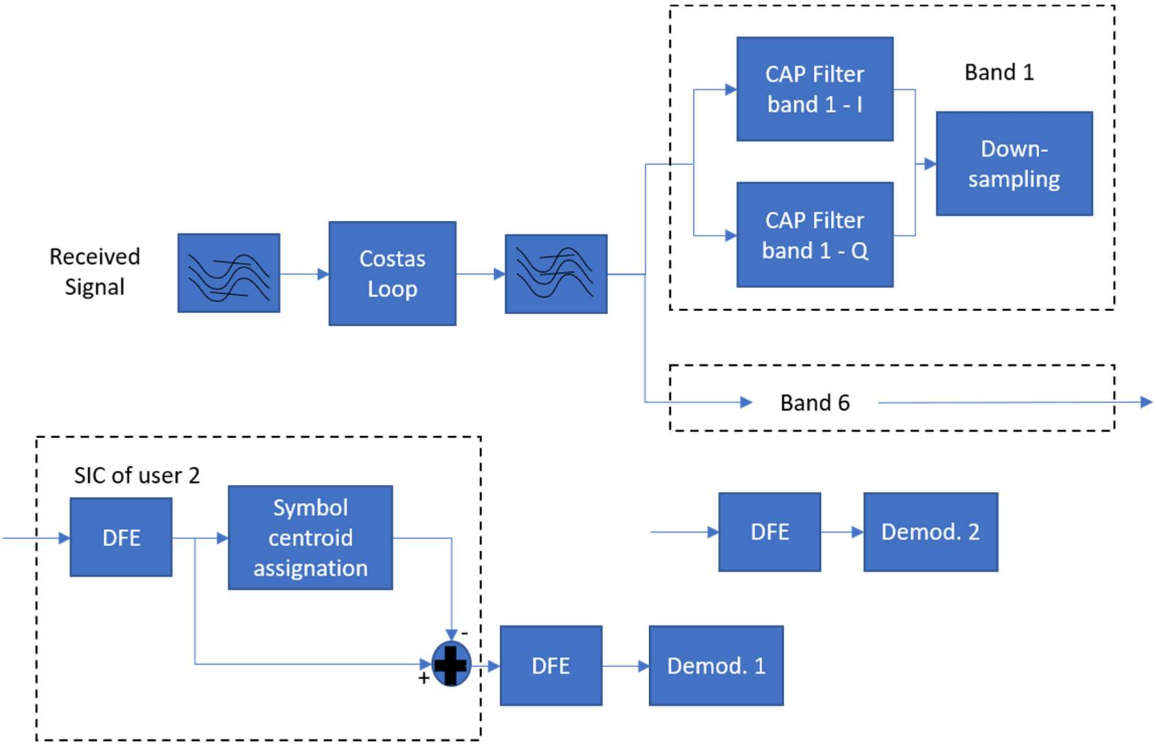


Figure 9 Receiver block diagram for NOMA-CAP signal reception

3. Methodology / project development:

3.1. Introduction to the project

My project goal was to develop an application that would let the user to freely simulate 5G NOMA-CAP signals. The application would take from the user the different inputs and would generate the Transceiver signal appropriately.

The generated signal can then be studied thanks to the Transmitter and Receiver Code, which had to be changed too in order to fit the new Transceiver signals.

3.2. Design of the application

First of all, I had to see what the code given did and which were the parameters, that were already fixed values, that the user would want to modify in order to generate the signals.

From those we chose those that we though the user would like to have a direct control of; like the number of bands; the PRBS pattern length; the type of modulation (to choose between 4, 16 or 64 QAM); symbols per second per subchannel, the subchannel weight, the samples per symbol for the transceiver and receiver signals; the length of the signal in symbols both per band and dimension; different CAP filter like filter order in symbol durations and Roll-off factor for the transceiver; different NOMA parameters like the power relation between the channels, or the user weight; and the option to show or not a graph of the generated signal.

Once we chose the different inputs that the application should have, we did some sketches to map the look of the application. Our first design was fairly simple dividing just the settings of both users as seen in figure 10

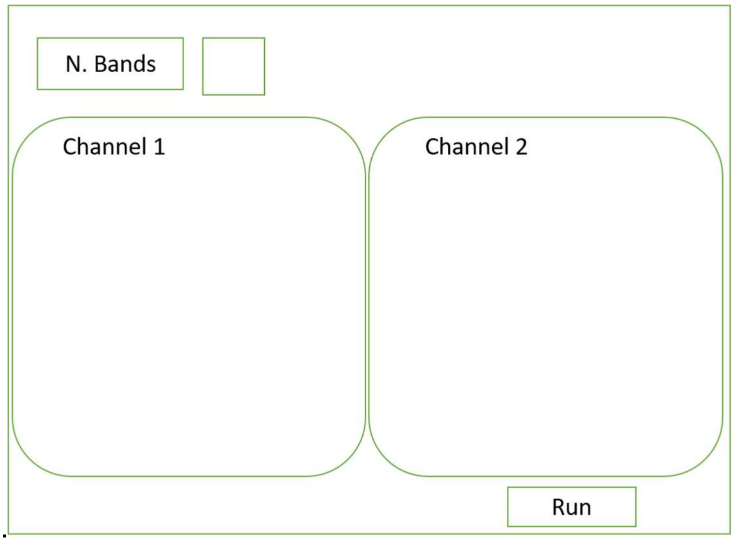


Figure 10 Preliminary design of the user interface

However, given the number of parameters, just grouping them into two boxes would make it hard for the user to follow, that is the reason we decided to group the parameters in smaller groups like PRBS, MAPPER, NOMA Parameters, multiCAP parameters, and Transceiver and Receiver parameters. In a similar manner as figure 12 shows.

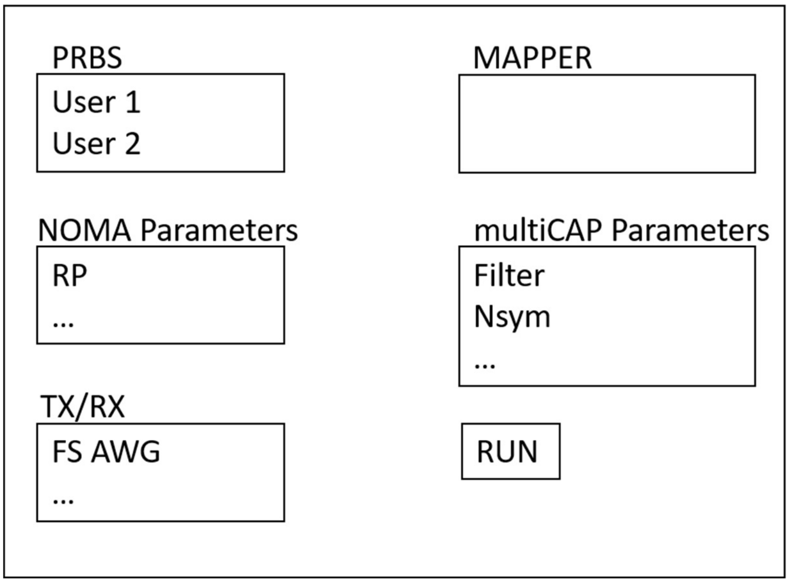
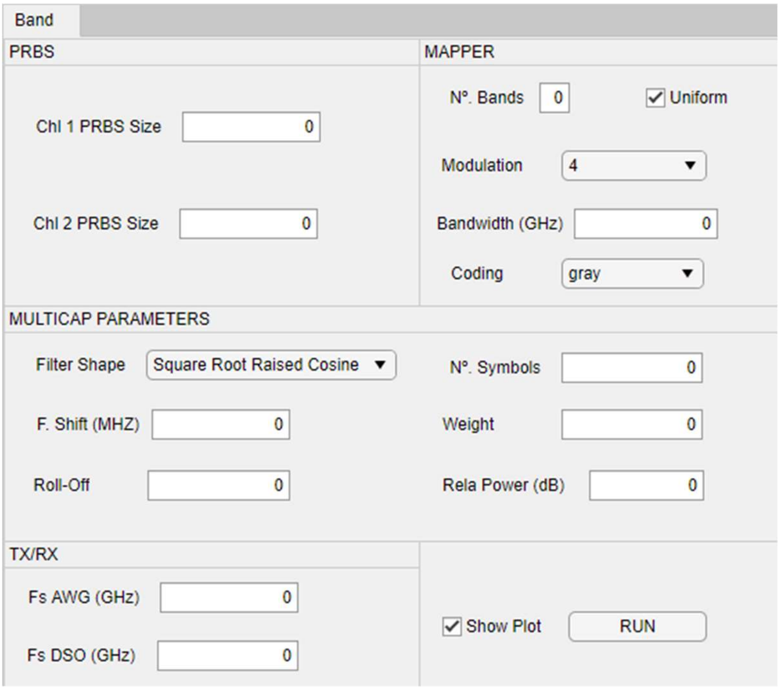


Figure 11 Improved design of the user interface

In the end the final user interface kept the idea of grouping the parameters. Although it is different it still holds some resemblance with the previous design as we can see in picture 12.



Chl 1 PRBS Size 0

Chl 2 PRBS Size 0

Nº. Bands 0 ☑ Uniform

Modulation 4 ▼

Bandwidth (GHz) 0

Coding gray ▼

Filter Shape Square Root Raised Cosine ▼

Nº. Symbols 0

F. Shift (MHZ) 0

Weight 0

Roll-Off 0

Rela Power (dB) 0

Fs AWG (GHz) 0

Fs DSO (GHz) 0

☑ Show Plot

Figure 12 User interface of the application

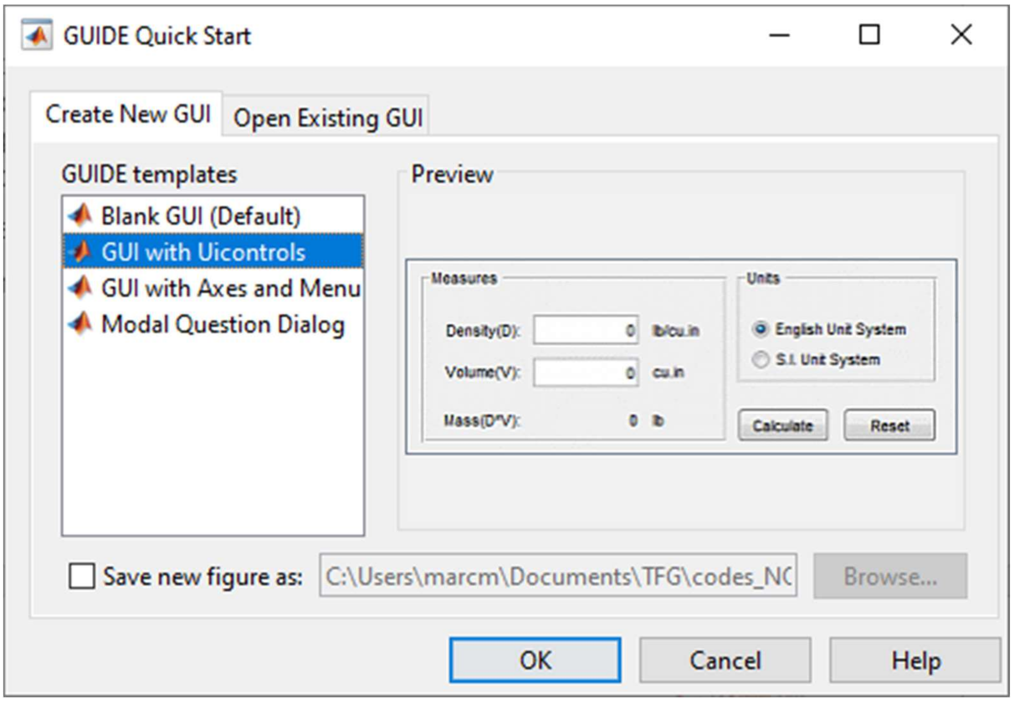
The most notable changes would be the check boxes. The “Uniform” check box lets us input the different parameters for each band individually while the “Show Plot” check box lets us choose if we want to see the generated signal representation. The main reason for these changes is that they were not planned from the beginning, but decided later as attractive options for the user. Specifically, the “uniform” check box as we will see later is one of the main focus of this project development.

3.3. Development of the application

Once the application was mapped, the next step was to actually develop the application.

MATLAB has different ways to make applications. The one I started with was GUIDE. A somewhat outdated way to develop MATLAB applications, however it was the one that the course I used to refresh my MATLAB knowledge taught.

However, GUIDE being outdated gave me some trouble in initially developing the user interface. For example, there was difficulty in aligning the different parts or switching to coding the application. In picture 13 we can see GUIDE’s user interface.



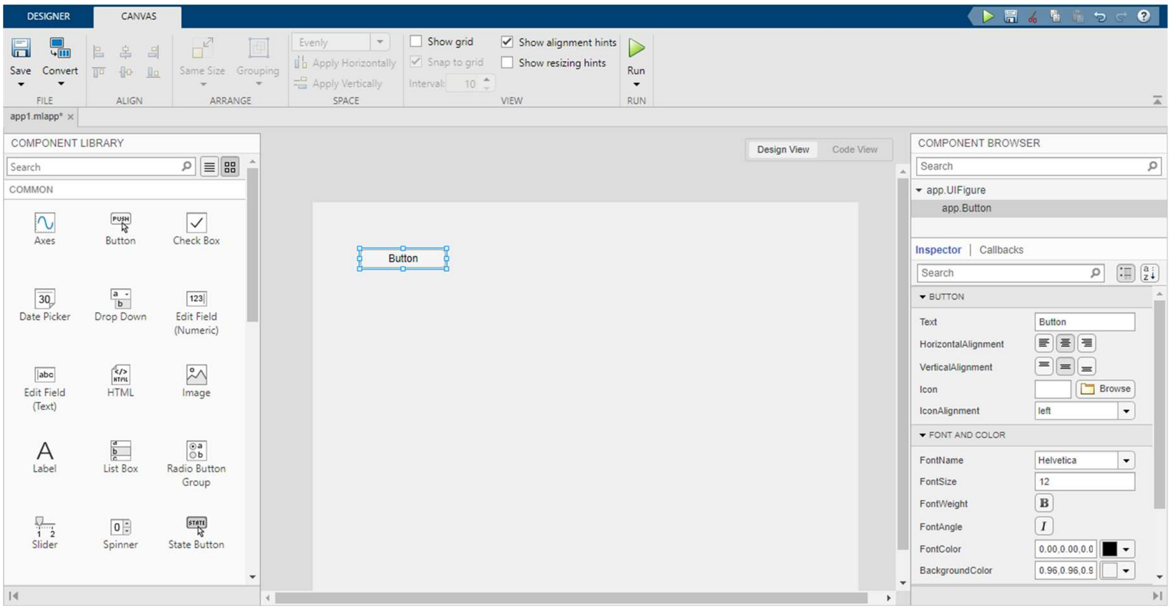
□ Save new figure as: C:\Users\marcm\Documents\TFG\codes_NC

Figure 13 GUIDE’s user interface

In the end, I decided to use the Application developer mode that MATLAB has. It is pretty much the evolution of GUIDE and includes clear quality of life improvements over GUIDE.

In picture 14 we can see MATLAB App Designer where we can easily toggle between the design view, where we design the components and look of the user interface and the code view, where the code is automatically written for the different components while the developer job is to make it so that it can communicate with a MATLAB function or write directly the full code there.

We opted for making it communicate the inputted variables to a function as the code was already written and its length made it much easier to work that way. Also, it is easier to check bugs or variables from a function than it is from the application directly.



Text: Button
HorizontalAlignment
VerticalAlignment
Icon
IconAlignment: left
FontName: Helvetica
FontSize: 12
FontWeight
FontAngle
FontColor: 0.00,0.00,0.0
BackgroundColor: 0.96,0.96,0.9

Figure 14 MATLAB App Designer

Once all the desired components are placed on the design view and we have the looks of the application, it was turn to switch to the code part.

To begin with, we had to turn the already existing Transceiver code into a function so that we can execute it every time we run the application with the variables we wish for. Most of the fixed values in the original code had to be changed into variables. We also had to set up properly the callback in the application, meaning what triggers the application to call a function, we will see that in the final application there are two callbacks. As seen on figure 12 we set a RUN button so that when pressed, it will execute the application code, take the values inputted on the user interface and execute the function transceiver function, giving us as the result the NOMA-CAP signal we designed in an executable file.

Finally, to make sure the application worked properly and that there were no errors when transmitting the variables to the transceiver function, we basically executed the application and checked the errors thus debugging the possible problems pretty much by trial and error.

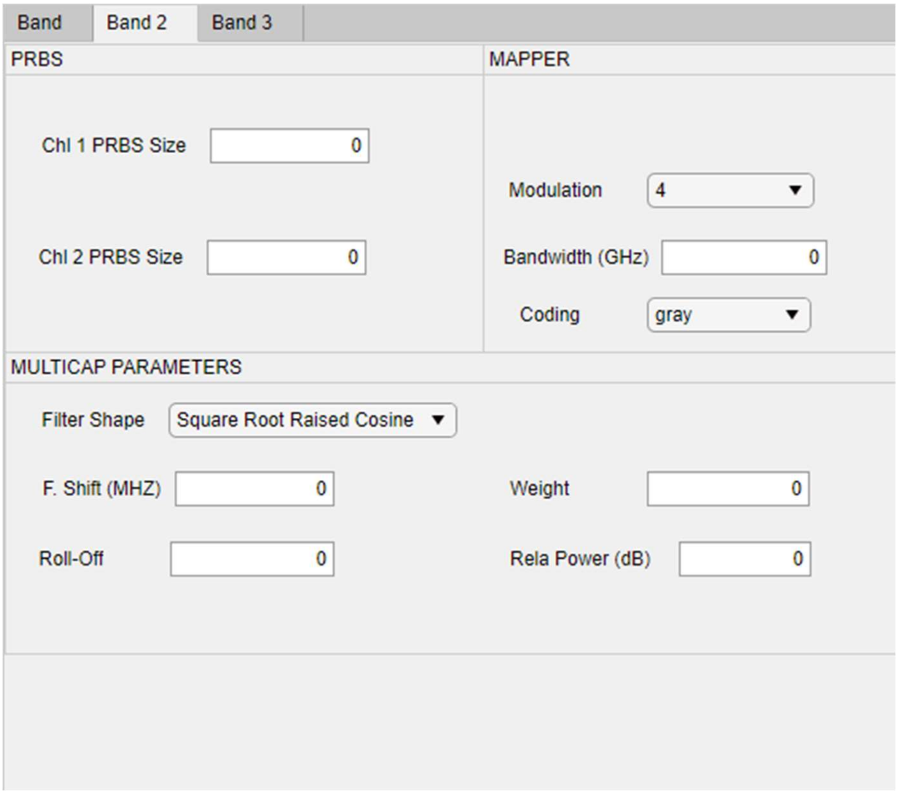
This was the initial part of the project, making a MATLAB where we could modify the values of a NOMA-CAP signal with ease, thus an application with an easy to use user interface.

As mentioned before there is a second callback, that second callback can be seen on picture 12, where there is a check box with the title “Uniform”. When this option is unchecked and there is a value inputted on “N.º Bands” different from zero it opens a new and different user interface with as many tabs as bands were written on the “N.º Bands” field.

This option exists so that the user can also simulate NOMA-CAP signals that have different settings for each band.

The first tab of the new user interface is pretty much the same as the one seen on picture 12, as on the first tab we wanted for the user to not only see the values of the individual band, but also the values that all of the bands share like the number of symbols or whether we want to see how the simulated signal looks.

When we change to a different band, we can still change the individual values of the band like the PRBS pattern length for both users, the type of modulation, the power relation between the channels, the Roll-off factor for the transceiver and the bandwidth of the signal. As seen in picture 15.



Chl 1 PRBS Size 0

Chl 2 PRBS Size 0

Modulation 4

Bandwidth (GHz) 0

Coding gray

Filter Shape Square Root Raised Cosine

F. Shift (MHZ) 0

Weight 0

Roll-Off 0

Rela Power (dB) 0

Figure 15 UI for different bands

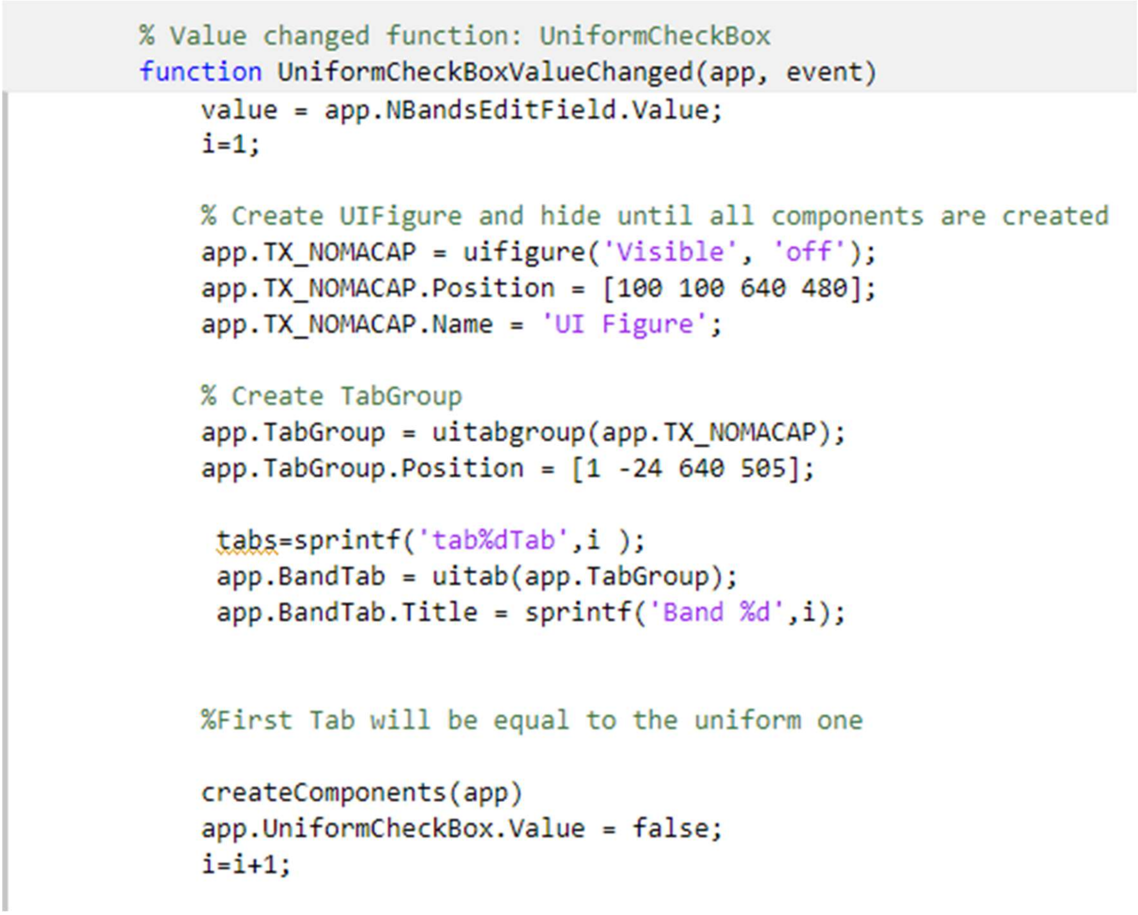
This part of the project was, personally, the most challenging one. Creating a new user interface different from the original when desired was not an easy task. Also, while MATLAB application designer has tabs as an option, these are designed in a way were a definite amount of them are created and automatically coded that way. However, what our application needed was an unknown number of tabs as a simulation may need five tabs while another may need ten tabs.

Even if there is a lot of MATLAB code shared online, after a lot of research, we could not find any instance of an already similar MATLAB application or a way to make any of the two previously mentioned points possible, so it had to be made from scratch.

We made a new simple application with only tabs and after checking its code an idea came to mind, this idea was to manually write a similar code to what it was already generated, but this time it would execute when the callback, unchecking the option, triggers.

We had to also make a way so that X tabs, X being the number inputted on the N.º Bands field would be generated, this was achieved with a simple loop, where it would execute the code to make a tab said amount of times.

As previously mentioned, the first tab maintains the same functions as the uniform bands interface does, so after creating the main user interface, we execute the function that the designer autogenerates when designing the application in the design view mode.



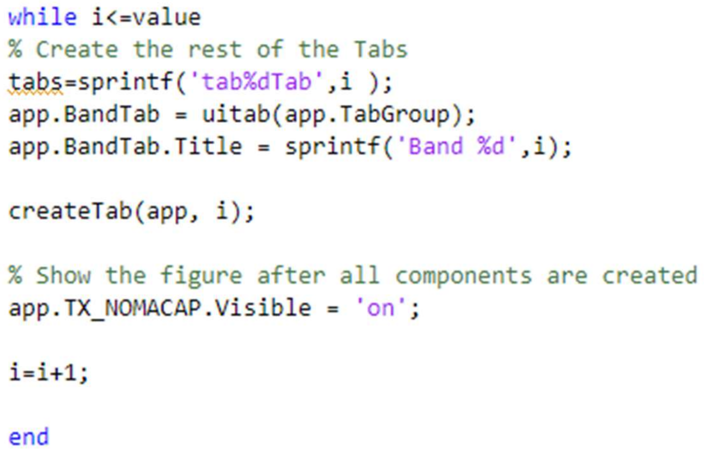
```
        % Value changed function: UniformCheckBox
        function UniformCheckBoxValueChanged(app, event)
            value = app.NBandsEditField.Value;
            i=1;

            % Create UIFigure and hide until all components are created
            app.TX_NOMACAP = uifigure('Visible', 'off');
            app.TX_NOMACAP.Position = [100 100 640 480];
            app.TX_NOMACAP.Name = 'UI Figure';

            % Create TabGroup
            app.TabGroup = uitabgroup(app.TX_NOMACAP);
            app.TabGroup.Position = [1 -24 640 505];

             tabs=sprintf('tab%dTab',i );
             app.BandTab = uitab(app.TabGroup);
             app.BandTab.Title = sprintf('Band %d',i);


            %First Tab will be equal to the uniform one

            createComponents(app)
            app.UniformCheckBox.Value = false;
            i=i+1;
```

Figure 16 Code for the first tab of the irregular mode

In picture 16 we can see the code that first creates a user interface, then generates the first tab, called Band 1, as it is identical to the one where all the bands are equal, we execute next the “createComponents” function, which is the automatically created code when the application is designed within the design view mode. We make use that it is a function so it can be called whenever we desire for.

On figure 17 we have the continuation of the user interface creation code. After what we see on figure 16, we make a loop that will iterate as many times as tabs were inputted minus one, the first tab, inside the loop we call the function “createTab” that will generate the different tabs with the variables we need the user to input into the user interface.

The function works in a similar way as the autogenerated code does, as in it copies the code it would autogenerate when making an application that from the beginning had different tabs.



```
while i<=value
% Create the rest of the Tabs
tabs=sprintf('tab%dTab',i );
app.BandTab = uitab(app.TabGroup);
app.BandTab.Title = sprintf('Band %d',i);

createTab(app, i);

% Show the figure after all components are created
app.TX_NOMACAP.Visible = 'on';

i=i+1;

end
```

Figure 17 The while loop calls the tab generating function

Once the code for the user interface was finally written, a new problem arose. The application would only save the values of the last generated tab variables, ignoring the ones on the first ones. We had to rework then the code for the “RUN” button callback function and the code for the “createTab” function.

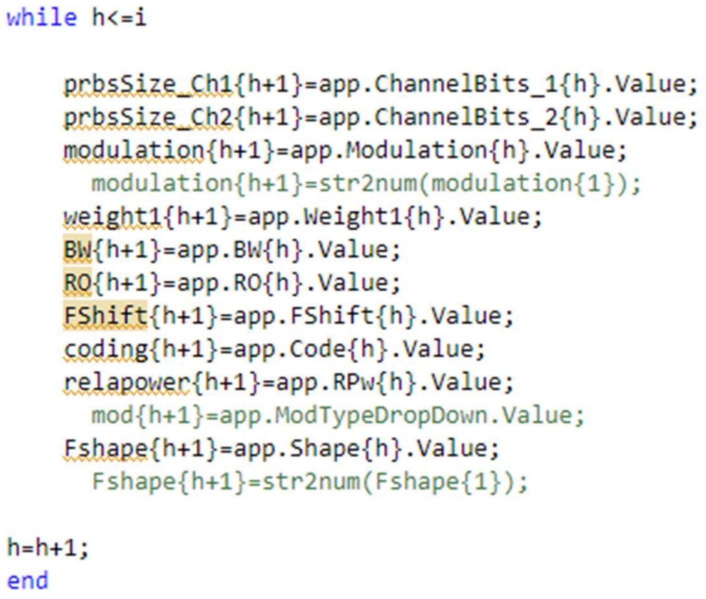
The solution to this problem was to turn the variables that are specific to every tab into an array of cells. We had to make sure that when each tab is generated, we assign a different name to each variable. In a similar way we name the tabs when we generate them, we use the same number to name the variables. This way we make sure that we can name all the variables with a different name no matter how many tabs there are.

In order to send the inputted information of all the tabs to the Transceiver function in a simple way, we can not just send them all with its own different name as that would make it impossible to work, we have to group each set of inputs into a single variable, we have just had to make save each variable into a different array of cells.

We have to make now another loop, just like the different tabs are created. The information of the first Band information is stored into the array on the first position before the loop starts, as it uses the code and variables that are automatically generated, and those are not numbered like the ones from the rest of the tabs as they do not need to be differentiated.

Then we save the rest of bands’ inputted information into the following positions of the different cell arrays and iterate through each band to store the values. The loop has to iterate both the position on the cell array and the name of the variable. The array position and the variable name number do not coincide, being the array position one higher than the variable name number, as it is just a simple addition of one, we just need to increase each loop and a single variable, h, which refers to the tab number variable. The position in the array will just be h+1.

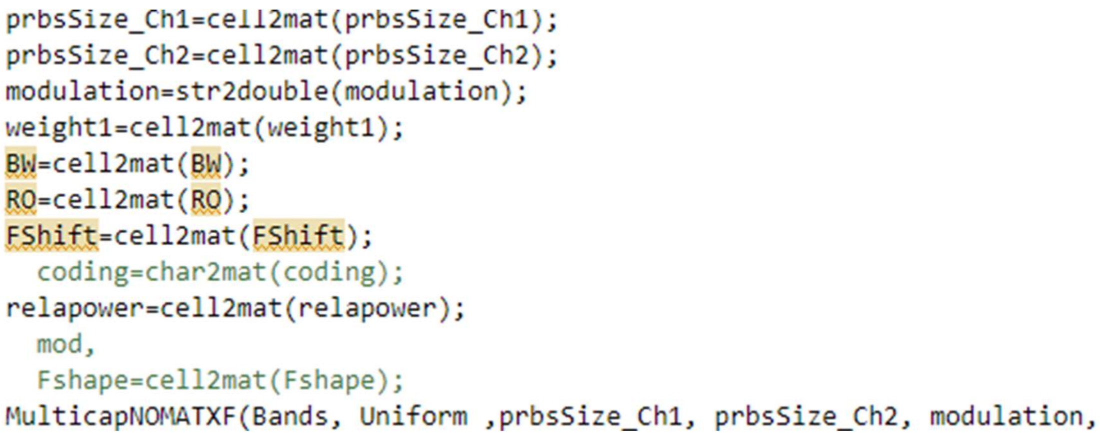
We can see the code of the while loop where the inputted information is stored into the arrays in picture 18. We can see how the value of each variable, app.VaribleName{h}.Value for example is stored into a new variable in the position h+1.



```
while h<=i

    prbsSize_Ch1{h+1}=app.ChannelBits_1{h}.Value;
    prbsSize_Ch2{h+1}=app.ChannelBits_2{h}.Value;
    modulation{h+1}=app.Modulation{h}.Value;
      modulation{h+1}=str2num(modulation{1});
    weight1{h+1}=app.Weight1{h}.Value;
    BW{h+1}=app.BW{h}.Value;
    RO{h+1}=app.RO{h}.Value;
    FShift{h+1}=app.FShift{h}.Value;
    coding{h+1}=app.Code{h}.Value;
    relapower{h+1}=app.RPw{h}.Value;
      mod{h+1}=app.ModTypeDropDown.Value;
    Fshape{h+1}=app.Shape{h}.Value;
      Fshape{h+1}=str2num(Fshape{1});

h=h+1;
end
```

Figure 18 The while loop stores the information into an array of cells

In order for the Transceiver function to use those variables, we had to turn them from array to matrixes, as seen on the code of picture 19, otherwise they would not be compatible with the requirements of functions that are in the Transceiver functions.



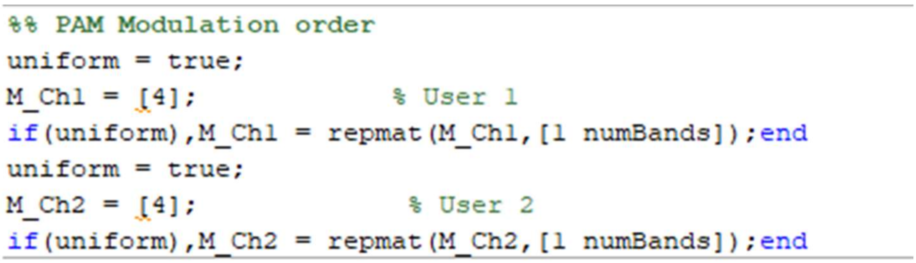
```
prbsSize_Ch1=cell2mat(prbsSize_Ch1);
prbsSize_Ch2=cell2mat(prbsSize_Ch2);
modulation=str2double(modulation);
weight1=cell2mat(weight1);
BW=cell2mat(BW);
RO=cell2mat(RO);
FShift=cell2mat(FShift);
  coding=char2mat(coding);
relapower=cell2mat(relapower);
  mod,
  Fshape=cell2mat(Fshape);
MulticapNOMATXF(Bands, Uniform ,prbsSize_Ch1, prbsSize_Ch2, modulation,
```

Figure 19 Converting the array of cells into matrixes

The last step to develop the application was to change the Transceiver code, this time so that it is able to work with bands with different settings.

The original code creator probably had an idea to implement a way to make different bands, but it was not coded.

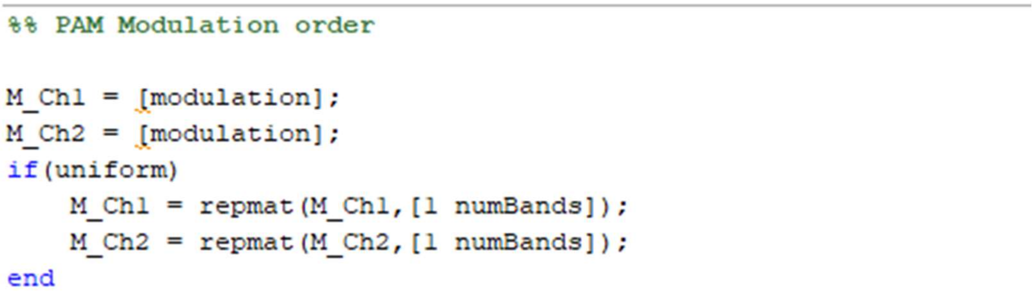
As seen in picture 20, there is an if function where it checks out if uniform is true, however it is always true as its written that way



```
%% PAM Modulation order
uniform = true;
M_Ch1 = [4];                % User 1
if(uniform),M_Ch1 = repmat(M_Ch1,[1 numBands]);end
uniform = true;
M_Ch2 = [4];                 % User 2
if(uniform),M_Ch2 = repmat(M_Ch2,[1 numBands]);end
```

Figure 20 Original code idea for non-uniform bands

On the application, if the uniform check box is unchecked, the uniform value will then be false. We had to make changes to the code so that when its false it works as it should. Most changes were as easy as just using the stored variables, as seen in picture 21. If the bands are uniform, the values for each band is the same so we need to repeat the value -repmat- as many times as bands are. If the band are not uniform, the values for each band are the ones we input on the application so we just need to refer to those values.

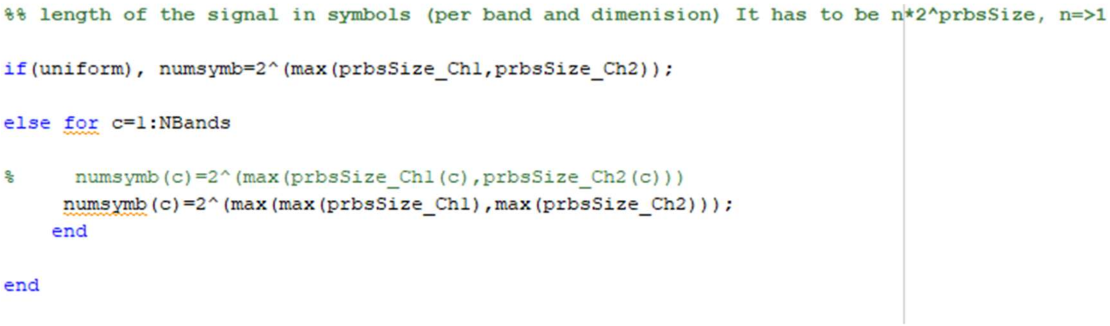


```
%% PAM Modulation order

M_Ch1 = [modulation];
M_Ch2 = [modulation];
if(uniform)
    M_Ch1 = repmat(M_Ch1,[1 numBands]);
    M_Ch2 = repmat(M_Ch2,[1 numBands]);
end
```

Figure 21 Modification so it can work with non-identical bands

However, in order to generate the PRBS signals, we need that all of the signals generated by the same simulated user have the same length.

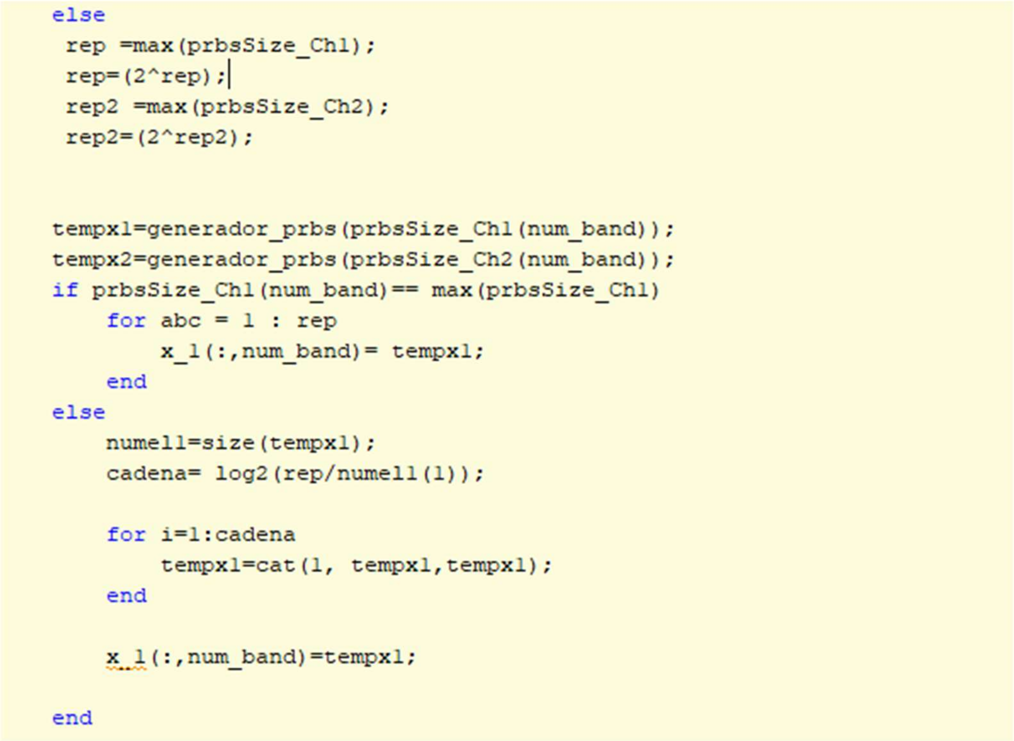
PRBS signals are powers of two, as seen in picture 22 where we store the length of the longest PRBS. This time we have to first check the longest one from each simulated user and then find the longest between those 2.



```
%% length of the signal in symbols (per band and dimenision) It has to be n*2^prbsSize, n=>1

if(uniform), numsymb=2^(max(prbsSize_Ch1,prbsSize_Ch2));

else for c=1:NBands

%      numsymb(c)=2^(max(prbsSize_Ch1(c),prbsSize_Ch2(c)))
      numsymb(c)=2^(max(max(prbsSize_Ch1),max(prbsSize_Ch2)));
     end

end
```

Figure 22 PRBS signal length

What we had to do was find the highest power from the different bands and then make each signal concatenate which itself as many times needed until it is as long as the longest one. Picture 23 has the code that shows how the PRBS generator for irregular bands work.



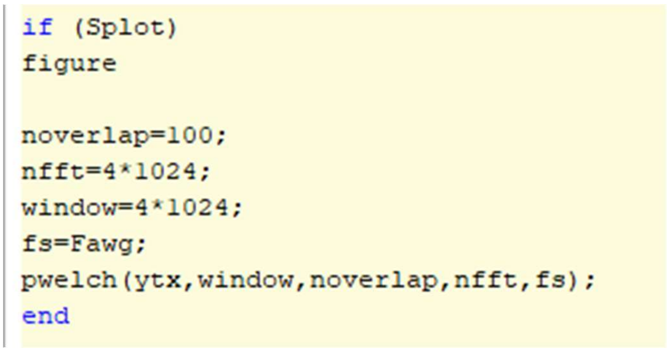
```
else
 rep =max(prbsSize_Ch1);
 rep=(2^rep);
 rep2 =max(prbsSize_Ch2);
 rep2=(2^rep2);


tempx1=generador_prbs(prbsSize_Ch1(num_band));
tempx2=generador_prbs(prbsSize_Ch2(num_band));
if prbsSize_Ch1(num_band)== max(prbsSize_Ch1)
    for abc = 1 : rep
        x_1(:,num_band)= tempx1;
    end
else
    numel1=size(tempx1);
    cadena= log2(rep/numel1(1));

    for i=1:cadena
        tempx1=cat(1, tempx1,tempx1);
    end

    x_1(:,num_band)=tempx1;

end
```

Figure 23 PRBS Generator code for non-regular bands

As seen on the code, the first step is to check which is the highest value from the different bands. Then we generate a temporal variable.

If the signal is already the longest, the temporal value is then the definitive for said band. If it is not, we need to check we have to make it as long as the longest. We check how many times the signal needs to concatenate with itself to be equally as long, as each concatenation can be taken as increasing by one the power of two.

The last part of the Transceiver code is the responsible for showing the representation of the simulated signal. We added an if that would show the plot or not depending on what the user chooses on the user interface. As see in figure 15, on the bottom right of the application, next to the "RUN" button there is a check box that says "Show Plot". Checking or unchecking the box will set the value to true or false respectively, thus going through the code seen in picture 24. We can see said representation in figure 25.



```
if (Splot)
figure

noverlap=100;
nfft=4*1024;
window=4*1024;
fs=Fawg;
pwelch(ytx,window,noverlap,nfft,fs);
end
```

Figure 24 Plot condition and generation

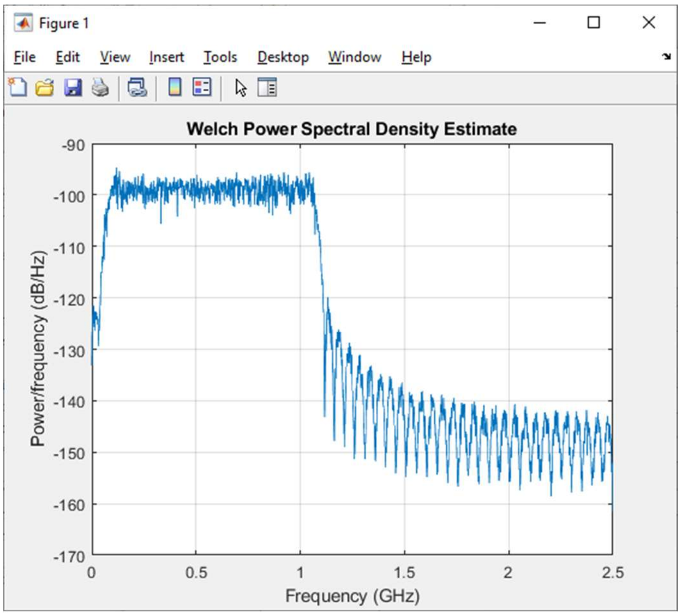
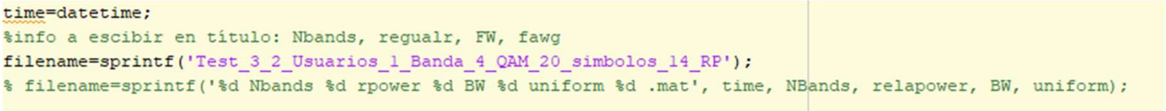


Figure 25 Show plot functionality can hide the signal representation

Not only did we work on the code for parts related to the application, but we also added to the code some tweaks to make it easier to work with.

For example, we added a way to name the files generated by the Transceiver code. These files are the signals that we generate that come out from the simulation, and will be processed by the Receiver code. We also added way to type where do you want to save the files, as if a lot are generated it can turn your MATLAB folder into a mess, thus making finding and sorting them much easier. The name applied to the files can be shown in the picture 26.



```
time=datetime;
%info a escibir en título: Nbands, regualr, FW, fawg
filename=sprintf('Test_3_2_Usuarios_1_Banda_4_QAM_20_simbolos_14_RP');
% filename=sprintf('%d Nbands %d rpower %d BW %d uniform %d .mat', time, NBands, relapower, BW, uniform);
```

Figure 26 Giving name to the created files

Now the name of the files can only be changed manually, by typing it on the filename line. The comment line below, is another option, to have the files named automatically. It takes some of the values that are inputted on the application and names the files based on those.

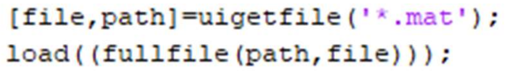
Right now, it is set so that the name of the file will include the number of bands, the power relation between users, the bandwidth, when was it created and if the bands are identical or not, however these can be changed by what the user wants to.

3.4. Modifying the receiver

The last step was to modify the receiver code so that it can properly work with all the modifications we made to the transceiver part.

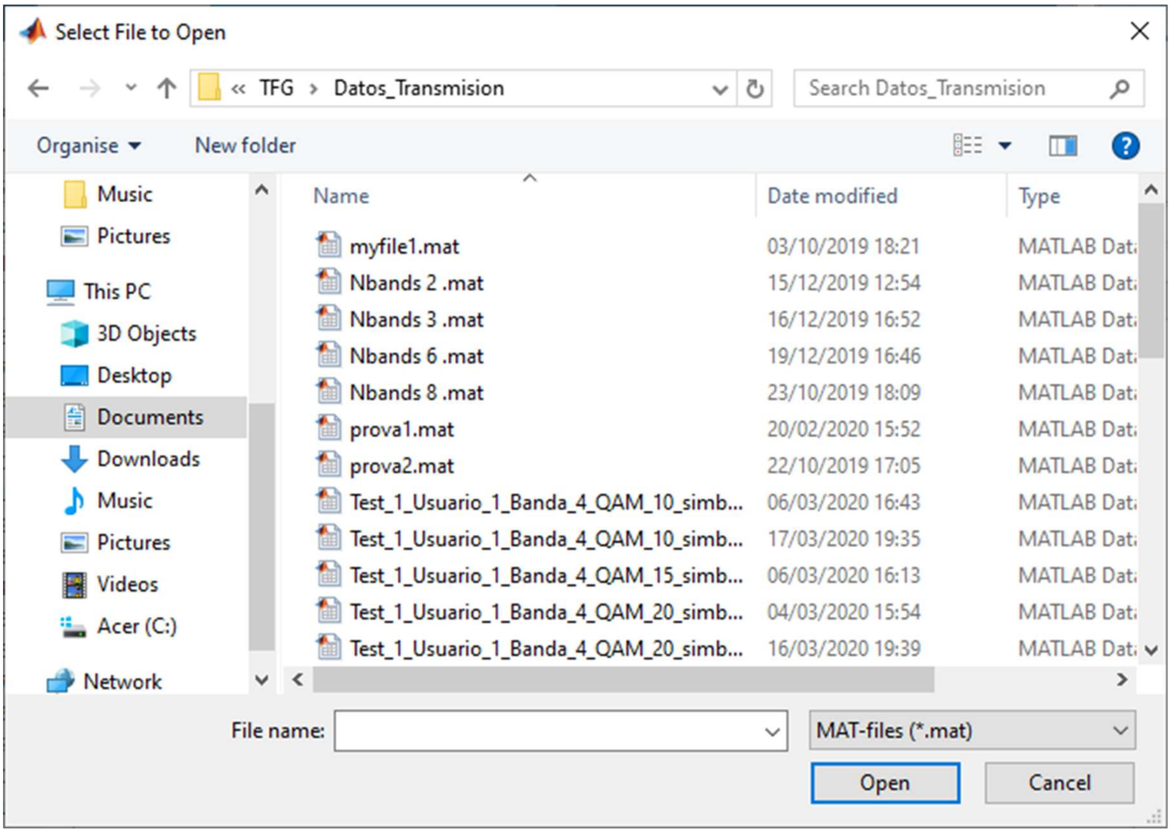
Not much had to be changed, mostly we had to concatenate the signal with itself to make it work properly with the receiver. We also added AWGN to the signal so that we could do some tests.

In order to easily choose the file that the receiver analyses, we added the “uigetfile” function, as seen on picture 27, that allows us to choose the file from the folder we want and easily navigate between folders, as seen in picture 28, instead of just copying the name of the file that must be in the same folder as the receiver code.



```
[file,path]=uigetfile('*.mat');
load((fullfile(path,file)));
```

Figure 27 The uigetfile function lets us choose from our computer folders



Name	Date modified	Type
myfile1.mat	03/10/2019 18:21	MATLAB Dat
Nbands 2 .mat	15/12/2019 12:54	MATLAB Dat
Nbands 3 .mat	16/12/2019 16:52	MATLAB Dat
Nbands 6 .mat	19/12/2019 16:46	MATLAB Dat
Nbands 8 .mat	23/10/2019 18:09	MATLAB Dat
prova1.mat	20/02/2020 15:52	MATLAB Dat
prova2.mat	22/10/2019 17:05	MATLAB Dat
Test_1_Usuario_1_Banda_4_QAM_10_simb...	06/03/2020 16:43	MATLAB Dat
Test_1_Usuario_1_Banda_4_QAM_10_simb...	17/03/2020 19:35	MATLAB Dat
Test_1_Usuario_1_Banda_4_QAM_15_simb...	06/03/2020 16:13	MATLAB Dat
Test_1_Usuario_1_Banda_4_QAM_20_simb...	04/03/2020 15:54	MATLAB Dat
Test_1_Usuario_1_Banda_4_QAM_20_simb...	16/03/2020 19:39	MATLAB Dat

Figure 28 Folder with the multiple signal files

4. Results

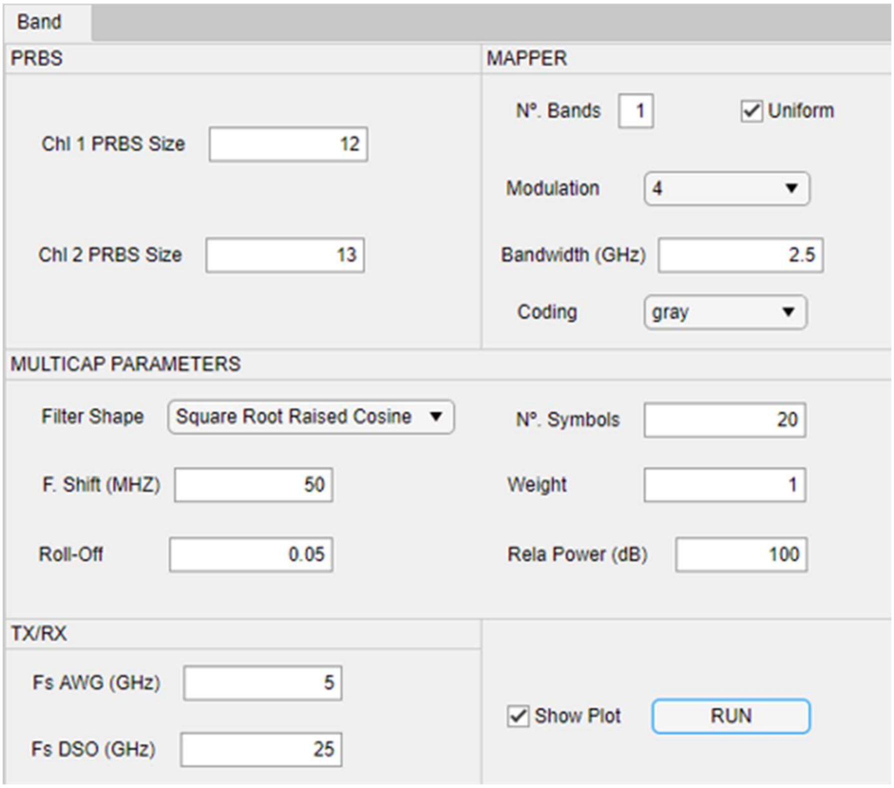
Once the development of the application was over, we tested it in different situations to check if the application and the newly additions to the code worked as expected.

4.1. First test: One user one band simulation

The first test was quite simple, it was to see if the SNR/BER values were within the expected margins.

The application recreated a one band with one user situation. In order to just have a user we had to set an extremely high relation power between both PRBS signals, making the second one ignorable.

In picture 29 we can observe the settings of the signals, we set for the experiment: one band, 4 QAM, 2,5 GHz, 20 symbols and a power relation between signals of 100.



PRBS

Chl 1 PRBS Size: 12

Chl 2 PRBS Size: 13

MAPPER

Nº. Bands: 1 ☑ Uniform

Modulation: 4

Bandwidth (GHz): 2.5

Coding: gray

MULTICAP PARAMETERS

Filter Shape: Square Root Raised Cosine

F. Shift (MHZ): 50

Roll-Off: 0.05

Nº. Symbols: 20

Weight: 1

Rela Power (dB): 100

TX/RX

Fs AWG (GHz): 5

Fs DSO (GHz): 25

☑ Show Plot

Figure 29 Configuration for the first test

As the Show Plot is checked we obtained a Welch power spectral density estimate. The previous configuration gave us the following estimate that we can observe in picture 30. The bandwidth specified on the application corresponds to that of the figure.

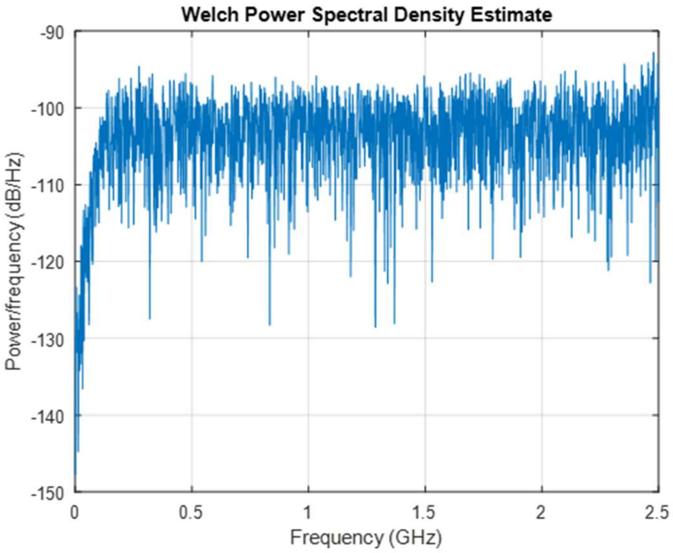


Figure 30 Welch power spectral density estimate from the first test

Then to further observe the SNR/BER relation we made two more signals with the same settings but changing the modulation to 16 and 64 QAM.

After we simulated all 3 signals, we had to analyse them with the receiver, in order to see the SNR/BER, we can see all three SNR/BER curves on figure 31. To make this curve we had to execute each signal through the receiver changing the SNR and checking what BER value we would obtain.

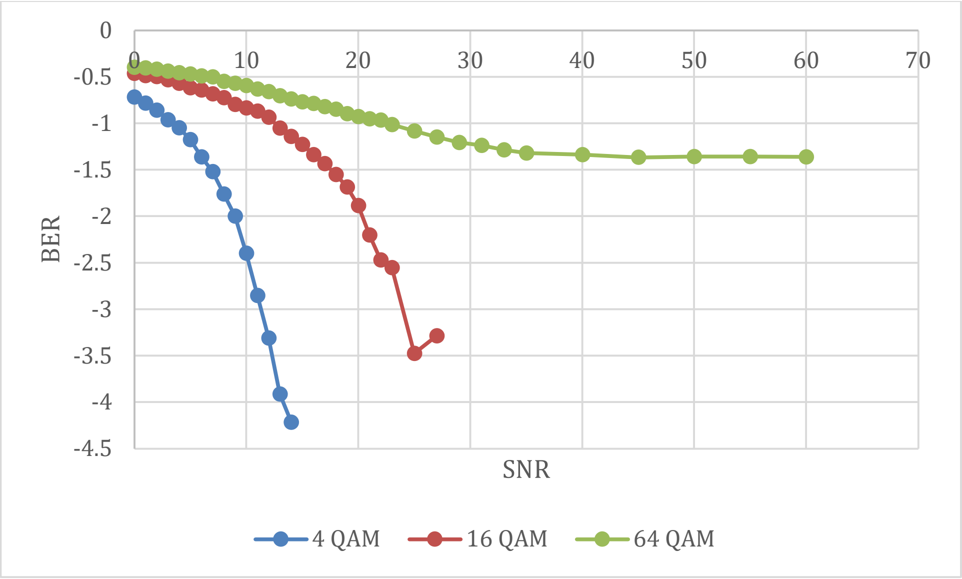


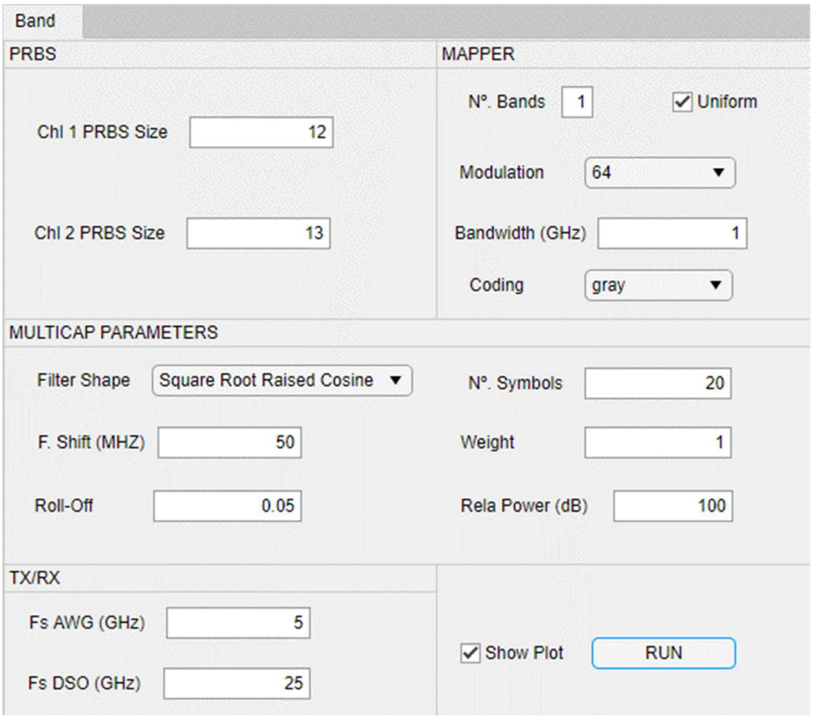
Figure 31 Comparison between the three curves

These results were not the ones we expected, especially for the 16 and 64 QAM.

After analysing the different results and thanks to the welch power spectral density estimate representation, we though that it could be due to an aliasing effect. We can observe that in in figure 30 the bandwidth occupies the whole grid.

We then chose to reduce the bandwidth of the following tests from 2,5 GHz to 1 GHz as it would let us to leave the remaining variables with the same values, figure 32.

And so, we redid the first test.



Band

PRBS

Chl 1 PRBS Size 12

Chl 2 PRBS Size 13

MAPPER

Nº. Bands 1 ☑ Uniform

Modulation 64 ▼

Bandwidth (GHz) 1

Coding gray ▼

MULTICAP PARAMETERS

Filter Shape Square Root Raised Cosine ▼

Nº. Symbols 20

F. Shift (MHZ) 50

Weight 1

Roll-Off 0.05

Rela Power (dB) 100

TX/RX

Fs AWG (GHz) 5

Fs DSO (GHz) 25

☑ Show Plot

Figure 32 Correct set up for the first test

The welch graphic also reflected the change as seen on figure 33, where the bandwidth drops at 1GHz.

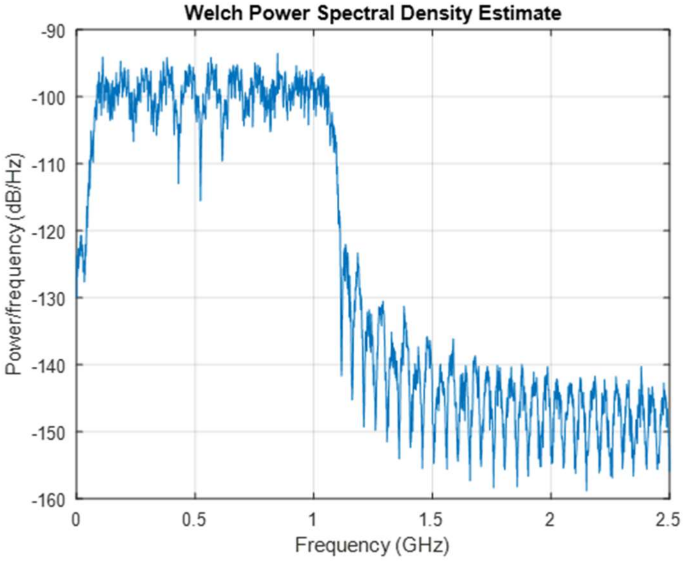


Figure 33 Welch graphic with 1GHz bandwidth

Just like the previous attempt we did the test for 4 QAM, 16 QAM and 64 QAM and obtained the curves seen on figure 34. The exact values for each of them are in the tables in the appendices.

Once the curves were done, we looked for the SNR limit on the curves, it should be the value closest to -2,5 on the BER axis. We rerun the receiver code with the SNR limit to obtain the scatter plot of each signal as seen on the on figures 35, 36, 37 respectively.

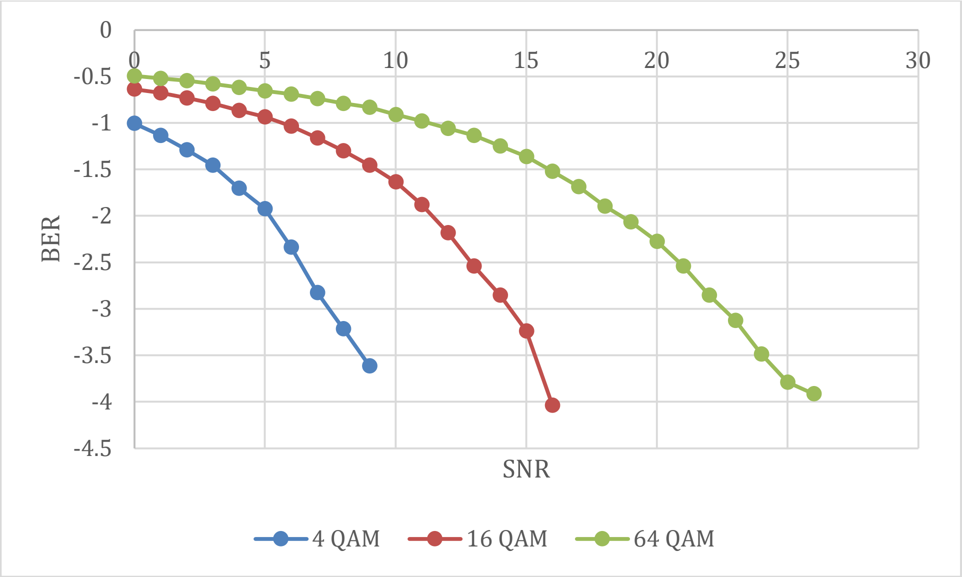


Figure 34 Comparison between the 4 QAM, 16 QAM and 64 QAM curves

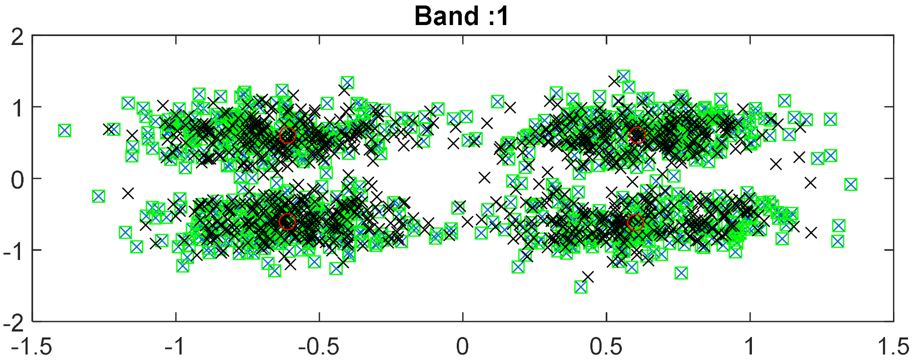


Figure 35 Scatter plot for 4 QAM

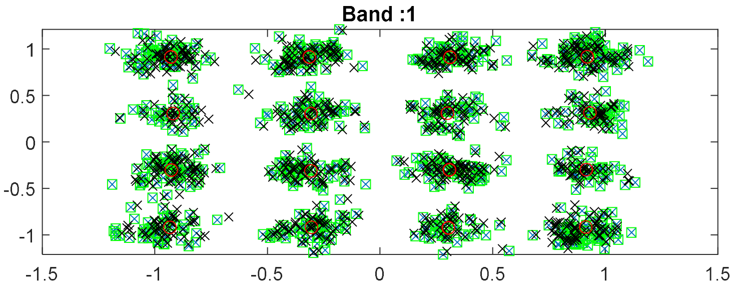


Figure 36 Scatter plot for 16 QAM

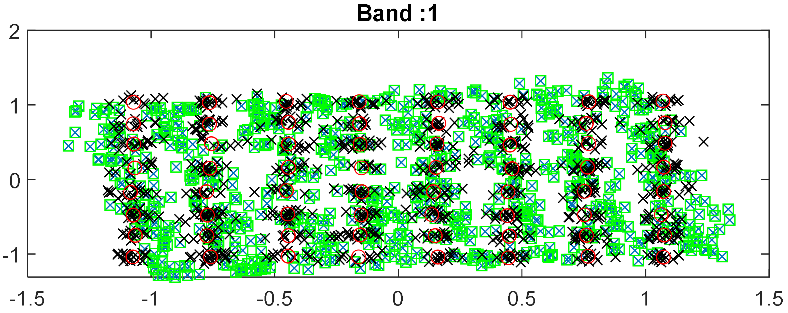


Figure 37 Scatter plot for 64 QAM

This time the results were within the expected margin. The error of the initial test helped us figure how to understand how to properly configure it.

We also repeated the tests but filter order with a duration of 10 symbols i.e., the filter spans ten symbol durations. We will just post on figure 38 the comparison chart between all three modulation values.

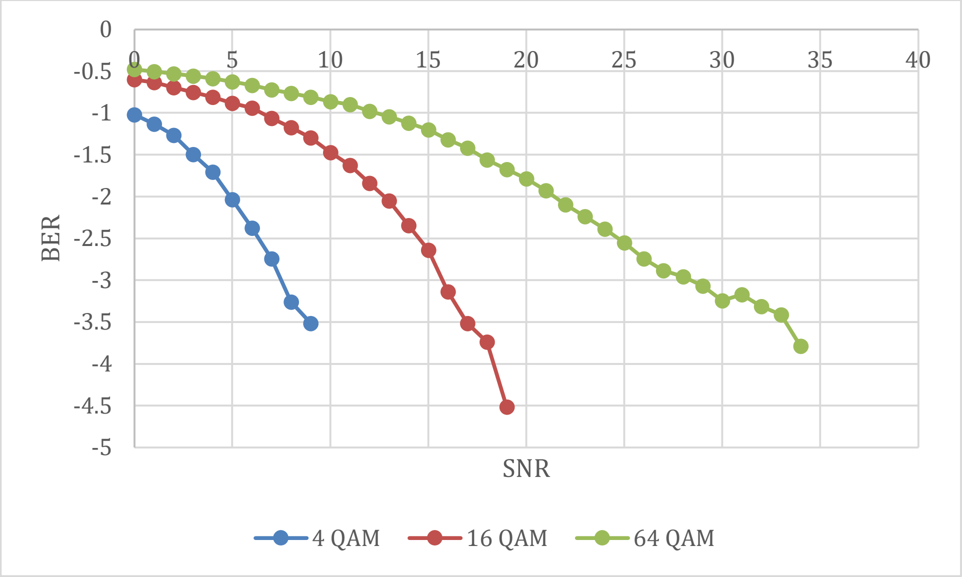


Figure 38 Comparison between the 4 QAM, 16 QAM and 64 QAM curves with 10 symbols

We then compared the SNR Limit for each modulation individually, depending on the filter order in symbol durations. We can observe the results on table 1.

Table 1 SNR needed to obtain the BER limit

	4 QAM	16 QAM	64 QAM
10 Symbols	8	13	25
20 Symbols	8	15	21

We can observe that for smaller constellations, lowering the filter order in symbol durations rate does not have much effect, as constellation symbols are separated meaning that noise is close to negligible. For bigger constellations however, it worsens, as symbols are closer to each other so noise has more effect.

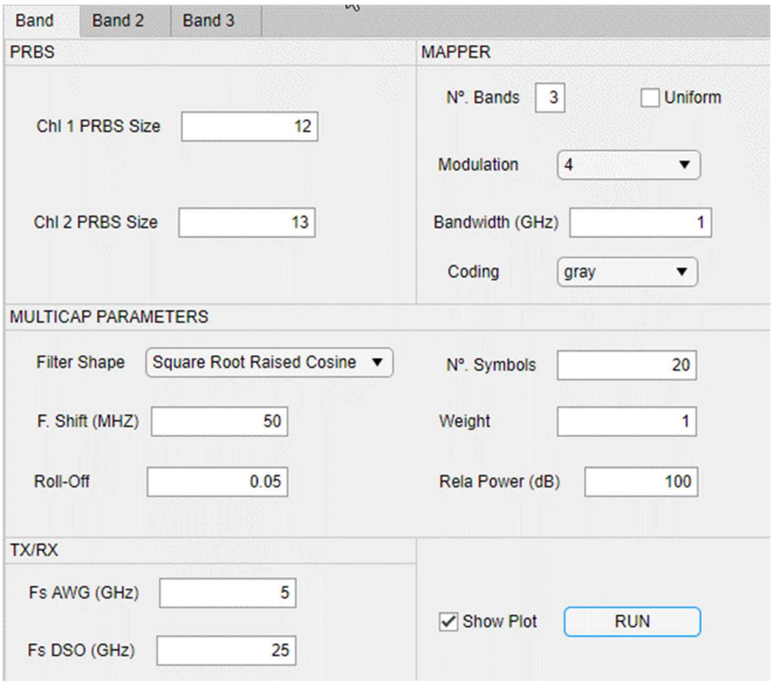
4.2. <u>Second test: One user three bands simulation</u>

The second test consisted on having three bands and a single user, each of the band had a different bandwidth and different modulation, meaning we are going to use the non-uniform bands mode. Like in the previous test, the single user is achieved with a high relation power between users.

The configuration for the test was the following, as seen in pictures 39 for the first band, 40 for the second band and 41 for the third band. This time we had to make sure that the sum of the three bands' bandwidth could not be equal or over 2,5 GHz, so we can avoid the aliasing. We will give the first band a bandwidth of 1 GHz the second and the second and third one a bandwidth of 0,5 GHz.

Also, each band will have a different modulation, the first band will have a modulation of 4 QAM, the second band will have a modulation of 16 QAM and the third Band will have a modulation of 64 QAM.

Apart from the previously mentioned modifications, the settings will be the same as on the first test.



PRBS

Chl 1 PRBS Size 12

Chl 2 PRBS Size 13

MAPPER

Nº. Bands 3 □ Uniform

Modulation 4

Bandwidth (GHz) 1

Coding gray

MULTICAP PARAMETERS

Filter Shape Square Root Raised Cosine

F. Shift (MHZ) 50

Roll-Off 0.05

Nº. Symbols 20

Weight 1

Rela Power (dB) 100

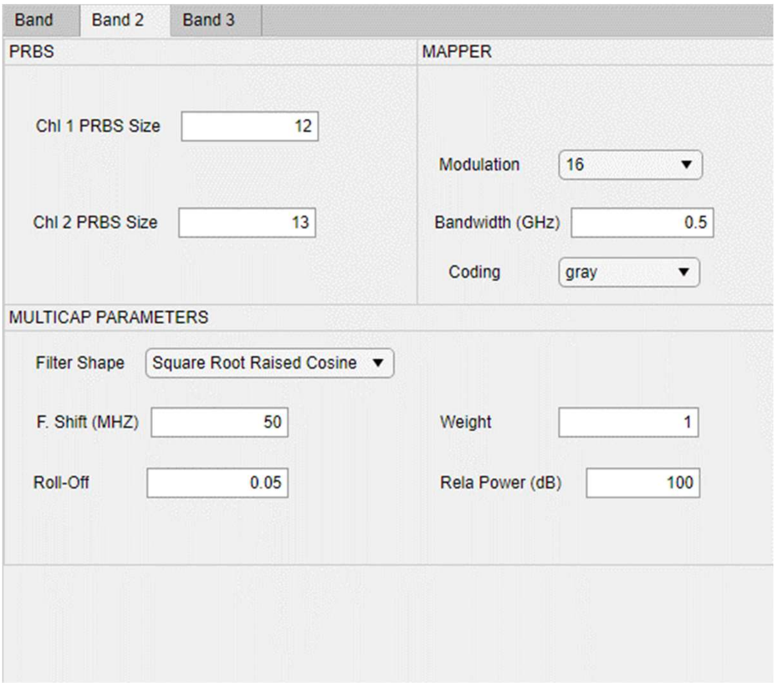
TX/RX

Fs AWG (GHz) 5

Fs DSO (GHz) 25

☑ Show Plot

Figure 39 Test 2 band 1 configuration



PRBS

Chl 1 PRBS Size 12

Chl 2 PRBS Size 13

MAPPER

Modulation 16

Bandwidth (GHz) 0.5

Coding gray

MULTICAP PARAMETERS

Filter Shape Square Root Raised Cosine

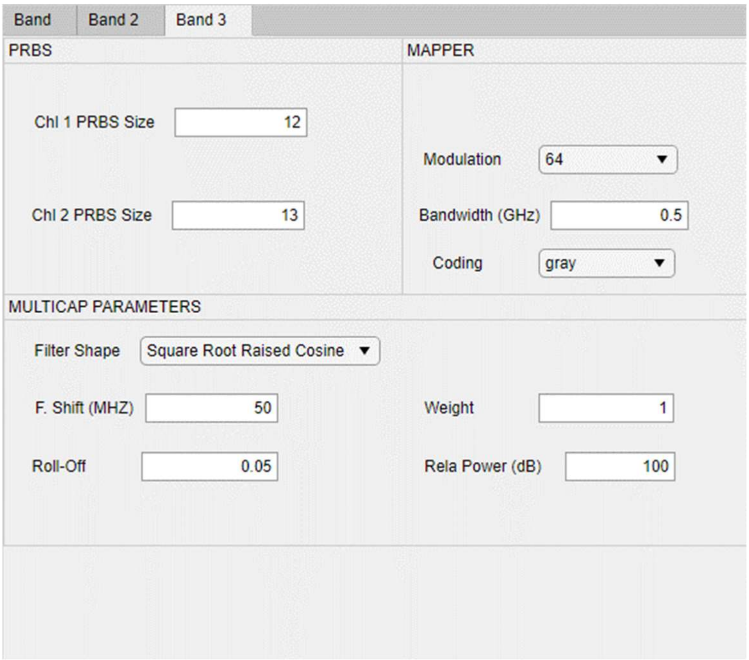
F. Shift (MHZ) 50

Roll-Off 0.05

Weight 1

Rela Power (dB) 100

Figure 40 Test 2 band 2 configuration



PRBS

Chl 1 PRBS Size 12

Chl 2 PRBS Size 13

MAPPER

Modulation 64

Bandwidth (GHz) 0.5

Coding gray

MULTICAP PARAMETERS

Filter Shape Square Root Raised Cosine

F. Shift (MHZ) 50

Roll-Off 0.05

Weight 1

Rela Power (dB) 100

Figure 41 Test 2 band 3 configuration

The sum of the bands gave us the following welch power spectral density estimate, as seen in fig 42. We can observe the three bands spectral density, the first one being twice as big as the other two, corresponding this way to the specified settings.

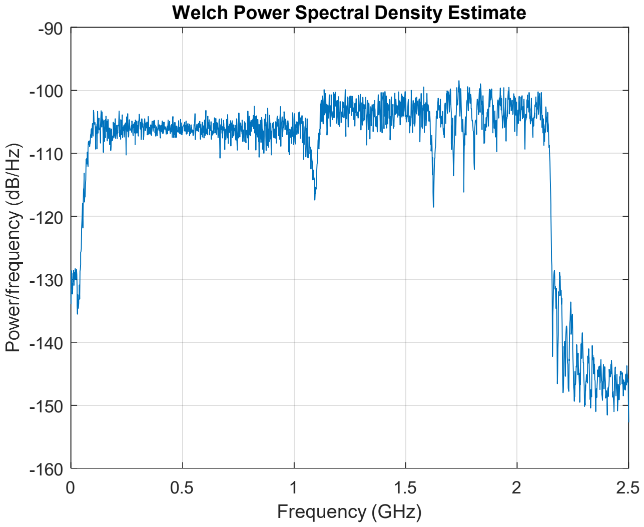


Figure 42 WPSDE of the second test

Then, like in the previous test, we ran the obtained file through the receiver multiple times with a different SNR value. For each time we obtained three different BERs, one for every band, we can observe the values on table 4. The SNR/BER curve of this test can be observed on figure 43.

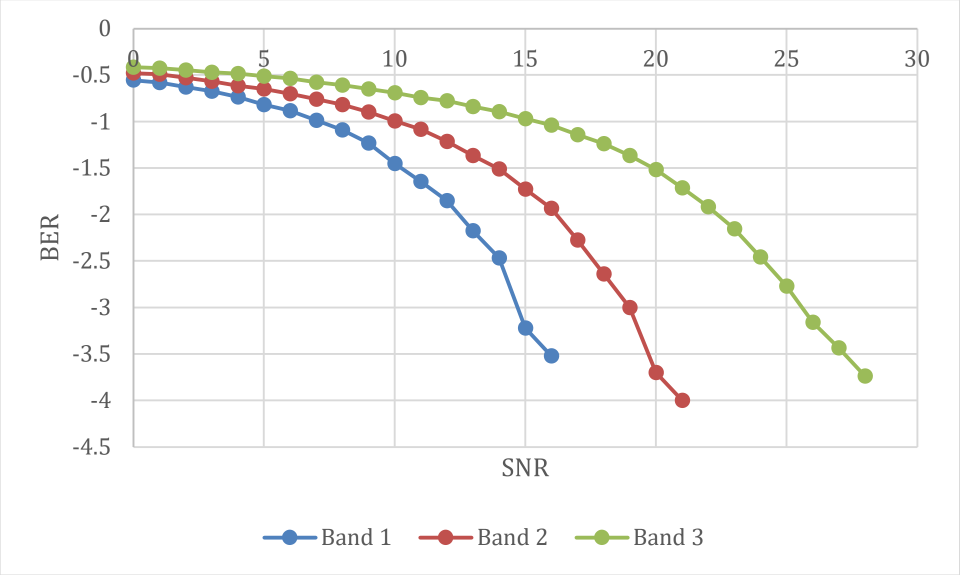


Figure 43 SNR/BER curve for the second test

Like in the first test we took the SNR value from the observing in the curve the BER limit and obtained the scatter plot at the 3 different points, for 4 QAM on picture 44, 16 QAM on picture 45 and 64 QAM on picture 46.

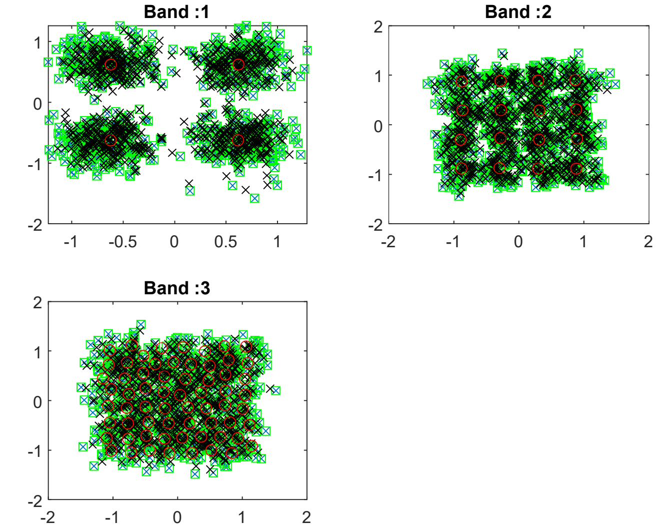


Figure 44 Scatter plot for 4 QAM

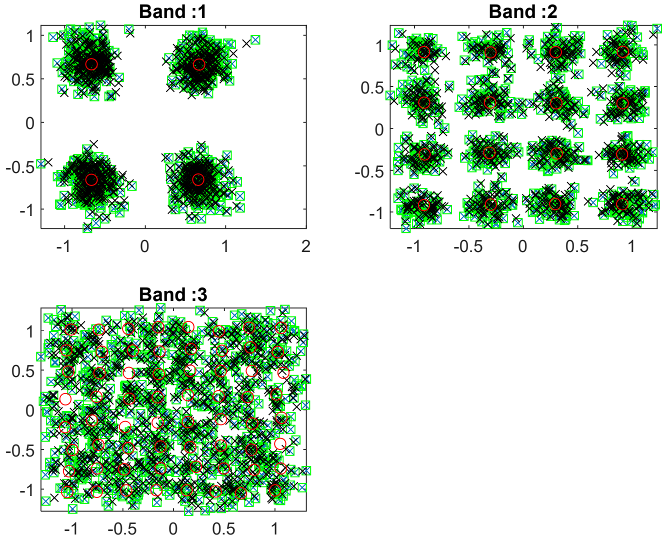


Figure 45 Scatter plot for 16 QAM

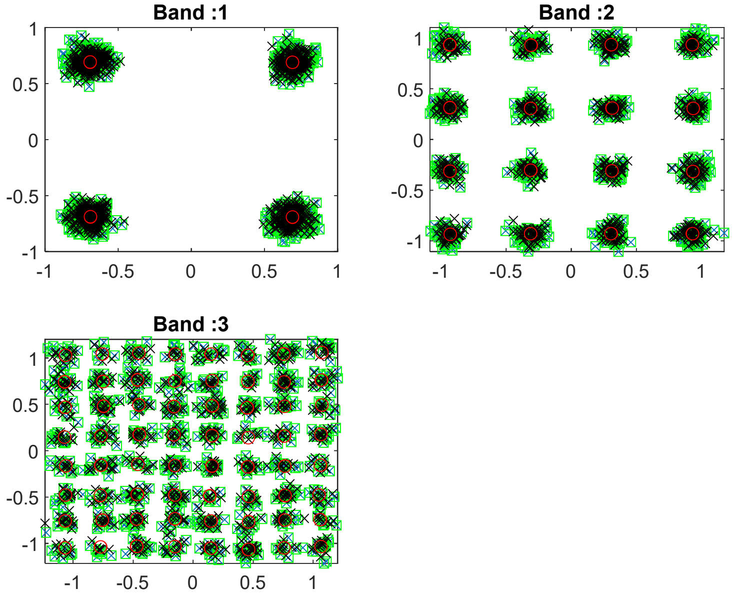


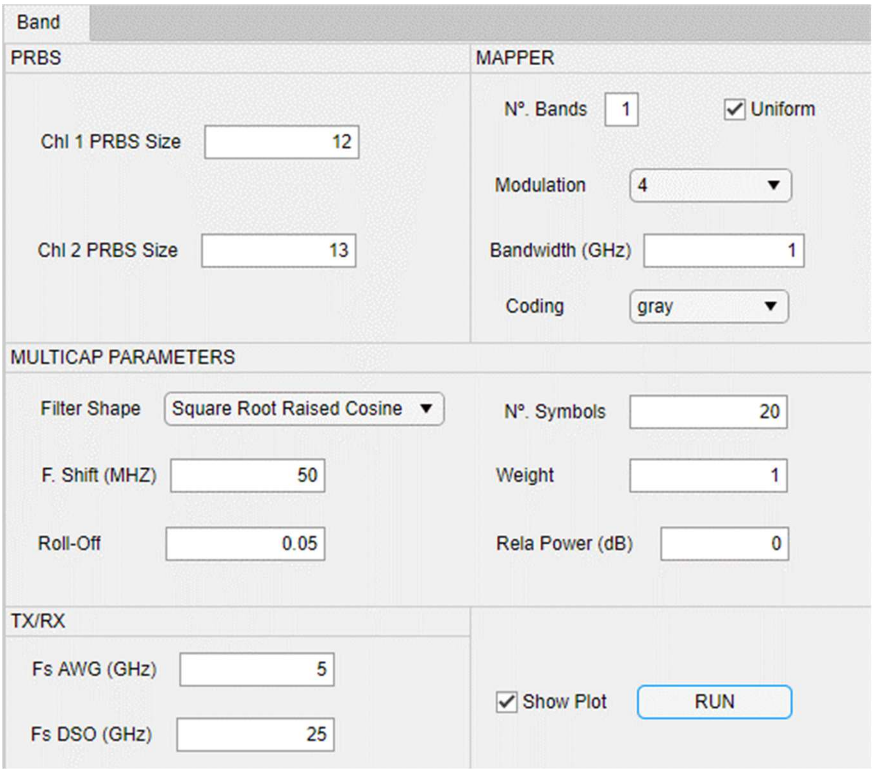
Figure 46 Scatter plot for 64 QAM

From this test we could first observe the correct functionality of the mode for irregular bands.

We can also observe the behaviour of the simultaneous signals over different modulations, and how the BER limit increases with the increase of modulation, as we expected.

4.3. Final test: One band two users and increasing relative power between them simulation

The third and final test consisted on having two users and generate different signals changing the relative power between the two of them, from 0 to 14. We generated seven signals, skipping by 2 on the relation power between each test. The settings were the ones on picture 47.



Band

PRBS

Chl 1 PRBS Size 12

Chl 2 PRBS Size 13

MAPPER

Nº. Bands 1 ☑ Uniform

Modulation 4 ▼

Bandwidth (GHz) 1

Coding gray ▼

MULTICAP PARAMETERS

Filter Shape Square Root Raised Cosine ▼

F. Shift (MHZ) 50

Roll-Off 0.05

Nº. Symbols 20

Weight 1

Rela Power (dB) 0

TX/RX

Fs AWG (GHz) 5

Fs DSO (GHz) 25

☑ Show Plot

Figure 47 Initial settings for the third test

After that I had to, like in the previous tests, increase the SNR on the receiver to see the different BER values. This time however, we had BER for both the weak and strong users. The values of the tests can be seen on the tables in the appendices.

The objective was to see the relation between relative power, the SNR and the BER. The results were the following for the Strong user, figure 48, and for the weak user, figure 49.

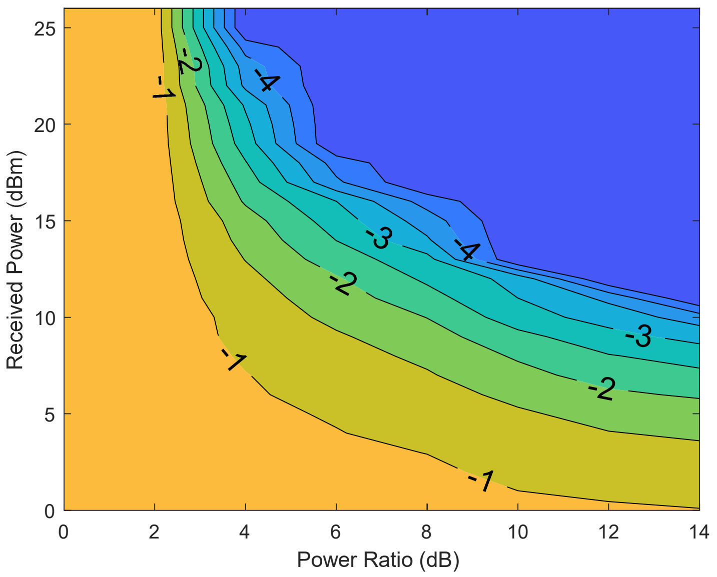


Figure 48 Relation between Power, SNR and BER for the Strong user

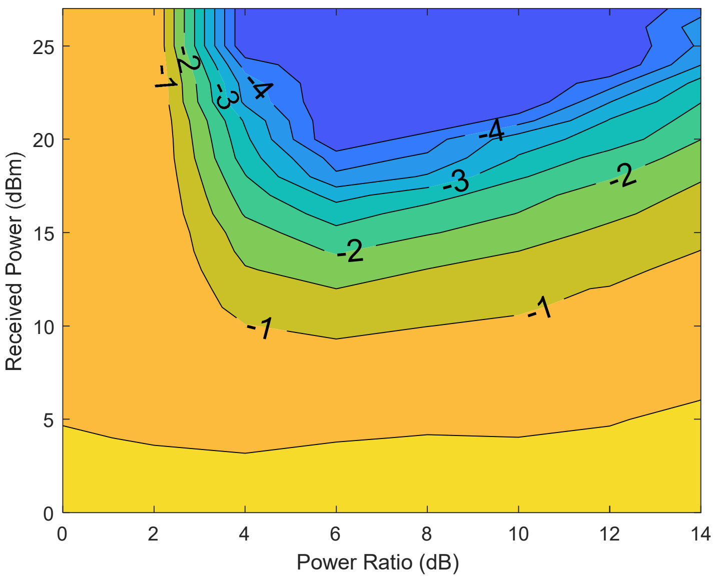


Figure 49 Relation between Power, SNR and BER for the Strong user

On figure 48 we can observe that, for the strong user, the higher the relative power is, the better. The strong user considers the weak as noise, that is why the more ignorable the weak one is better. The first and second tests were built upon this fact.

On figure 49 we can observe that for the weak user its better a medium relative power between both signals. With both high and weak relative power, the signal gets distorted with noise as the SIC is not ideal

Finally, we compared the SNR/BER relation of both users with a relative power of 8 and we compared to that of a single user taken on the first test with 16 QAM and the same number of symbols.

The reason as why we chose 8 is that it was the average value from the taken measures, also the normal value should be 7, so it was close to it. We can see the comparison on figure 50.

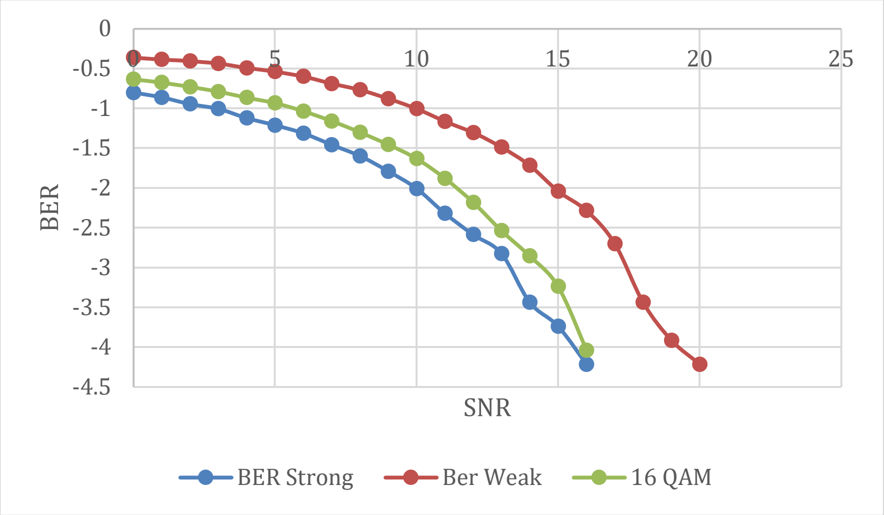


Figure 50 Comparison between one user at 16 QAM and two users at 4 QAM

In conclusion we can say that the signal sent with one user is better than with two users.

However, the flexibility and benefits of being able to send two signals at once far outweigh the penalties of having just one user

With these three tests we could observe the correct functionality of the application as well as take some experimental results over different situations.

5. Budget

The realization of this project was fairly simple when counting costs:

- Acer Aspire 3 a315-41-r8zc - 500€
- MATLAB license – 250€ / year
- Junior engineer - 3750€

The total cost sums up to 4500 €.

For the Laptop, the application can be developed and executed, probably, by cheaper computers.

The salary for junior engineers is 21.948 € [19] per year which would translate to approximately the amount specified on the costs, taking into account it is a salary for nine daily working hours.

While the cost of the application development may seem initially high, this application will allow to future testers of NOMA-CAP to save time when making signal simulations which will translate in the end to saved money.

6. Conclusions and future development:

With the increasing demand over high speed fixed and wireless networks, NOMA-CAP appears to be a solution that promises to increase 5G networks efficiency.

The objective of this project was to create an application that would let users simulate 5G NOMA-CAP signals with whatever configuration they wish for. While developing the application the original code was also improved to add more functionalities. The end goal of this project is to study NOMA-CAP and to assist other people to further study NOMA-CAP with the help of the application.

Three different situations were also studied. We studied the effects that symbol rate, modulation type and relative power between users has on NOMA-CAP. At the same time, we proved the correct functionality of the application as well as the easiness of using the application and configuring a NOMA-CAP signal with it.

Bibliography:

[1] Jose Antonio Altabas, Simon Rommel, Rafael Puerta, David Izquierdo, Juan Ignacio Garces, Jose Antonio Lazaro, Juan José Vegas Olmos, and Idelfonso Tafur Monroy, "Nonorthogonal Multiple Access and Carrierless Amplitude Phase Modulation for Flexible Multiuser Provisioning in 5G Mobile Networks," J. Lightwave Technol. 35, 5456-5463 (2017)

[2] S. Sarmiento et al., "Optical Power Budget Enhancement in 50-90 Gb/s IM-DD PONs with NOMA-CAP and SOA-based Amplification," in IEEE Photonics Technology Letters.

[3] Samael Sarmiento, José Manuel Delgado Mendinueta, José Antonio Altabás, Salvatore Spadaro, Satoshi Shinada, Hideaki Furukawa, Juan José Vegas Olmos, José Antonio Lázaro and Naoya Wada, “Split-enabled 350-630 Gb/s Optical Interconnect with Direct Detection NOMA-CAP and 7-Core Multi-Core Fiber,” Elsevier Optics Communications, vol. 463, pp. 125321, May 2020. https://doi.org/10.1016/j.optcom.2020.125321

[4] Samael Sarmiento, José Manuel Delgado Mendinueta, José Antonio Altabás, Salvatore Spadaro, Satoshi Shinada, Hideaki Furukawa, Juan José Vegas Olmos, José Antonio Lázaro and Naoya Wada, “Converged RoF-based mobile fronthaul and passive optical network with NOMA-CAP modulation format,” in Proc. Broadband Access Communication Technologies XIV, San Francisco, USA, February 2020.

[5] Samael Sarmiento, José Manuel Delgado Mendinueta, José Antonio Altabás, Salvatore Spadaro, Satoshi Shinada, Hideaki Furukawa,Juan José Vegas Olmos, José Antonio Lázaro and Naoya Wada, “Split-enabled 490 Gb/s optical interconnect with direct detectionNOMA-CAP and 7-CORE multi-core fibre,” in Proc. European Conference on Optical Communication (ECOC), Dublin, Ireland, September 2019.

[6] Samael Sarmiento, José Manuel Delgado Mendinueta, José Antonio Altabás, Salvatore Spadaro, Satoshi Shinada, Hideaki Furukawa, Juan José Vegas Olmos, José Antonio Lázaro and Naoya Wada, “Optical Power Budget Enhancement in 50 Gb/s IM-DD PONs with NOMA CAP Modulation and SOA-based Amplification,” in Proc. International Conference of Transparent Optical Network (ICTON), Angers, France, July 2019.

[7] Samael Sarmiento, José Manuel Delgado Mendinueta, José Antonio Altabás, Salvatore Spadaro, Satoshi Shinada, Hideaki Furukawa, Juan José Vegas Olmos, José Antonio Lázaro and Naoya Wada, “Experimental Investigation of 50-90 Gb/s IM-DD NOMA-CAP Modulation for Short Range Optical Transmission Applications,” in Proc. OptoElectronics and Communications Conference (OECC)/International Conference on Photonics in Switching and Computing (PSC), Fukuoka, Japan, July 2019.

[8] Samael Sarmiento, José. A. Altabás,Salvatore Spadaro and José A. Lázaro, “Experimental Assessment of 10 Gbps 5G Multicarrier Waveforms for High-Layer Split U-DWDM-PON-Based Fronthaul,” IEEE Journal of Lightwave Technology, vol. 37, no. 10, pp. 2344-2351, May 2019. https://doi.org/10.1109/JLT.2019.2904114

[9] Samael Sarmiento, José A. Altabás, Salvatore Spadaro, José A. Lázaro, “From 4.2 Gbps Asymmetrical Clipping (ACO)-OFDM to 8.7 Gbps Layered-ACO-FBMC with Intensity-Modulation Direct-Detection for PONs,” in Proc. Conference on Lasers & Electro-Optics (CLEO)/Europe, Munich, Germany, June 2019.

[10] Samael Sarmiento, Albert Gran, José A. Altabás, Marco Scalabroni, Salvatore Spadaro, Ignacio Garcés, José A. Lázaro, "Experimental Assessment of 5-10 Gbps 5G Multicarrier Waveforms with Intensity-Modulation Direct-Detection for PONs," in Proc. Photonics in Switching and Computing (PSC), Limassol, Cyprus, September 2018.

[11] Mahmoud Aldababsa, Mesut Toka, Selahattin Gökçeli, Güneş Karabulut Kurt, Oğuz Kucur "*A Tutorial on Nonorthogonal Multiple Access for 5G and Beyond*" Article ID 9713450

[12] Refik Caglar Kizilirmak "*Non-Orthogonal Multiple Access (NOMA) for 5G Networks*" December 2016 DOI: 10.5772/66048

[13] Yingmin Wang, Bin Ren, Shaohui Sun, Shaoli Kang, Xinwei Yue "*Analysis of non-orthogonal multiple access for 5G*" 2016 IEEE 10.1109/CC.2016.7833460

[14] Miguel Iglesias Olmedo, Tianjian Zuo, Jesper Bevensee Jensen, Qiwen Zhong, Xiaogeng Xu, Sergei Popov, and Idelfonso Tafur Monroy "*Multiband Carrierless Amplitude Phase Modulation for High Capacity Optical Data Links*" in Journal of Lightwave Technology, vol. 32, no. 4, pp. 798-804, Feb.15, 2014.

[15] K. Werfli *et al.*, "Multi-band carrier-less amplitude and phase modulation with decision feedback equalization for bandlimited VLC systems," *2015 4th International Workshop on Optical Wireless Communications (IWOW)*, Istanbul, 2015, pp. 6-10.J. P. Wilkinson, "Nonlinear resonant circuit devices," U.S. Patent 3 624 125, July 16, 1990.

[16] Mohd Ridzuan, Noridah; Abdullah, Mohammad Faiz Liew; Othman, Maisara; Jaafar, Marliana "*A Carrierless Amplitude Phase (CAP) Modulation Format: Perspective and Prospect in Optical Transmission System*" 10.11591/ijece.v8i1.pp585-595 International Journal of Electrical and Computer Engineering

[17] Rafael Puerta, Simon Rommel, Jose A. Altabas, Line Pyndt, Raya Idrissa, Albert Kh. Sultanov, Juan J. Vegas Olmos, Idelfonso Tafur Monroy *"Multiband carrierless amplitude/phase modulation for ultrawideband high data rate wireless communications"* mop.29866

[18] https://commons.wikimedia.org/wiki/File:CAP_transmitter_and_receiver.svg

[19] https://www.glassdoor.es/Sueldos/barcelona-junior-engineer-sueldo-SRCH_IL.0,9_IM1015_KO10,25.htm?countryRedirect=true

Appendices:

On this appendix we will observe the different values for all the tests in the following tables:

Table 2 Test 1: SNR/BER values for 4 QAM

SNR	BER	log10(BER)
0	0.0989	-1.00480371
1	0.0735	-1.13371266
2	0.0515	-1.28819277
3	0.035	-1.45593196
4	0.0198	-1.70333481
5	0.0119	-1.92445304
6	0.0046	-2.33724217
7	0.0015	-2.82390874
8	6.10E-04	-3.21442105
9	2.44E-04	-3.61236106

Table 3 Test 1: SNR/BER values for 16 QAM

SNR	BER	log10(BER)
0	0.2311	-0.63620005
1	0.2111	-0.67551177
2	0.1861	-0.73025363
3	0.1619	-0.79075315
4	0.1364	-0.86518563
5	0.1166	-0.93330145
6	0.0923	-1.0347983
7	0.0691	-1.16052195
8	0.05	-1.30103
9	0.0351	-1.45469288
10	0.0233	-1.63264408
11	0.0132	-1.87942607
12	0.0066	-2.18045606
13	0.0029	-2.537602
14	0.0014	-2.85387196
15	5.80E-04	-3.23669932
16	9.16E-05	-4.03832742

Table 4 Test 1: SNR/BER values for 64 QAM

SNR	BER	log10(BER)
0	0.3209	-0.49363028
1	0.3021	-0.51984927
2	0.2856	-0.5442418
3	0.2626	-0.58070528
4	0.2418	-0.6165437
5	0.2212	-0.65521488
6	0.2043	-0.68973163
7	0.182	-0.73992861
8	0.1616	-0.79155864
9	0.1473	-0.83179725
10	0.123	-0.91009489
11	0.1051	-0.97839728
12	0.0872	-1.05948352
13	0.0733	-1.13489603
14	0.0564	-1.2487209
15	0.0434	-1.36251027
16	0.0301	-1.5214335
17	0.0206	-1.68613278
18	0.0127	-1.89619628
19	0.0086	-2.06550155
20	0.0053	-2.27572413
21	0.0029	-2.537602
22	0.0014	-2.85387196
23	7.53E-04	-3.1233377
24	3.26E-04	-3.48742232
25	1.63E-04	-3.78845232
26	1.22E-04	-3.91339106

Table 5 Test 2: SNR BER values for all three bands

SNR	Band1	Band2	Band3	log10(Band1)	log10(Band2)	log10(Band3)
0	0.2787	0.3321	0.3842	-0.55486303	-0.47873112	-0.41544264
1	0.2629	0.3218	0.3746	-0.58020941	-0.49241396	-0.42643223
2	0.2358	0.2948	0.3579	-0.6274562	-0.53047252	-0.4462383
3	0.2121	0.2712	0.3393	-0.67345933	-0.56671031	-0.46941614
4	0.183	0.2432	0.3278	-0.73754891	-0.61403643	-0.48439105
5	0.1519	0.2235	0.3071	-0.81844223	-0.65072247	-0.51272018
6	0.1304	0.1992	0.2913	-0.88472241	-0.70071067	-0.53565952
7	0.1026	0.1742	0.2653	-0.98885264	-0.75895185	-0.57626275
8	0.0809	0.1515	0.2472	-1.09205148	-0.81958737	-0.60695153
9	0.0585	0.1266	0.2248	-1.23284413	-0.89756629	-0.64820369
10	0.0352	0.1017	0.2042	-1.45345734	-0.99267905	-0.68994426
11	0.0226	0.0822	0.181	-1.64589156	-1.08512818	-0.74232143
12	0.0141	0.0609	0.1675	-1.85078089	-1.21538271	-0.77598519
13	0.0067	0.0432	0.1448	-2.1739252	-1.36451625	-0.83923144
14	0.0034	0.0308	0.1271	-2.46852108	-1.51144928	-0.89585445
15	0.0006	0.0188	0.1072	-3.22184875	-1.72584215	-0.96980521
16	0.0003	0.0116	0.0918	-3.52287875	-1.93554201	-1.03715732
17		0.0053	0.0722		-2.27572413	-1.1414628
18		0.0023	0.0578		-2.63827216	-1.23807216
19		0.001	0.0431		-3	-1.36552273
20		0.0002	0.0303		-3.69897	-1.51855737
21		0.0001	0.0194		-4	-1.71219827
22			0.0121			-1.91721463
23			0.007			-2.15490196
24			0.0035			-2.45593196
25			0.0017			-2.76955108
26			0.00069173			-3.16006228
27			0.00036621			-3.43626869
28			0.00018311			-3.73729868

Table 6 Test 3: SNR BER values for 0 RP

SNR	BER Strong	Ber Weak
0	0.2891	0.3777
1	0.2751	0.3644
2	0.2665	0.3613
3	0.2517	0.3365
4	0.2507	0.3285
5	0.2463	0.3099
6	0.2491	0.3031
7	0.2468	0.2994
8	0.2499	0.3069

Table 7 Test 3: SNR BER values for 2 RP

SNR	BER Strong	Ber Weak
0	0.2339	0.3936
1	0.2219	0.3743
2	0.2122	0.3579
3	0.2012	0.3327
4	0.1914	0.3062
5	0.1934	0.3091
6	0.2021	0.3162
7	0.2047	0.315

Table 8 Test 3: SNR BER values for 4 RP

SNR	BER Strong	Ber Weak
0	0.1987	0.402
1	0.1804	0.3766
2	0.1638	0.3544
3	0.1518	0.323
4	0.1336	0.2864
5	0.1232	0.252
6	0.1129	0.2179
7	0.1031	0.1866
8	0.0882	0.1532
9	0.074	0.1151
10	0.0687	0.1037
11	0.0521	0.068
12	0.0422	0.0515
13	0.0306	0.0346
14	0.0222	0.0231
15	0.0167	0.017
16	0.0087	0.009
17	0.0057	0.0057
18	0.0034	0.0034
19	0.0018	0.0018
20	0.0013	0.0013
21	8.54E-04	8.54E-04
22	2.44E-04	2.44E-04
23	1.83E-04	1.83E-04
25	6.10E-05	6.10E-05

Table 9 Test 3: SNR BER values for 6 RP

SNR	BER Strong	Ber Weak
0	0.1774	0.4159
1	0.1578	0.4011
2	0.1375	0.3714
3	0.1182	0.3465
4	0.1036	0.308
5	0.0915	0.2663
6	0.072	0.2272
7	0.0602	0.1859
8	0.0477	0.151
9	0.0363	0.1107
10	0.0236	0.0787
11	0.0173	0.0521
12	0.0113	0.0316
13	0.0056	0.0183
14	0.0031	0.009
15	0.0018	0.0043
16	0.001	0.0019
17	1.22E-04	6.71E-04
18	6.10E-05	1.22E-04
19	10^-6	6.10E-05

Table 10 Test 3: SNR BER values for 8 RP

SNR	BER Strong	Ber Weak
0	0.1577	0.434
1	0.1372	0.4103
2	0.1133	0.3932
3	0.0987	0.3658
4	0.0755	0.3222
5	0.0613	0.2886
6	0.0486	0.2513
7	0.0349	0.2045
8	0.0252	0.1708
9	0.0162	0.1321
10	0.0098	0.099
11	0.0048	0.0684
12	0.0026	0.0497
13	0.0015	0.0325
14	3.66E-04	0.0192
15	1.83E-04	0.0091
16	6.10E-05	0.0052
17	0	0.002
18		3.66E-04
19		1.22E-04
20		6.10E-05

Table 11 Test 3: SNR BER values for 10 RP

SNR	BER Strong	Ber Weak
0	0.1206	0.4262
1	0.1003	0.4109
2	0.0842	0.3833
3	0.0652	0.3529
4	0.0481	0.3174
5	0.0366	0.2831
6	0.0239	0.2522
7	0.0147	0.2112
8	0.0086	0.1779
9	0.0045	0.145
10	0.0017	0.1147
11	0.001	0.0908
12	6.71E-04	0.0894
13		0.0505
14		0.0317
15		0.0182
16		0.0103
17		0.0066
18		0.0028
19		0.0011
20		6.10E-04
21		6.10E-05

Table 12 Test 3: SNR BER values for 12 RP

SNR	BER Strong	Ber Weak
0	0.1104	0.436
1	0.0886	0.4214
2	0.063	0.3881
3	0.0484	0.3731
4	0.0329	0.3335
5	0.0218	0.3071
6	0.0126	0.2813
7	0.006	0.2471
8	0.0034	0.2078
9	0.0015	0.1891
10	6.10E-04	0.1586
11	2.44E-04	0.1296
12		0.1032
13		0.0813
14		0.0621
15		0.0413
16		0.0248
17		0.0176
18		0.009
19		0.0051
20		0.0016
21		0.0011
22		3.05E-04
23		6.10E-05

Table 13 Test 3: SNR BER values for 14 RP

SNR	BER Strong	Ber Weak
0	0.1026	0.4492
1	0.0801	0.4517
2	0.0607	0.422
3	0.0403	0.3881
4	0.027	0.3635
5	0.0192	0.3508
6	0.0084	0.3173
7	0.0041	0.2937
8	0.002	0.2637
9	6.71E-04	0.2382
10	1.83E-04	0.2133
11		0.1812
12		0.152
13		0.1308
14		0.1014
15		0.0802
16		0.0576
17		0.0406
18		0.0286
19		0.0162
20		0.0099
21		0.005
22		0.0031
23		0.0016
24		3.05E-04
25		1.22E-04
26		1.83E-04
27		6.10E-05

Glossary

NOMA - Non-Orthogonal Multiple Access

CAP - Carrierless Amplitude Phase Modulation

SNR - Signal-to-noise Ratio

BER - Bit Error Ratio

UI – User Interface

WBS - Work-breakdown Structure

OMA - Orthogonal Multiple Access

FDMA - Frequency Division Multiple Access

TDMA - Time Division Multiple Access

CDMA - Code Division Multiple Access

OFDMA - Orthogonal Frequency Division Multiple Access

SIC - Successive Interference Cancellation

PDMA - Pattern Division Multiple Access

BDM - Bit Division Multiplexing

QAM - Quadrature Amplitude Modulation

ITU - International Telecommunication Union

DMT - Discrete Multitone Modulation

ADSL - Asymmetric Digital Subscriber Line

RAN – Radio Access Network

DFE - Decision Feedback Equalizer

PRBS - Pseudorandom Binary Sequence