



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA

– **TELECOM** ESCUELA  
TÉCNICA **VLC** SUPERIOR  
DE INGENIERÍA DE  
TELECOMUNICACIÓN

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

School of Telecommunications Engineering

Polarization effects in Visible Light Communications

End of Degree Project

Bachelor's Degree in Telecommunication Technologies and  
Services Engineering

AUTHOR: Chirivella Ciruelos, Alejandro Miguel

Tutor: Ortega Tamarit, Beatriz

External cotutor: ZVANOVEC, STANISLAV

Experimental director: GUERRA YANEZ, CARLOS

ACADEMIC YEAR: 2021/2022



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA

— **TELECOM** ESCUELA  
TÉCNICA **VLC** SUPERIOR  
DE INGENIERÍA DE  
TELECOMUNICACIÓN

## Resumen

Los sistemas de comunicación por luz visible (VLC) forman parte de una de las áreas de las comunicaciones ópticas más prometedoras en los próximos años debido a sus múltiples aplicaciones. Uno de los retos que presentan estos sistemas es poder trabajar a altas velocidades, ya que el uso de LED en lugar de láseres limita mucho el ancho de banda que se puede usar. Existen diversas formas de conseguir este aumento en la velocidad de transmisión como la ampliación del ancho de banda con el diseño de circuitos específicos, el uso de modulaciones avanzadas, el empleo de técnicas de procesamiento digital de la señal o la multiplexación por división de polarización (PDM). En nuestro caso, estudiaremos la implementación de esquemas PDM en enlaces VLC. Así bien, haremos uso de diversos componentes ópticos para la división y combinación de haces para llevar a cabo la multiplexación de las polarizaciones, y estudiaremos, bajo diferentes condiciones de trabajo, cómo se comporta el sistema y cuáles son las posibles velocidades que obtenemos, siempre conservando unos valores suficientes en los parámetros de medida (EVM, BER y SNR) que aseguren la calidad del enlace.

## Resum

Els sistemes de comunicació amb llum visible (VLC) formen part d'una de les àrees de les comunicacions òptiques més prometedores en les pròxims anys degut a les seues diverses aplicacions. Un dels reptes que presenten aquests sistemes es poder treballar a altes velocitats, ja que l'ús de LED en lloc de làsers limita molt l'ample de banda que es pot utilitzar. Existeixen diverses formes d'aconseguir aquest augment en la velocitat de transmissió com: l'ampliació de l'ample de banda amb el disseny de circuits específics, l'ús de modulacions avançades, processament digital de la senyal o la multiplexació per divisió de polarització (PDM). En aquest cas, estudiarem la implementació d'esquemes PDM en enllaços VLC. Així, farem ús de diversos components òptics per a la divisió i combinació de feixos per a dur a terme la multiplexació de les polaritzacions, i estudiarem, baix diferents condicions de treball, com es comporta el sistema i quines son les possibles velocitats que obtenim, sempre conservant uns valors suficients en els paràmetres de mesura (EVM, BER i SNR) que asseguren la qualitat de l'enllaç.

## Abstract

Visible light communication (VLC) systems are one of the most promising areas of optical communications in the coming years due to their multiple applications. One of the challenges presented by these systems is to be able to work at high speeds, since the use of LEDs instead of lasers greatly limits the bandwidth that can be used. There are several ways to achieve this increase in transmission speed, such as: increasing the bandwidth with the design of specific circuits, the use of advanced modulations, digital signal processing or polarization division multiplexing (PDM). In our case, we will study the implementation of PDM schemes in VLC links. Thus, we will make use of various optical components for division and combination of beams to carry out the polarization multiplexing, and we will study, under different working conditions, how the system behaves and what are the possible speeds that we obtain, always maintaining sufficient values in the measurement parameters (EVM, BER and SNR) to ensure the quality of the link.



## Table of contents

Chapter 1.	Introduction .....	1
1.1	Motivation and context.....	1
1.2	Goals .....	1
1.3	Structure .....	2
1.4	Gantt Diagram .....	2
Chapter 2.	VLC Fundamentals .....	6
2.1	State of the Art .....	6
2.2	Description of VLC systems .....	8
2.3	Components of VLC systems.....	8
2.4	Modulation techniques .....	9
2.5	Multiplexing techniques .....	9
2.5.1	Spatial Division Multiplexing (SDM) and MIMO.....	9
2.5.2	Color Division Multiplexing (CDM) .....	10
2.5.3	Polarization Division Multiplexing (PDM).....	10
Chapter 3.	Experimental setup.....	11
3.1	Experimental schemes.....	11
3.2	Equipment .....	13
3.3	Instrumentation Control .....	15
3.3.1	What is it and why do we need it?.....	15
3.3.2	SCPI Standard .....	15
3.3.3	Useful Commands .....	16
3.4	MATLAB Processing.....	17
Chapter 4.	Experimental results .....	19
4.1	Measurements .....	19
4.2	Rectangular pulse .....	20
4.3	Shaped pulse.....	25
4.4	Shaped pulse with EQ .....	37
Chapter 5.	Conclusion and future research lines.....	43
5.1	Conclusion.....	43
5.2	Future research lines.....	44



References.....	45
Appendix A. MATLAB code.....	46
Waveform Generator API .....	46
Oscilloscope API.....	47
OOK.....	48
Simple 4-QAM.....	48
Rectangular pulse measurements .....	49
Shaped pulse (raised cosine) measurements .....	51
Shaped pulse (raised cosine) with EQ measurements .....	56
Appendix B. Component datasheets .....	62
PDA36A and PDA10A (Photodetectors).....	62
LB1761-ML (Lens for LEDs).....	63
ACL2520-A (Lens for Photodetectors).....	64
PBS251 (Polarizing Beamsplitter Cube) .....	65
LPVISE100-A (Polarizer) .....	66

## Figure index

Figure 1. VLC Applications [1].	7
Figure 2. LED-Based Indoor Positioning System diagram.	8
Figure 3. Basic setup diagram.	11
Figure 4. Polarization multiplexing setup diagram.	12
Figure 5. Polarization multiplexing setup: General view.	12
Figure 6. Polarization multiplexing setup: Transmitter.	13
Figure 7. Polarization multiplexing setup: Receiver.	13
Figure 8. Waveform Generator (T3AWG3252).	14
Figure 9. Oscilloscope (MSOS104A).	14
Figure 10. Hierarchical structure of CALibration command.	16
Figure 11. Rectangular pulse in time and in frequency.	20
Figure 12. EVM for rectangular pulse and 250 kbaud of symbol rate.	21
Figure 13. BER for rectangular pulse and 250 kbaud of symbol rate.	22
Figure 14. SNR for rectangular pulse and 250 kbaud of symbol rate.	23
Figure 15. Constellation for rectangular pulse and 250 kbaud of symbol rate with 4-QAM, at 50 cm and a carrier frequency of 500kHz.	23
Figure 16. Constellation for rectangular pulse and 250 kbaud of symbol rate with 4-QAM, at 100 cm and a carrier frequency of 500kHz.	24
Figure 17. Constellation for rectangular pulse and 250 kbaud of symbol rate with 64-QAM, at 70 cm and a carrier frequency of 500kHz.	24
Figure 18. Constellation for rectangular pulse and 250 kbaud of symbol rate with 64-QAM, at 70 cm and a carrier frequency of 750kHz.	25
Figure 19. Electrical spectrum for rectangular pulse and 250 kbaud of symbol rate with 4-QAM, at 100 cm and a carrier frequency of 500kHz.	25
Figure 20. Raised cosine filter.	26
Figure 21. EVM for shaped pulse and 250 kbaud of symbol rate.	26
Figure 22. BER for shaped pulse and 250 kbaud of symbol rate.	27
Figure 23. SNR for shaped pulse and 250 kbaud of symbol rate.	28
Figure 24. Constellation for shaped pulse and 250 kbaud of symbol rate with 4-QAM, at 50 cm, 0° and a carrier frequency of 250 kHz.	28
Figure 25. Constellation for shaped pulse and 250 kbaud of symbol rate with 64-QAM, at 70 cm, 0° and a carrier frequency of 250 kHz.	29
Figure 26. EVM in terms if angle for shaped pulse and 250 kbaud of symbol rate.	29
Figure 27. BER in terms if angle for shaped pulse and 250 kbaud of symbol rate.	30
Figure 28. SNR in terms if angle for shaped pulse and 250 kbaud of symbol rate.	30



Figure 29. Constellation for shaped pulse and 250 kbaud of symbol rate with 4-QAM, at 50 cm, -10° and a carrier frequency of 250 kHz. ....	31
Figure 30. Electrical spectrum for shaped pulse and 250kbaud of symbol rate with 4-QAM, at 100 cm, 0° and a carrier frequency of 250 kHz.....	31
Figure 31. EVM for shaped pulse and 500 kbaud of symbol rate.....	32
Figure 32. BER for shaped pulse and 500 kbaud of symbol rate.....	33
Figure 33. SNR for shaped pulse and 500 kbaud of symbol rate.....	33
Figure 34. Constellation for shaped pulse and 500 kbaud of symbol rate with 4-QAM, at 50 cm, 0° and a carrier frequency of 500 kHz.....	34
Figure 35. Constellation for shaped pulse and 500 kbaud of symbol rate with 64-QAM, at 70 cm, 0° and a carrier frequency of 500 kHz.....	34
Figure 36. EVM in terms of angle for shaped pulse and 500 kbaud of symbol rate.....	35
Figure 37. BER in terms of angle for shaped pulse and 500 kbaud of symbol rate.....	35
Figure 38. SNR in terms of angle for shaped pulse and 500 kbaud of symbol rate.....	36
Figure 39. Constellation for shaped pulse and 500 kbaud of symbol rate with 4-QAM, at 50 cm, -10° and a carrier frequency of 500 kHz. ....	37
Figure 40. Electrical spectrum for shaped pulse and 500 kbaud of symbol rate with 4-QAM, at 100cm, 0° and a carrier frequency of 500 kHz.....	37
Figure 41. EVM in terms if angle for shaped pulse with EQ and 250 kbaud of symbol rate. ....	38
Figure 42. BER in terms if angle for shaped pulse with EQ and 250 kbaud of symbol rate.....	39
Figure 43. SNR in terms if angle for shaped pulse with EQ and 250 kbaud of symbol rate.....	39
Figure 44. Constellation for shaped pulse with EQ and 250 kbaud of symbol rate with 4-QAM, at 50 cm, -10° and a carrier frequency of 250 kHz.....	40
Figure 45. EVM in terms if angle for shaped pulse with EQ and 500 kbaud of symbol rate. ....	41
Figure 46. BER in terms if angle for shaped pulse with EQ and 500 kbaud of symbol rate.....	41
Figure 47. SNR in terms if angle for shaped pulse with EQ and 500 kbaud of symbol rate.....	42
Figure 48. Constellation for shaped pulse with EQ and 500 kbaud of symbol rate with 4-QAM, at 50 cm, -10° and a carrier frequency of 500 kHz.....	42

## Table index

Table 1. Gantt Diagram.....	3
Table 2. Useful commands for Waveform Generator. ....	16
Table 3. Useful commands for Oscilloscope.....	17
Table 4. Parameters for each scenario of measurement. ....	20
Table 5. Suitable conditions for variable distance measurements.....	43
Table 6. Suitable conditions for variable angle measurements. ....	43



## Chapter 1. Introduction

### 1.1 Motivation and context

VLC (Visible Light Communication) is one of the most rising optical communication techniques in the last times because of its possible implementation in a lot of different areas of work. A lot of different researchers and business are making efforts on developing this area of communications to be able to give new services to their customers, such as Li-Fi connectivity for home and office networks, hospital networks, underwater communication, street lightning communication or even vehicle communication [1][2].

Some of the current investigations try to deal with the goal of improving the transmission speed we can achieve using these systems, because as we do not use lasers, speeds are usually lower than they would if we used a conventional optical link, such as an optical fiber link. As we said, we do not use lasers in this type of systems, instead we use LEDs which have much lower bandwidth of some MHz, as a main limitation in order to have large transmission speeds.

Some of the newest researches have obtained aggregate data rates up to 1 Gb/s using memory-controlled deep LSTM (long short-term memory) neural network post equalizers and PAM4 and PAM8 modulation schemes [3]. Also, data rates up to 4.5 Gb/s have been obtained using WDM (wavelength division multiplexing), CAP64 modulation and Volterra series-based nonlinear equalizers [4]. Both of them use LEDs as transmitters instead of lasers and using these techniques they are able to achieve these high data rates even though LEDs have much lower bandwidth. These speeds have been tested in indoor spaces and with distances around 2 meters.

One of the most promising ways in order to achieve this improvement in transmission speed is using polarization multiplexing schemes to maximize the number of information we are able to properly transmit per unit of time. This project focuses on analyzing how we can implement these polarization multiplexing schemes to the simplest possible VLC scheme (point-to-point link) and measuring how different pre and post processing mechanism can improve the quality of the communication link in order to maximize the transmission speed while conserving good signal quality (good BER, EVM, SNR, etc.)

This project is tutorized by Beatriz Ortega, professor in UPV (Universitat Politècnica de València), and prof. Ing., Ph.D. Stanislav Zvánovec, professor in CTU (Czech Technical University in Prague), and Carlos Guerra, PhD student at the latter university, are also co-advisors of this project. Both universities work in collaboration projects in the optical communications field. All the measurements and laboratory work were done in the CTU optical laboratory in Prague, where they let me use their laboratory, components, and equipment in order to implement all the experimental setups and make all the needed measurements about them.

### 1.2 Goals

As explained in the previous section, one of the goals nowadays in VLC is to increase the transmission speed by the use of different techniques. In our case, this technique is polarization multiplexing. So, in this project the main goal is to demonstrate polarization multiplexed VLC communication systems as a Proof of Concept. This will give us a good overview on how polarization multiplexing schemes work and why they are suitable to be deployed in future VLC systems. Moreover, different pre- and post- processing techniques will be employed to improve the achieved data rates while keeping the quality of the data link.

More specifically, the objectives of this work are the following:

- Demonstrate a polarization multiplexed VLC link.
- Evaluate the system performance versus distance and orientation angle between the transmitter and the receiver.



- Evaluate the link performance for different modulating pulses.
- Evaluate the link performance for different orders of QAM modulation (4-, 16- and 64-QAM).
- Evaluate the use of carrier and symbol synchronizers and equalizers.

### 1.3 Structure

This project is structured in 5 chapters plus references and appendices, as described in the following.

First, in this chapter (Chapter 1) a brief introduction is presented to justify the interest of the topic and the main goals of this work. The structure of this text and also a time diagram of the carried work have been included.

Then, in Chapter 2 we establish some context on VLC for the better understanding of the topic, and we talk about the State-of-the-art and fundamentals of VLC systems also including polarization multiplexing among other multiplexing techniques.

In Chapter 3, we describe the experimental setups we used for testing and to perform the measurements. Also, we talk about Instrumentation Control, which is key in our experimental approach for measuring.

In Chapter 4 we dive right into our measurements, describing how was our approach on each one of the different scenarios of our work and comparing the different results we obtained in each scenario to understand the impact of different parameters and changes on the system performance.

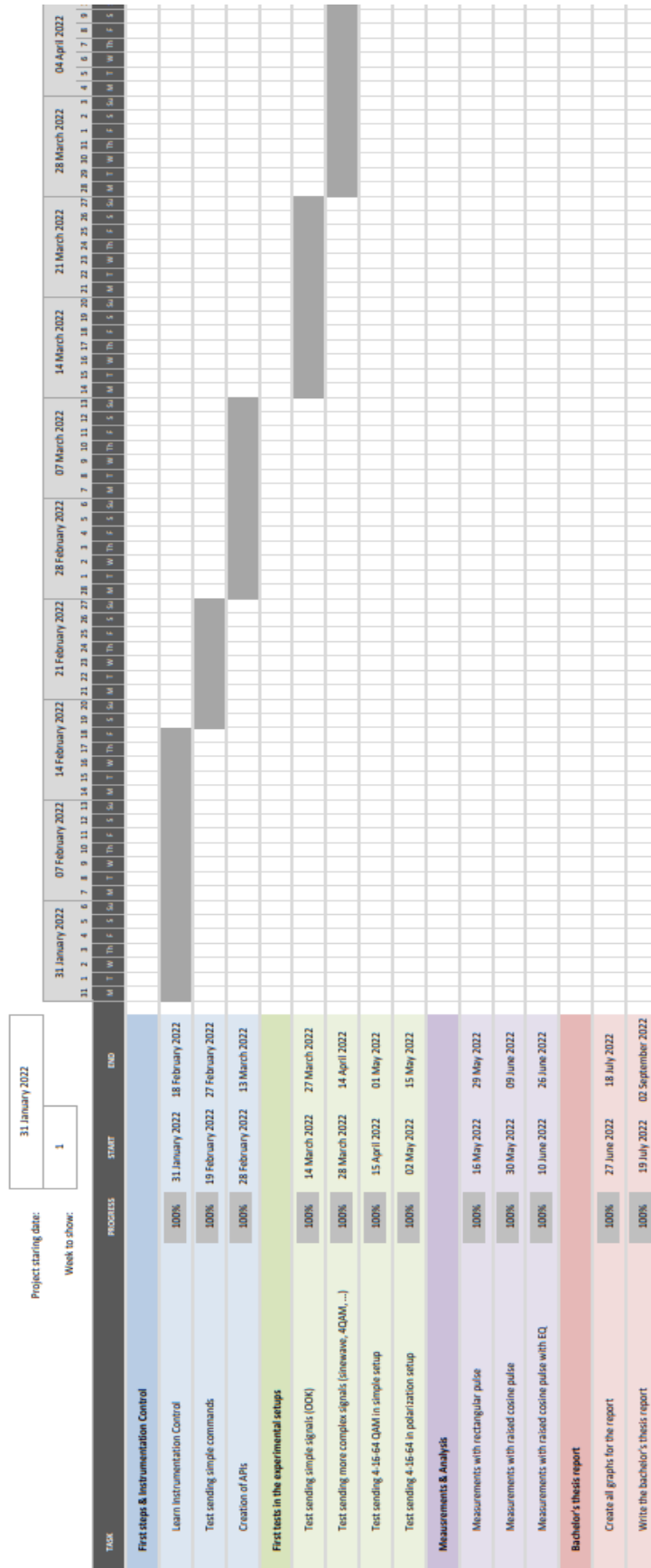
Finally, in Chapter 5, we provide the main conclusions of this work. Also, we propose future works that can be done to further improve the system.

References have been listed at the end of the document and also, two appendices: the first one about the MATLAB codes we have used during our project, and another one with some datasheet information about the components used on our VLC link.

### 1.4 Gantt Diagram

This work started on January 31<sup>st</sup> and was structured in different tasks as described in Table 1. Time duration of the tasks is also shown in the table.

Table 1. Gantt Diagram







## Chapter 2. VLC Fundamentals

### 2.1 State of the Art

VLC stands for Visible Light Communication. It is a communications technique that uses light only in the visible area of the radioelectric spectrum (from 375 to 780 nm). In this technique, the optical beam does not go through an optical fiber nor other type of waveguide but transmits through free space. Its main purpose is to be integrated with already existing light systems which were installed to light up rooms or other spaces, so they were not meant to be able to handle communications [5]. That is why VLC systems use fluorescent lamps or LEDs as the transmitters of the communication system, they do not use other transmitter such as lasers. This fact makes more difficult to be able to transmit at high speeds, because fluorescent lights can transmit at a maximum speed of 10 kbps approximately and LEDs at 500 Mbps for short path links. However, key features that motivate using VLC are energy efficiency, big unregulated bandwidth, license-free operation, high signal-to-noise ratio, interference immunity, low-cost hardware and data security amongst others [1].

As mentioned in Chapter 1. Introduction, new researches have obtained aggregate data rates up to 1 Gb/s using memory-controlled deep LSTM (long short-term memory) neural network post equalizers and PAM4 and PAM8 modulation schemes [3]. Also, data rates up to 4.5 Gb/s have been obtained using WDM (wavelength division multiplexing), CAP64 modulation and Volterra series-based nonlinear equalizers [4]. Both of them use LEDs as transmitters instead of lasers and using these techniques they are able to achieve these high data rates even though LEDs have much lower bandwidth. These speeds have been tested in indoor spaces and with distances around 2 meters.

Regarding the receiver, we only need a photodiode that transforms the optical signal into electrical signal, or even cameras such as those included in mobile phone cameras or digital cameras.

Wireless communications are getting day by day more demanding in terms of speed and network capacity, that is why it is necessary to find ways of optimizing them using new techniques and making them compatible with the already existing ones. Nowadays, is considered as a promising technology, especially for indoor systems. Indeed, it can be used in point-to-point communications between devices or also in multiaccess systems such as indoor Li-Fi access points [6].

In terms of multiplexing, we can talk about different ways in which we can approach it in VLC. These are: spatial, color and polarization multiplexing. The first is the simplest one; due to the high directionality of VLC systems we can easily spatially multiplex the signals without causing much interference between them, but we can have problems with obstacles that can shadow the signals. Secondly, we can do color multiplexing which consists in sending different signals in different wavelength. However, since the optical sources are LEDs, their spectrum is relatively broad compared with lasers, so we are not going to be able to work with very dense WDM systems. Also, the different LED colors are quite limited, so we do not have a huge variety of them. This limits us to use just three or four colors (RGB, RGYB). A lot of research is currently being conducted towards using of polymer-based color-converters and GaN micro-LEDs to achieve a bigger variety of colors and expand our WDM density [7].

In Fig. 1, we see some of the applications VLC can have. One of the most important applications of VLC is Li-Fi which is bound to be the alternative to the very famous Wi-Fi [8]. This technology could be implemented in local area networks for hotels, offices buildings or even home networks. Also, it could be used in more critical networks such as hospital networks. Moreover, VLC is not only seen as a technology for networks in an office or home environment but, a technology with multiple applications. For example, applications in the robotic industry for communication between machines because of its natural immunity to electromagnetic radiation. Also, car communication to enable autonomous driving cars that connect between them and with traffic

systems [2], mobile underwater communication or even broadcasting systems implemented in the streetlights systems. Also, there are some businesses in the toy industry such as Disney that are trying to implement these types of system in their own products using LED-to-LED communication [1][9].

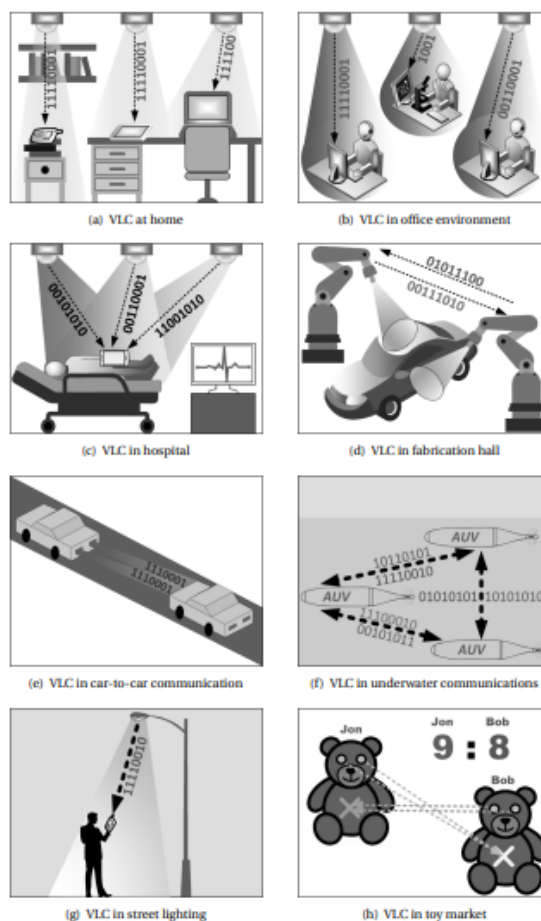


Figure 1. VLC Applications [1].

Some businesses are already offering commercial solutions over all these applications that VLC has. For example, the company Philipps is already offering LED-based indoor positioning systems. These systems use VLC links to connect smartphones with the lightning system using the mobile phone's camera to determine the position of the person [10]. In Fig. 2, we can see a brief diagram of how these technology works and its possible implementation in big supermarket shops to guide costumers.



**Figure 2. LED-Based Indoor Positioning System diagram.**

## 2.2 Description of VLC systems

A Visible Light Communication system has one or more transmitters and one or more receivers, and it uses the free space (no use of any type of waveguide such as fiber, etc.) as the channel for communication using the visible light spectrum which goes from 375 to 780 nm, which corresponds to frequencies from 430 THz to 790 THz. This spectrum is broader than RF one [11].

This type of systems can only receive the signal if they are in the same room as the transmitter (even if they are intended to be multi-access) because these wavelengths are very easily blocked by any obstacle, not as the RF signals which can cross walls and other types of obstacles. This gives security benefits to VLC systems to avoid eavesdropping of the communication [11].

Also, these systems can be combined with illumination ones, so it reduces the power consumption because the same system is used for both purposes: illumination and communication.

## 2.3 Components of VLC systems

Usually in VLC systems, LEDs are employed as the light sources for communicating because they are the best components to achieve both goals of illumination and communication. It is also possible to use incandescent light components, but LEDs perform way better in power usage, luminous efficiency, and reliability. The most used type of LEDs is the ones that emit white light, and its spectrum can be generated in various types of ways: tetra-chromatic, dichromatic, or tri-chromatic. The most typical one is tri-chromatic which uses the combination of red, green, and blue to generate the whole white light spectrum because it provides us a high bandwidth which results in high data rates, but it can be difficult to modulate. Dichromatic white LEDs use blue and yellow to generate the white spectrum and tetra-chromatic LEDs use blue, cyan, green and red. Also, phosphor-based LEDs can be used and modulated more easily but they do not perform as well as RGB LEDs do. RGB LEDs can achieve maximum data rates of 100Mbps and phosphor-based 50Mbps, being the firsts more expensive than the second ones [11].

Secondly, we must also talk about the other key component of a VLC system, the receiver. It has to be an optical detector that converts the incident light to an electrical signal, so we must use a photodiode. This will generate the electrical photo current signal with which we will be able to demodulate the signal. There are three main types of photodiodes used in VLC: silicon photodiode, PIN diode and avalanche photodiode. It is well-known that with PIN diode we will not have as much gain as if we use an avalanche photodiode, but the economic cost of our system will be reduced because PIN diodes are cheaper than the avalanche ones. Sometimes, it is also interesting to use optical filters to try to minimize interference from other light sources (we have to remember that VLC uses free space as the channel of communication) or even sunlight [11].



Apart from these two main components we also have some complementary components that help us with our communication. For example, if we have a point-to-point VLC link it would be interesting to use some lenses, both in the transmitter and in the receiver, to collimate the light and optimize the optical power received by the photodiode. The received power depends on how much light shines the area of the photodiode, so the more focused is the light inside this area the more power is received. Light cannot be endlessly collimated; it is something limited but even so it is possible to achieve some optimization in the received power.

On the other hand, we can also use polarizers to split our optical signals with different polarization states which, in combination with polarization splitters, can be very useful for deploying polarization multiplexing schemes in VLC (one of our schemes will use this).

## 2.4 Modulation techniques

Before sending our data through our channel, we have to prepare our data, i.e. modulating signal. It means to combine our symbols with a carrier frequency in a way that we can efficiently use the spectrum and we can maximize the information we want to send.

In VLC (as in any other field of communications), we also need to modulate the optical carrier by using the modulating signal in a direct modulation scheme. The most basic one is On-Off Keying (OOK) which consists of sending pulses when it is a '1' and not sending them when it is a '0'. Also, there are a bit more complex techniques as Pulse-Position Modulation (PPM) or Pulse-Amplitude Modulation (PAM). In the first one, we divide the symbol period in some slots and depending at which one the pulse appears we distinguish the symbols. In the other one, we distinguish between symbols depending on the amplitude of the received pulse.

We also consider Quadrature Amplitude Modulation (QAM) format, which is a very popular modulation that is widely used in the world of communications. In this modulation, symbols are distinguished depending on the phase (Q) and the amplitude (I) of the signal. This creates the very well-known I-Q constellations. Depending on the order of the modulation (4-QAM, 64-QAM, ...) more bits will be encoded in each symbol, although higher modulation levels lead to closer symbols in the constellation, and therefore, symbols decoding is more difficult. This will be the modulation used in our experimental setup [12].

## 2.5 Multiplexing techniques

In the world of telecommunications, multiplexing is one of the most important techniques to enhance the capacity of the systems. It consists of combining two or more signals, so they are able to be transmitted through the same transmission medium and then to be separated again in the receiver without any modification. This is a very easy way to optimize as much as possible the transmission medium and it can be used for example to improve the transmission speeds as we will use on our system. There are a lot of different multiplexing techniques, such as: time division multiplexing (TDM), frequency division multiplexing (FDM), wavelength division multiplexing (WDM) or code division multiplexing (CDM). Some of this are used very frequently in RF links and infrared optical links (be it fiber link or any other type). Nonetheless, we have some other types of multiplexing more specific to optical communications (i.e. VLC) such as: spatial division multiplexing (SDM), color division multiplexing (CDM) and polarization division multiplexing (PDM). Since these ones have been proposed to be used in VLC links to be able to enhance their transmission speeds, we will take a closer look to them in the next paragraphs.

### 2.5.1 Spatial Division Multiplexing (SDM) and MIMO

Spatial Division Multiplexing is a relatively new technique in the world of optical communications, while it was already used in the electronic world. It consists of separating each channel we want to multiplex in terms of the path it follows until the receiver. In VLC, we can achieve high directional channel, so SDM is a suitable option. We could even have an array of

LEDs, each one transmitting a different channel. However, VLC links, due to the nature of light, are very easy to block or shadow so this could be an important inconvenient in this type of setup. It is the most famous MIMO (multiple input multiple output) scheme because it transmits data streams at the same time independently at very high speeds. But there is one problem with it: channel correlation. If there is a channel with high correlation, the receivers will have a hard time trying to separate each one of the streams. As this is a usual feature of VLC links one solution is to use repetition codes (RC) but the tradeoff is that then we will not be able to multiplex gain, so we will have to use higher order constellation to maintain high speeds [13].

### **2.5.2 Color Division Multiplexing (CDM)**

In Color Division Multiplexing the visible light spectrum is divided into several color sub-bands that do not overlap with each other. In this way, the spectrum is shared by different colors, and they do not interfere with each other. Using some bandpass filters, it is possible select only one of these sub-bands both in the transmission and receiving process. In transmission we can use a white light LED and reduce its spectrum to a sub-band with this filter, and in reception we can select which sub-band we want to process also with this filter. To reduce possible interference, it is good to have some guard interval between sub-bands [14].

### **2.5.3 Polarization Division Multiplexing (PDM)**

The polarization of an electromagnetic wave is referred to the shape the electric field describes during its propagation. So, we can define 3 types of different polarizations: linear polarization, circular polarization, and elliptical polarization.

Linear polarization is the simplest one, the shape the electric field describes is a straight line. Here we can talk about vertical and horizontal polarization, which refer to when the straight line is vertical or horizontal. Combining these two, if they are in phase, we can create another vertical polarization turned by an angle. Also, if they are  $90^\circ$  out of phase, we will create a circular polarization. But if the vertical and horizontal polarizations have different magnitudes, we will create an elliptical polarization. Finally, we can also describe an elliptical polarization as the combination of two different circular polarizations.

So, if we take two different polarization that are orthogonal (for example vertical and horizontal polarization) we can combine them and then separate them again without any mixing between them. This is the fundamental idea behind Polarization Division Multiplexing (PDM). Here, we can theoretically double our bitrate by sending two signals in the same link using two orthogonal polarizations, so we can multiplex and demultiplex them in the transmitter and in the receiver.

## Chapter 3. Experimental setup

### 3.1 Experimental schemes

During the whole project we have used two different experimental schemes to make our measurements: the basic setup and the polarization multiplexing setup.

The first one (Fig. 3), the basic setup, was only used in the early scenarios of our project before making any measurements. We used it as a simple link to test the communication between the transmitter and the receiver to check that we were correctly receiving every piece of information. It consisted of a LED (the transmitter) a collimating lens and then the PD (photodetector, the receiver) that had another collimating lens attached to it. In VLC is not common to use collimating lenses because the purpose of VLC is to make point-to-multipoint connection, so collimating lenses are not useful there. But in our case, we are just trying to make a point-to-point link to, then, be able to study how we can get some advantages using polarization independently of any other effect (reflections, out-of-sight link, ...) Finally, the LED is connected to the waveform generator from where it receives the electric signal, and the APD is connected to the oscilloscope to read the received signal, send it to the computer and process it using MATLAB. The computer, the oscilloscope and the waveform generator are connected using a switch for the LAN (Local Area Network)

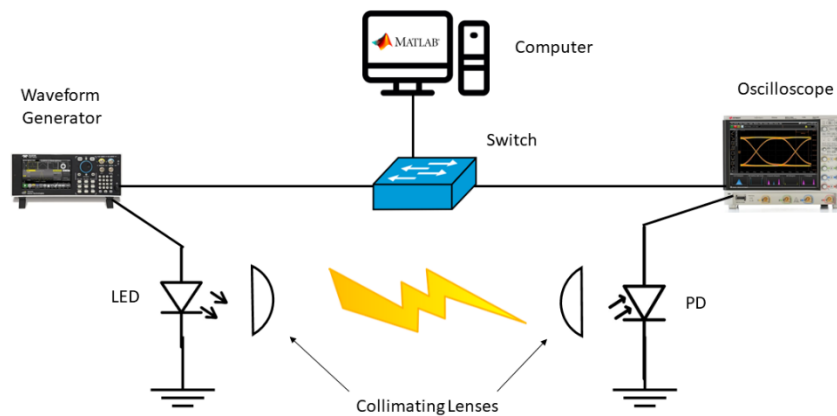


Figure 3. Basic setup diagram.

Then, we used the polarization multiplexing setup. This one (Fig. 4) is more complex than the other one, but it is somehow based on the basic setup. In this one, we have two channels, one for each polarization (vertical and horizontal). So, we have to create two links just as the one in the basic setup but with two new additions: something to polarize each channel and something to multiplex/demultiplex the signals. In order to achieve this, we added to the system two polarizers right next to the collimating lenses and two multiplexing beamsplitter cubes (one in the transmitter and another one in the receiver) to multiplex/demultiplex the signal. Just as in the basic setup the LED is connected to the waveform generator, and the PD (Photodetector) is connected to the oscilloscope but, now we have to use two different channels in both equipment, one for each polarization, to be able to send and receive different signals through each polarization. Also in this second experimental setup, the whole receiver is attached to a rail and a rotating wheel that makes us able to modify the distance and angle between the transmitter and the receiver in our measurements.

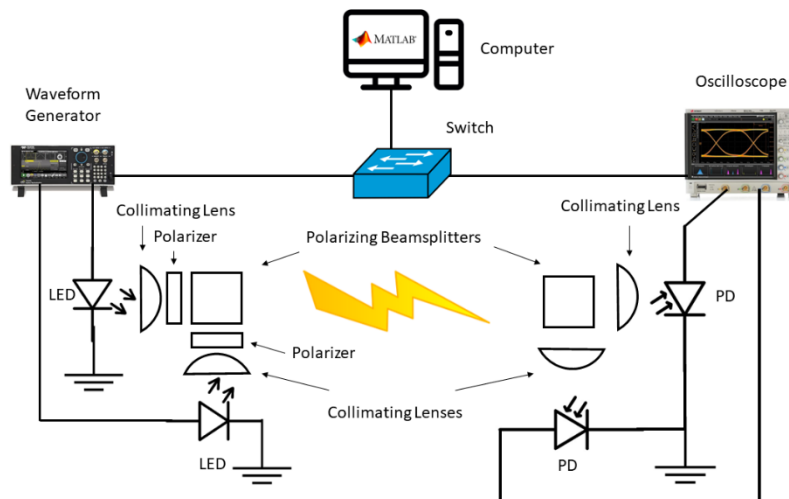


Figure 4. Polarization multiplexing setup diagram.

In Fig. 5 we can see a general view of how the real polarization multiplexing setup was implemented. In the left part of the figure, we can see the waveform generator and the transmitter part of the system. In the right part of the figure, we can see the oscilloscope and the receiver part of the system. We can notice how the receiver was mounted on some rails that made possible to move it forward and backwards, and also rotate it to change the angle.



Figure 5. Polarization multiplexing setup: General view.

In Fig.6 we can see the part of the system belonging to the transmitter. We can see the waveform generator and also the components that make the transmitter. It is easily noticeable both lenses with their LEDs inside, both polarizers right after them and then the polarizing beamsplitter cube.



Figure 6. Polarization multiplexing setup: Transmitter.

In Fig. 7 we can see the part of the system belonging to the receiver. We can see the oscilloscope and the components that make the receiver. We can easily see both PD with their lenses mounted and the polarizing beamsplitter cube. Also, we can see the rails that make possible the movement of the receiver.

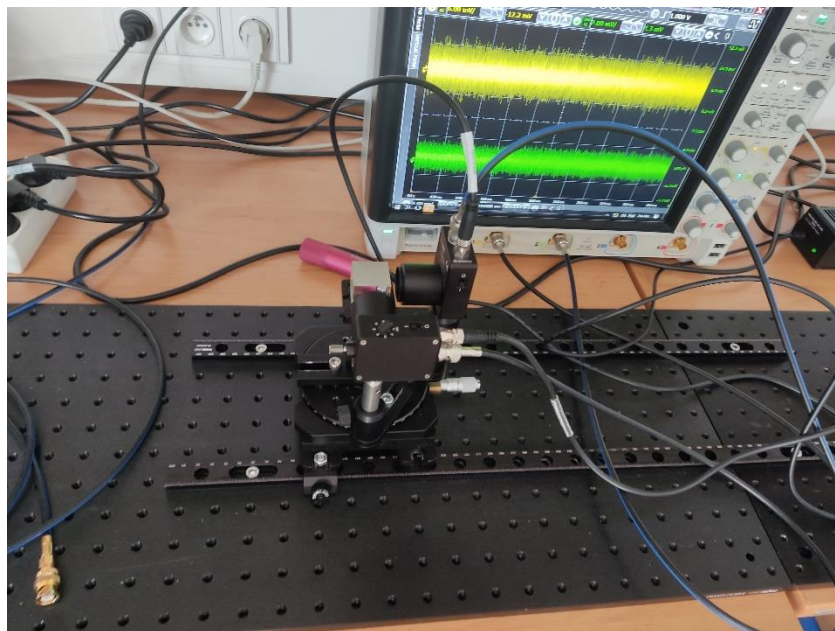


Figure 7. Polarization multiplexing setup: Receiver.

### 3.2 Equipment

We have two key important pieces of equipment in this project: the waveform generator and the oscilloscope.

The waveform generator is the one in charge of generating the electrical signal that will modulate the LED (transmitter of our system). We have connected each LED of each channel to different

output channels of the waveform generator, and it is also connected via LAN to the computer. Using MATLAB in the computer we are able to create the signal we want to send to each channel, and then configure the waveform generator with this information so it sends it to the corresponding LED. We used the waveform generator T3AWG3252 from the Teledyne LeCroy brand and we can see a picture of it in Fig. 8. Some of the key specifications of this waveform generator are frequency range from 1UHz to 250 MHz, a sample rate up to 1 GS/s, 16 bits of vertical resolution, memory up to 1Gpoint/Ch and an output voltage (peak to peak) of 12 V.



Figure 8. Waveform Generator (T3AWG3252).

On the other hand, we have the oscilloscope. This one is in charge of receiving the electrical signal from both APDs and the send it to the computer. It is connected to the computer via the same LAN as the waveform generator. When the computer receives the information from the oscilloscope it is able to process it using MATLAB to demodulate the signal and check the received data and calculate all our desired measurements. We used the oscilloscope MSOS104A from the Keysight Technologies brand and we can see a picture of it in Fig. 9. Some of the key specifications of this oscilloscope are analog bandwidth of 1 GHz, max sample rate of 20 GS/s for 2 channels or 10 GS/s for 4 channels, an ADC of 10 bits and memory up to 800 Mpoint/Ch for 2 channels or 400 Mpoint/Ch for 4 channels.

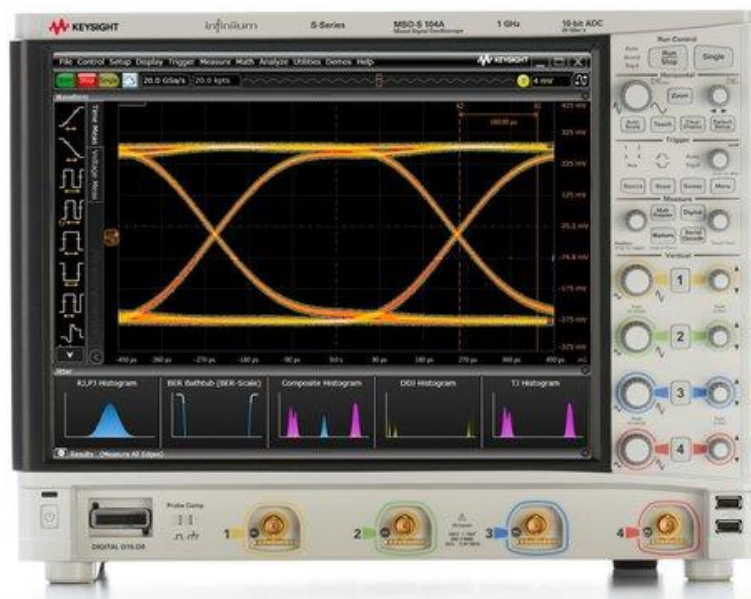


Figure 9. Oscilloscope (MSOS104A).

As mentioned above, these two machines and the computer were interconnected using a LAN network implemented by a simple switching device and some Ethernet cables. Both machines had

static private IP addresses, so the computer was able to send and receive information from them using a TCPIP socket (Oscilloscope) and a VISA socket (Waveform Generator).

### 3.3 Instrumentation Control

#### 3.3.1 *What is it and why do we need it?*

In our system we have two instruments we need to configure and control: the waveform generator and the oscilloscope. The first one generates the electric signal that the LED needs to transmit the data through our system and the second one receives the electric signal coming from the photodetector (APD) to analyze what data we received. We can configure and control both instruments manually, using all the buttons and manual controls they physically have, but there is a way to do this in a more efficient way.

Here is where Instrumentation Control comes into play. It makes us able to control the instruments and receive information from them remotely and in an automatic way, sending commands from another device (for example a computer). This makes it very easy and fast to configure and control the instruments. In a scenario like this, where we must twinkle a lot of different parameters, send different types of signals, and repeat the process a lot of times, the fact that we can change the configurations very fast and precisely is a key tool in our success and efficiency with our measurements.

MATLAB has a package called Instrumentation Control Toolbox to be able to implement this in an easy way in MATLAB scripts. This is perfect because we can generate the signal with MATLAB, send it and received through our system using the functionalities of this toolbox and then process it with MATLAB again where we can analyze the performance of our system and calculate our measurements. All the MATLAB scripts are in Appendix A.

#### 3.3.2 *SCPI Standard*

SCPI (Standard Commands for Programmable Instruments) is a standard that defines the syntax of the commands that are used to communicate with instruments. It also defines the data formats to use in the commands. These commands are encoded using ASCII and then encapsulated into the physical layer. There can be different types of physical links such as GPIB, Ethernet, USB, VISA, etc.

The commands have a hierarchical structure to group commands of the same kind under the same branch. Each level of the command is separated with colon (:) and then they are separated from the parameter (if there is any) with a blank space.

In Fig. 9 we can see, an example of the hierarchical structure of the command CALibration. Some example commands from this tree could be: CALibration:AUTO <Boolean>|ONCE, CALibration: BINertia:INITiate or :CALibration:PLOS:LSpeed <numeric\_values>.

KEYWORD	PARAMETER FORM	NOTES
CALibration		
[:ALL]		[event; no query] 1994
[:ALL]?		[event; query]
:AUTO	<Boolean> ONCE	
:BINertia		1999
:AVErage?	<numeric_value>	[query only] 1999
:HSPeed	<numeric_value>	1999
:INITiate		[event; no query] 1999
:LSPeed	<numeric_value>	1999
:NRUNs	<numeric_value>	1999
:SDEviation?	<numeric_value>	[query only] 1999
:UPDate		[event; no query] 1999
:DATA	<arbitrary block program data>	
:PLOSs		1999
:APCoeffs	<numeric_value>,<numeric_value>,...	1999
:INITiate		[event; no query] 1999
:LATime	<numeric_value>	1999
:STIME	<numeric_value>	1999
:UPDate		[event; no query] 1999
:SOURce	INternal EXternal	
:STATe	<Boolean>	
:VALue	<numeric_value>	
:WARMup		1999

Figure 10. Hierarchical structure of CALibration command.

Also, commands can be abbreviated a bit to make it more efficient while they are still different. If we want to use the abbreviated form of a command, we must only use the uppercase part of it. For example, the commands from last paragraph will look like this if we used their abbreviated form: CAL:AUTO <Boolean>|ONCE, CAL:BIN:INIT and CAL:PLOS:LSP <numeric\_value>.

Finally, if our commands it's a query (we need to get information back to the device who sent the command), we put an interrogation sign (?) after the command. For example, CAL:BIN:AVE?.

### 3.3.3 Useful Commands

In Table 2 and 3 we can find some of the most useful commands we can use in the configuration of our wave generator and oscilloscope:

Table 2. Useful commands for Waveform Generator.

Command	Parameters	Usage
*RST	None	Reset the device
WLIST:WAV:IMP	"name of file","destination folder"	Imports waveform from PC to the wave generator
SEQ:ELEM[num]:WAV	"name of the waveform"	Set a waveform in the element number [num]
AWGC:SRAT	<number>	Set the sampling rate
SEQ:ELEM[num]:LENG	<number>	Set number of samples
SEQ:ELEM[num]:VOLT:HIGH	<number>	Set high voltage value
SEQ:ELEM[num]:VOLT:LOW	<number>	Set low voltage value



OUTP1	<ON OFF>	Set output to ON or OFF
AWGC:RUN	None	Run the wave generator
MARK:LEV1	<number>	Set the level of the marker for synchronization

**Table 3. Useful commands for Oscilloscope.**

Command	Parameters	Usage
:TRIG:LEV	{ {CHANnel   AUX}, <level> }	Set trigger level
:TRIG:SWE	{ AUTO   TRIGgered   SINGLE }	Set trigger sweep mode
:TRIG:MODE	{ EDGE   GLITch   PATTErn   STATE   DELay   TIMEout   TV   COMM   RUNT   SEQuence   SHOLd   TRANsition   WINDow   PWIDth   ADVanced   SBUS }	Set trigger mode
:CHANnel[num]:INP	{ DC   DC50   AC   LFR1 }	Set input coupling
:ACQ:POIN	<number>	Set number of simples
:TIM:POS	<number>	Set central time position
:TIM:RANG	<number>	Set time range
:WAV:FORM	{ ASCii   BINary   BYTE   WORD   FLOat }	Set data format for transmission
:WAV:BYT	{ MSBFirst   LSBFirst }	Set byte order
:WAV:PRE?	None	Fetch preamble
:WAV:DATA?	None	Fetch data

### 3.4 MATLAB Processing

As I mentioned before, we have used MATLAB to control the equipment and to, then, process the data we receive and make our measurements.

First, in order to be able to send data through our link we need to be able to control the equipment, so MATLAB is perfect to do this using its native package called Instrumentation Control Toolbox. Therefore, we can establish our connection with the equipment and send commands to configure them and retrieve information from them. For the sake of simplicity and modularity, the best approach was to create two different API classes in MATLAB to control both of our pieces of equipment (Oscilloscope and Waveform Generator). The API for the oscilloscope consists of a constructor that creates the TCPIP socket to enable the connection between the computer and the oscilloscope, a configuration method that sets the sampling frequency, the number of samples, the time scale, the selected channel and the endianness, and finally, a read data method to fetch the data received from the photodetector through the oscilloscope.



On the other hand, we have the Waveform Generator API. This one has a constructor that creates the VISA socket to enable the communication between the computer and the waveform generator, a configuration method to set the sampling frequency the number of samples, the selected element (elements are the different sequencies we can set in the generator), the selected channel, the high and low voltage values, and the offset of the signal, and finally, a run and a reset method to start the sending of the signal and to reset the system.

Using these API objects, we created different scripts to test some transmissions. Initially we tried to send some simple OOK (On-Off Keying) signal, and little by little we tried to send more complex signal such as a sine wave and, finally a 4-QAM modulated signal. The Communications Toolbox from MATLAB makes us very easy to work with different types of modulations because it has a lot of methods that help us process the signals very easily (for example `qammod()` or `qamdemod()`) and the only thing we had to take care of was to properly modulate these signals with the carrier frequency and then demodulate and filter them in the postprocessing scenario. Finally, in these early scenarios of the project we just take as an important measure the BER (or symbol error rate) of our system because it was enough to assure if we were going in the right direction or not.

In this work, we sent 4-QAM signals through two different polarizations by using different carrier frequencies or symbol rates. These adjustments made synchronization of the system much more difficult, so we had to use some other objects from the Communication Toolbox such as a Carrier Synchronizer, a Symbol Synchronizer and a Linear Equalizer. We made some measurements with and without them to see how effective they were.

Finally, with the processed data we created constellations and electrical spectra of the received signals, and make the calculations needed to fulfill all the selected measurements.

After having measured everything that we wanted, we created another script that went through the .mat files where the measurements were saved to create all the graphics that we wanted to show the results in a more visual way than just a table made up with numbers. We also arranged the different charts in a way that we can easily compare the improvements between different order of modulation (4-QAM, 16-QAM and 64-QAM), different carrier frequencies and different symbol rates. All the MATLAB scripts we programmed can be seen in Appendix A.

## Chapter 4. Experimental results

### 4.1 Measurements

In this section, we show the measurements carried on the Polarization Multiplexing scheme. Here, we tried to modify our MATLAB pre- and post- processing to optimize our system and achieve quality measurements with the biggest bitrates possible. We can divide our measurements in 3 different scenarios: rectangular pulse, shaped pulse, and shaped pulse with EQ. These three scenarios were consecutive, so each one aimed to improve the measurements of the previous one, in order to obtain better quality in the communication with higher data rates. Later, when we talk about the results in each scenario, we will talk about the peculiarities of each one.

On the other hand, the signal quality measurements we decided to do on each scenario were the same: BER (total and for each polarization), EVM (for each polarization) and SNR (for each polarization). The first one is an easy measure, because we know what bits we sent (we generated them with MATLAB, modulated them and then sent them to the Waveform Generator) so using the biterr or symberr methods in MATLAB with the demodulated received signal we can calculate this parameter. To calculate the EVM is quite a similar process, because we also know the m-QAM symbols (we generated them in MATLAB) so we can use these and the m-QAM symbols we demodulated from the oscilloscope's received signal and apply the following formula to calculate it:

$$EVM = \sqrt{\frac{\frac{1}{N} \sum_i |S_i - R_i|^2}{\frac{1}{N} \sum_i |S_i|^2}} \quad (4.1)$$

Where  $N$  is the number of m-QAM symbols of our signal,  $S_i$  is one of the ideal m-QAM symbols and  $R_i$  is one of the received m-QAM symbols.

Finally, SNR was the hardest measure to do because we had to make an approximation. Our approach here was to measure the received electrical power before sending the signal to measure the noise level in the most accurate way. Then, we measured again the received electrical power while we sent the signal, so we obtained an approximation of the signal's power level. This is an approximation because in the second measurement we cannot avoid to also measure the noise level that is added to the signal's true power level:

$$SNR^* = \frac{S+N}{N} \quad (4.2)$$

$$SNR \cong SNR^* \text{ only if } N \ll S \quad (4.3)$$

Where  $SNR^*$  is the approximate SNR,  $SNR$  is the true SNR,  $S$  is the signal's power level and  $N$  is the noise power level. So, as we can see our approximation is only good if the noise power level is quite small compared to the signal's true power level. This fact, as we will see, will affect our measurements because we have one channel significantly noisier than the other. This will occur due to the nature of our scheme since Channel 2 signal has to reflect two times during its path to the receiver, while Channel 1 signal goes in a straight line which creates this difference between both channels. So, in Channel 2 our SNR calculations will not be as good as in Channel 1 where the approximation is somehow fairer.

So, once we decided which measurements we wanted to make, we had to decide which parameters to modify and observe the impact on the measurements. These parameters were: the symbol rate, the carrier frequency, the order of the QAM modulation, the distance from the transmitter to the receiver and the angle between them. We did not modify them in the same way in each of the aforementioned three scenarios or scenarios because we had to adapt a little bit to the performance of the system in each of them. It makes no sense to demand from the system the same performances in the worst scenario and the best scenario, because one would work well but the other one would work extremely well or extremely badly. In Table 4 we have a summary of the parameters in each scenario:

**Table 4. Parameters for each scenario of measurement.**

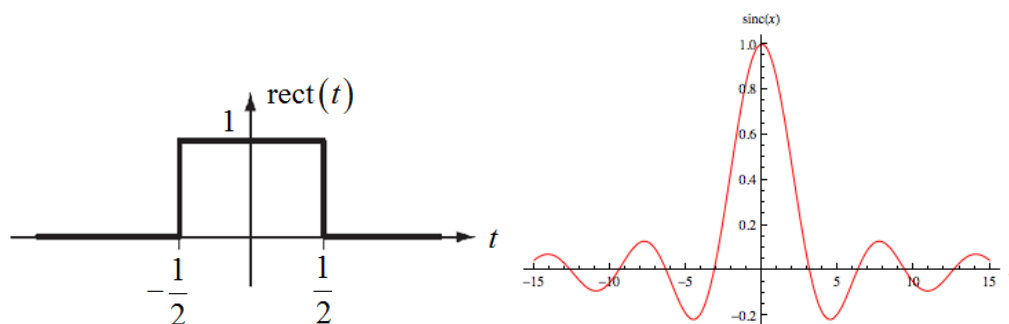
	Rectangular pulse	Shaped pulse		Shaped pulse with EQ	
Symbol rate (kbaud)	250	250	500	250	500
Carrier frequency (kHz)	500/750	250/500	500/750	250/500	500/750
QAM order	4/16/64	4/16/64		4/16	
Distance (cm)	From 50 to 100	From 50 to 100		50	
Angle (°)	0°	From -10° to 10° (d=50)		From -10° to 10° (d=50)	

## 4.2 Rectangular pulse

This is the first scenario of our measurements. Here, we tried to send our modulated QAM signal using rectangular pulses. As their own name indicates, they have a rectangular shape in time, but according to the Fourier transform, we know that they will have a sinc shape in frequency:

$$\text{rect}\left(\frac{t}{a}\right) \stackrel{\mathcal{F}}{\leftrightarrow} a * \text{sinc}\left(\frac{a\omega}{2}\right) \quad (4.4)$$

As we can see in Fig. 11, the fact of using a rectangular pulse in time creates a broader frequency spectrum due to the secondary lobes that extend to infinity. They have a much lower amplitude than the principal one, but we will see that the effect is visible when we look to the electrical spectra of the received signal.



**Figure 11. Rectangular pulse in time and in frequency.**

In this scenario we used a symbol rate of 250 kbaud and two different carrier frequencies: 500 kHz and 750 kHz. With first carrier frequency the signal has less space in the frequency spectrum,

obviously because the carrier frequency is lower, but we will see that there are only some differences between each carrier frequency. We tested this for 4-, 16- and 64-QAM and changing the distance from the transmitter to the receiver from 50 cm to 100 cm by steps of 10 cm (always with an angle of  $0^\circ$  between them, i.e. perfectly aligned).

In the following, all the results we present will have the same (or almost the same) structure as this one. Each column represents one channel (one polarization) and each row represents one order of the QAM modulation, in this case 4, 16 and 64-QAM. Then, in each chart, we have two lines with different colors, each one representing a different carrier frequency, in this particular case the blue one represents 500 kHz and the orange one 750 kHz.

First, we are going to see the results of the EVM for each polarization. Looking at Fig. 12 we can see that, more or less as we predicted, both carrier frequencies behave in the same way, achieving the same EVM percentages. Although, we can see that as we increase the order of the QAM, the bigger carrier frequency (750 kHz) behaves better than the small one (500 kHz). There is no big difference ( $\pm 2\%$  in the worst cases) but still noticeable. Also, we can notice that the channel number 2 has worse quality than the channel number 1. This will occur during all our measurements because the second channel is much noisier than the first one, due to the nature of our scheme since Channel 2 signal has to reflect two times during its path to the receiver, while Channel 1 signal goes in a straight line which creates this difference between both channels. For example, in 64-QAM for Channel 1 we have 3.7% for both carriers but Channel 2 already has 6% for 500 kHz of carrier frequency and 8% for 750 kHz of carrier frequency.

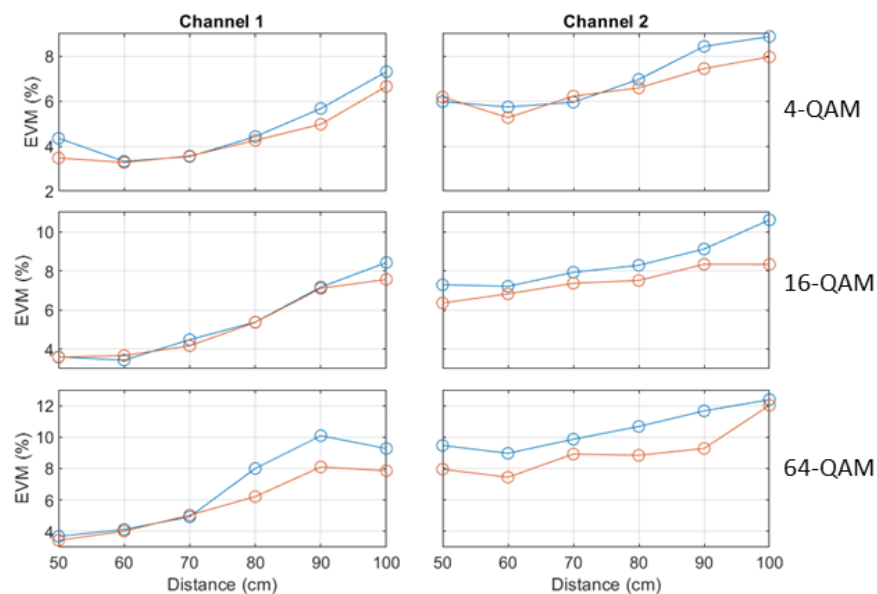


Figure 12. EVM for rectangular pulse and 250 kbaud of symbol rate.

In Fig. 13, we can see the results regarding the BER of our system. We can notice that for 4-QAM both channels and carrier frequency have the same BER of 0, but as we increase the order of the modulation, we can see that it increases. We have to say that as we sent 20000 symbols, which means 40000 bits for 4-QAM, 80000 bits for 16-QAM and 120000 bits for 64-QAM, we cannot assure BER values under 1 divided by this number of bits, which would be  $5 \cdot 10^{-5}$  for 4-QAM,  $2.5 \cdot 10^{-5}$  for 16-QAM and  $8.333 \cdot 10^{-6}$  for 64-QAM. But as these values are under our threshold of good quality ( $3.8 \cdot 10^{-3}$ ), we can work with them. For 16-QAM, in the Channel 1 it is almost always 0 but it increases a bit in 100cm for 500 kHz. In the Channel 2 we have some

strange values for 60 and 70 cm because it does not make sense to have worse BER there than in 80 cm, but probably it was due to some error in the measurement and finally at 100 cm for 500 kHz we see that it increases to  $2 * 10^{-4}$ . Finally, for 64-QAM we can notice that the BER degrades faster and in a more noticeable way, achieving very high values in high distances. As we expected, it degrades faster in Channel 2 than in Channel 1.

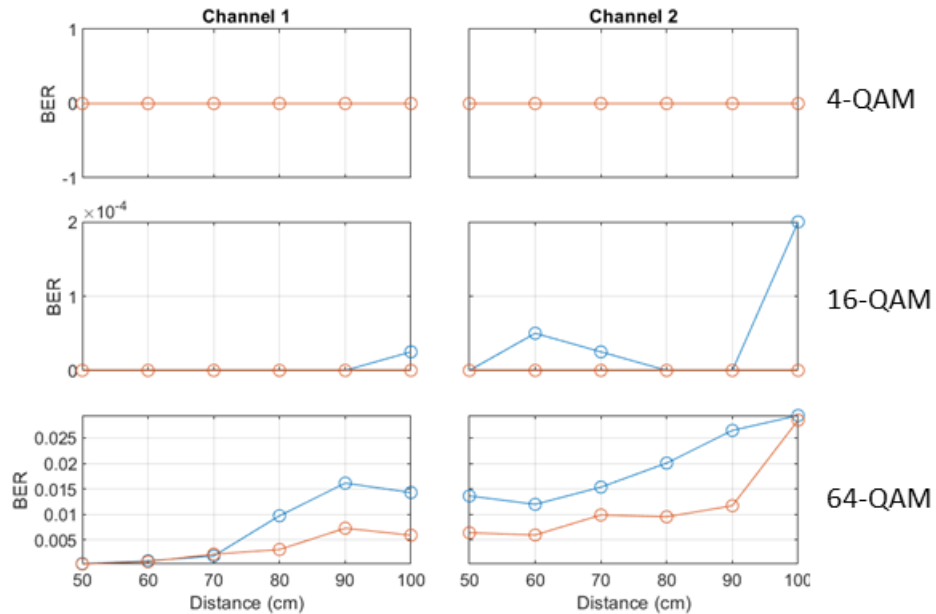


Figure 13. BER for rectangular pulse and 250 kbaud of symbol rate.

In Fig. 14, we show how SNR changes when we increase the distance. As we can see, both carrier frequencies behave almost in the same way. For each one of the three different modulation orders (4, 16 and 64) we notice that it produces a linear descendent line. It makes sense that as we increase the distance, the SNR diminishes around 10 dB for each order of QAM. Also, as we increase the modulation order, the SNR values are a little bit lower (worse). For example, for 4-QAM, we have a value of 45 dB at the beginning, but for 16-QAM and 64-QAM we have 42.5 dB and 41dB.

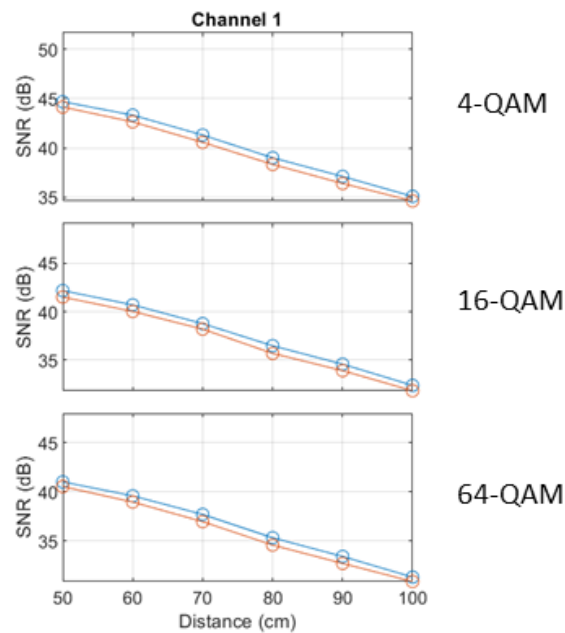


Figure 14. SNR for rectangular pulse and 250 kbaud of symbol rate.

Now, we should talk about different constellations we made up from our received data. In Fig. 15 we have the constellation for 4-QAM at 50 cm using a carrier frequency of 500 kHz for both channels. In this case we have an EVM of 4.35% for Channel 1 and 6% for Channel 2. Channel 1 is on the left and Channel 2 on the right, as we will keep doing in every comparison for the sake of simplicity. We can see, as mentioned above, that under same conditions we can appreciate some degradation in the constellation of the Channel 2 in respect to the Channel 1.

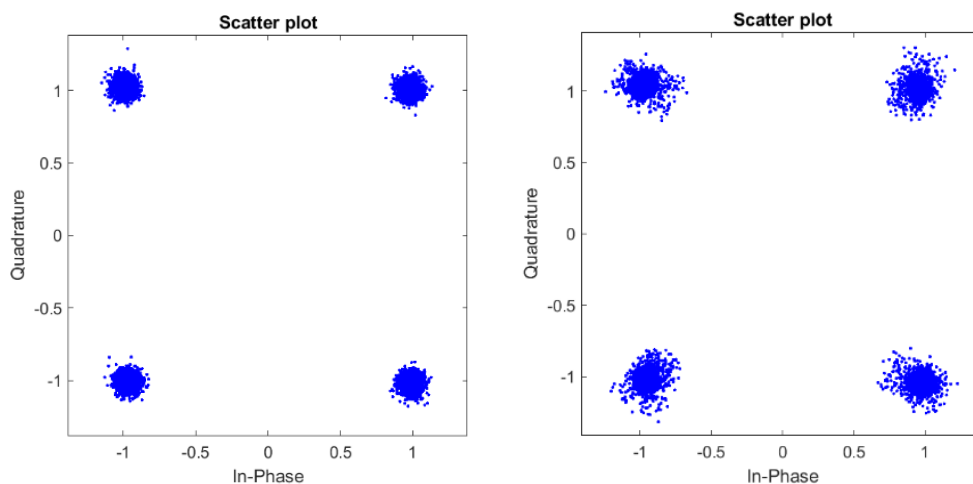
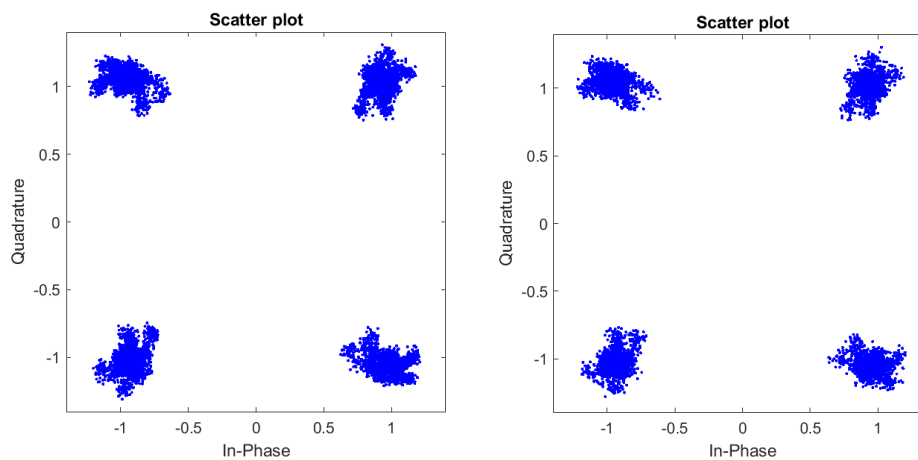


Figure 15. Constellation for rectangular pulse and 250 kbaud of symbol rate with 4-QAM, at 50 cm and a carrier frequency of 500kHz.

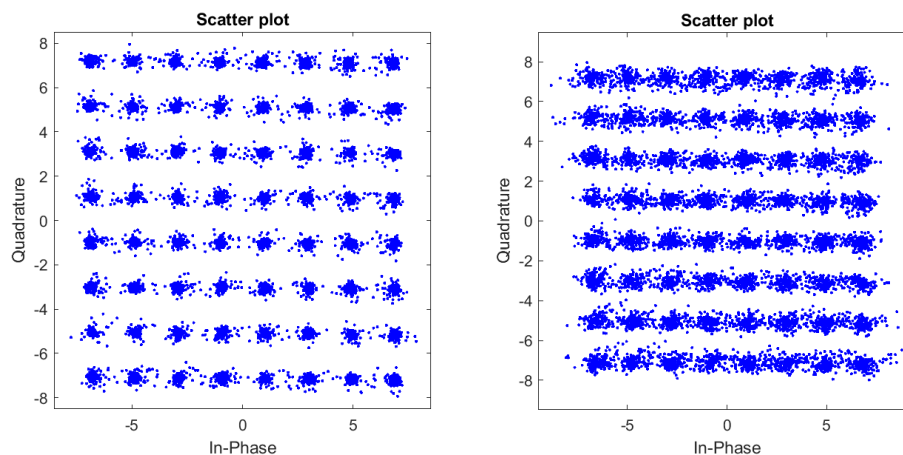
Here in Fig. 16, we can see the effect of increasing the distance. Now the transmitter was 100cm apart from the receiver, while in the other one was only 50cm, so we have doubled the distance and EVM has changed to 7.32% for Channel 1 and 8.88% for Channel 2. We can see that the constellation is degraded in respect to 50 cm but, 4-QAM signal can be still demodulated easily.

For higher order modulations this degradation would be fatal and BER would get bigger fast (as we saw in the BER chart).



**Figure 16. Constellation for rectangular pulse and 250 kbaud of symbol rate with 4-QAM, at 100 cm and a carrier frequency of 500kHz.**

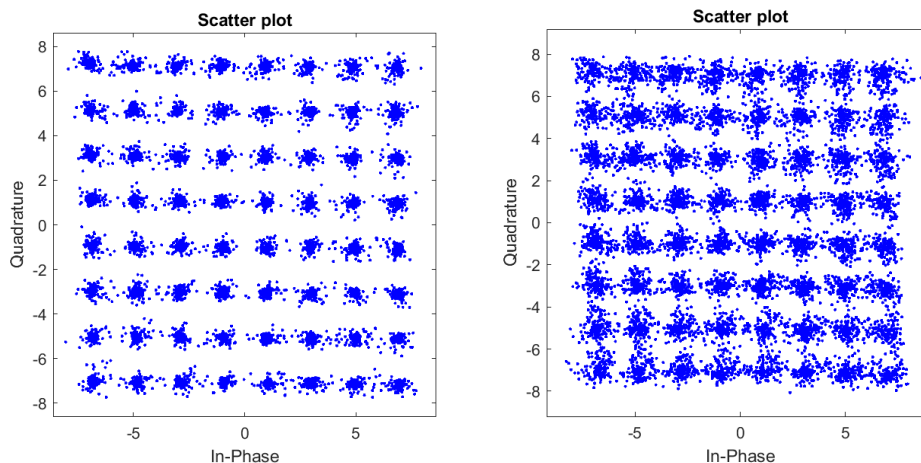
As we make bigger the order of the modulation, the fact of the Channel 2 being noisier than the Channel 1 gets more relevant and we can see it just by looking at Fig. 17. EVM for Channel 1 is 5% and for Channel 2 9.86%. The Channel 2 constellation looks much worse than Channel 1, as we could expect from the EVM values.



**Figure 17. Constellation for rectangular pulse and 250 kbaud of symbol rate with 64-QAM, at 70 cm and a carrier frequency of 500kHz.**

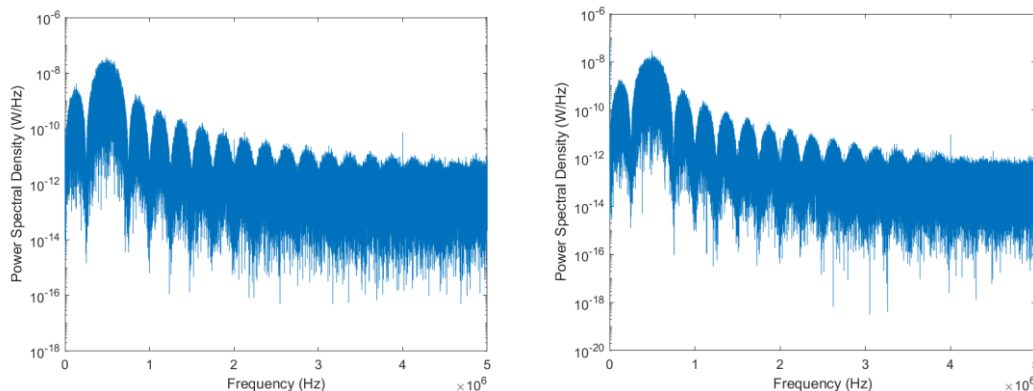
Finally, in Fig. 18 we have the same constellations as in Fig. 17 but for a carrier frequency of 750 kHz instead of 500 kHz. Here EVM is 5% for Channel 1 and 9% for Channel 2, what is very similar to the last constellation. As we said previously in this chapter, we expected no big difference between both carriers and as we can see if we compare EVM values and these two constellations with the last two ones, there is no big difference.





**Figure 18. Constellation for rectangular pulse and 250 kbaud of symbol rate with 64-QAM, at 70 cm and a carrier frequency of 750kHz.**

Moreover, the electrical spectra of the received signals were measured. In Fig. 19, we can see the effect that the rectangular pulse has in the frequential dominium. We can notice all the lobes created by the sinc. Also, we can see that the principal lobe is centered around 500 kHz and has approximately an extension of 500 kHz (from 250 kHz to 750 kHz) what corresponds exactly to the carrier frequency of 500 kHz and the symbol rate of 250 kbaud ( $250 * 2 = 500$ ) respectively,



**Figure 19. Electrical spectrum for rectangular pulse and 250 kbaud of symbol rate with 4-QAM, at 100 cm and a carrier frequency of 500kHz.**

### 4.3 Shaped pulse

Now, we go to the second scenario of our measurements. Here, we tried to improve somehow our system's performance, because in the previous one when we tried to send data with higher symbol rate (for example 500 kbaud) our system stopped decoding properly the received data. So, the first thing we thought about was to change the pulse shape from a rectangular pulse to a shaped pulse. With this change, we would have a rectangular shape in the frequency range instead of this sinc-shape frequency response we had with the rectangular pulse. The shaped pulse we chose was a raised cosine, using the `comm.RaisedCosineTransmitFilter` object from MATLAB Communications Toolbox.

In Fig. 20, we can see how a raised cosine filter looks in frequency with the different roll-off parameter values. In our case we chose a roll-off factor of 0.4. Later in this section we will see how this affects our frequency spectrum, making it much less noisy.

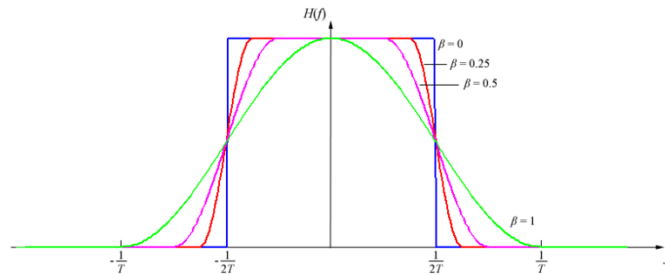


Figure 20. Raised cosine filter.

On the other hand, we added a symbol synchronizer and a carrier synchronizer to make an automatic synchronization with the received signal to be able to demodulate it properly. We used the objects `comm.SymbolSynchronizer` and `comm.CarrierSynchronizer` from the MATLAB Communication Toolbox.

First, we tried to use the same symbol rate as in the previous scenario (250 kbaud) but with smaller carrier frequencies of 250 kHz and 500 kHz. We measured this for 4-, 16- and 64-QAM and for distances from 50 cm to 100 cm with steps of 10 cm. Herein this case, we were able to also measure some angle changes for the distance of 50, from  $-10^\circ$  to  $10^\circ$  with steps of  $5^\circ$ .

In Fig. 21, the blue line is for a carrier frequency of 250 kHz and the orange line for 500 kHz. We can see that for a carrier of 250 kHz behave similarly except for 64-QAM but probably that was an error in the measurement rather than a real difference. For a carrier frequency of 500 kHz Channel 1 performs better than Channel 2 except for 64-QAM where they are very similar. At 50cm for 4-QAM for Channel 1 we have 5%, but for Channel 2 we obtain 7.6%. Compared to the last scenario, we didn't improve our results so then we will try the same but with 500 kbaud of symbol rate.

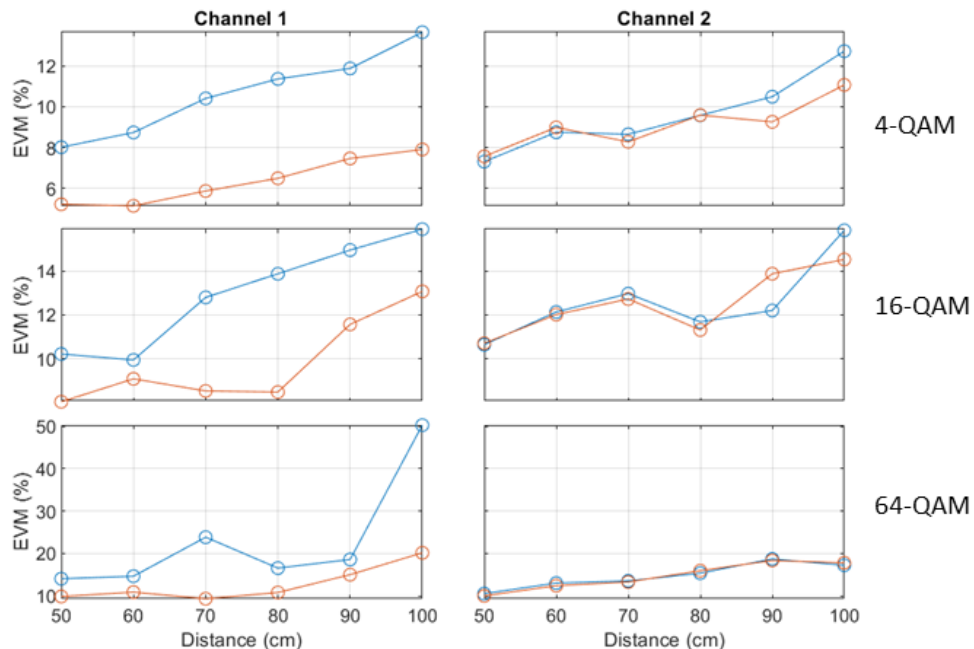


Figure 21. EVM for shaped pulse and 250 kbaud of symbol rate.

When we look at the results regarding BER (Fig. 22), we see that both channels behave different from each other. In Channel 1 the carrier frequency of 500 kHz always achieves better results than 250 kHz, while in Channel 2 there is no carrier frequency better than the other. For 4-QAM we still observe some margin of acceptable performance, but 16-QAM and 64-QAM do not allow the correct transmission. In 4-QAM we always stay below  $1.5 \times 10^{-3}$  which is good. Again, we underperform if we compare this to the previous scenario.

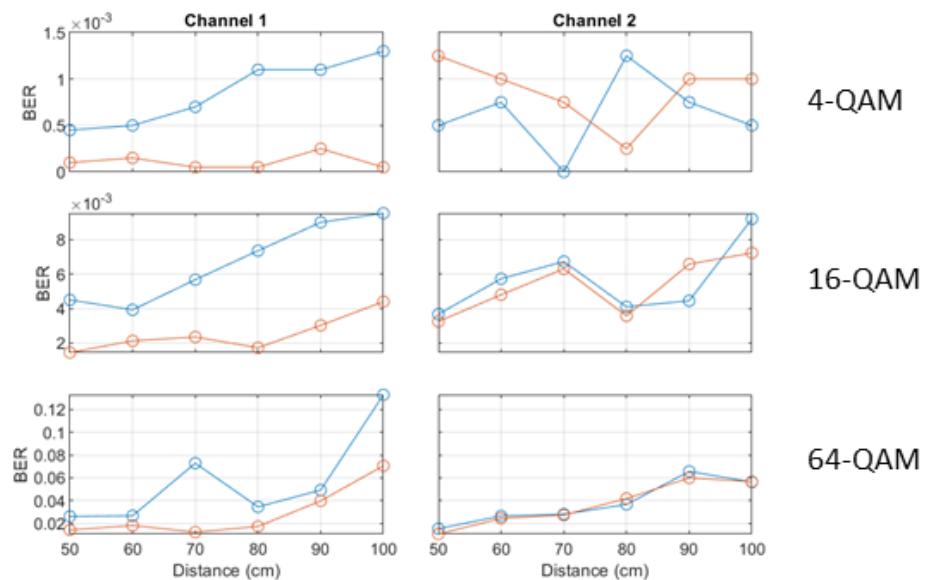


Figure 22. BER for shaped pulse and 250 kbaud of symbol rate.

In Fig. 23, we have a graph that represents how SNR changes when we increase the distance. As we can see, both carrier frequencies behave almost in the same way. For each one of the three different modulation orders (4-, 16- and 64-) we notice that it produces a, more or less, linear descendent line. It makes sense that as we increase the distance, the SNR decreases. Also, as we increase the modulation order, the SNR values keep very similar, except from the 4-QAM one, where we have a little bit better SNR. At 50 cm we go from 42.5 dB for 4-QAM to 41 and 40 dB for 16- and 64-QAM.

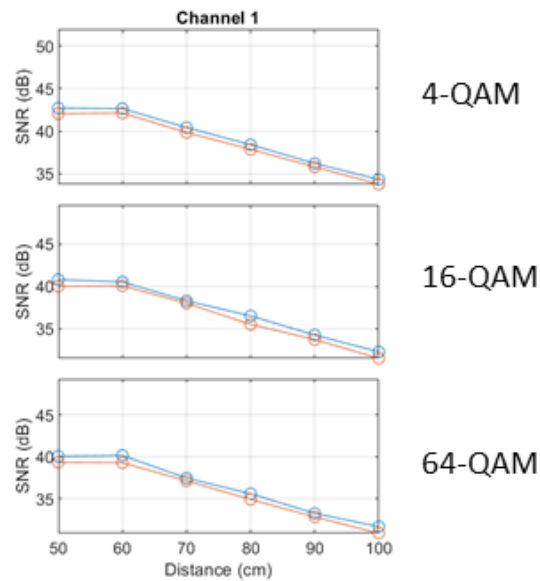


Figure 23. SNR for shaped pulse and 250 kbaud of symbol rate.

If we look to the constellation belonging to the 4-QAM at 50 cm and  $0^\circ$  (Fig. 24), we can see how much noisier it is if we compare it with the rectangular pulse scenario (Fig. 15). Here we obtain EVM values of 8% for Channel 1 and 7.3% for Channel 2. And if we increase the distance or the order of modulation, we see how heavily it degrades. This is what we expected regarding the results we obtained.

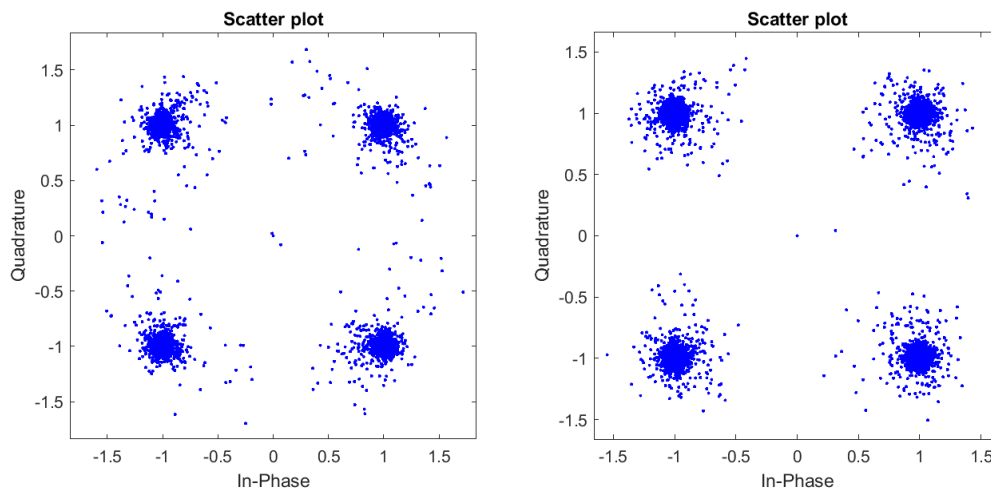
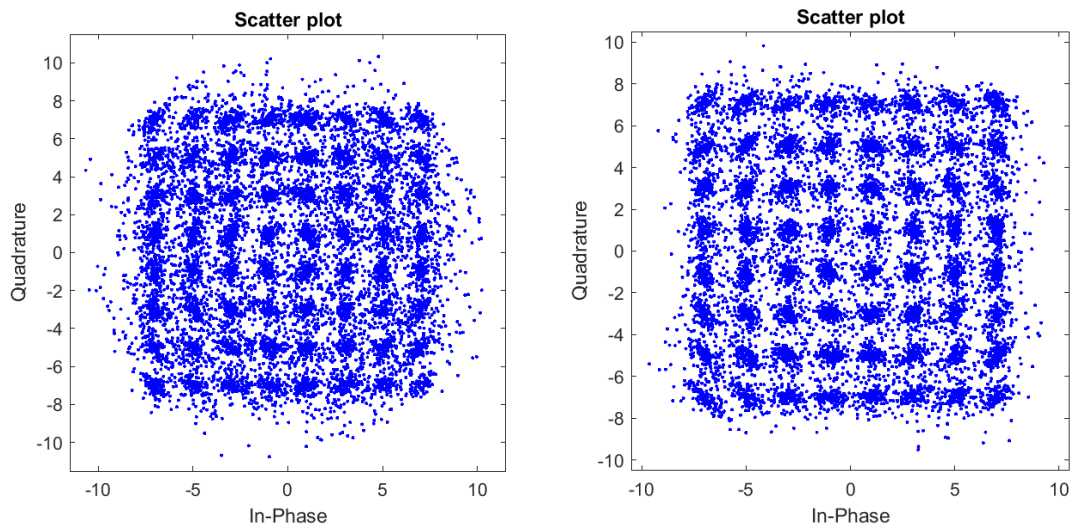


Figure 24. Constellation for shaped pulse and 250 kbaud of symbol rate with 4-QAM, at 50 cm,  $0^\circ$  and a carrier frequency of 250 kHz.

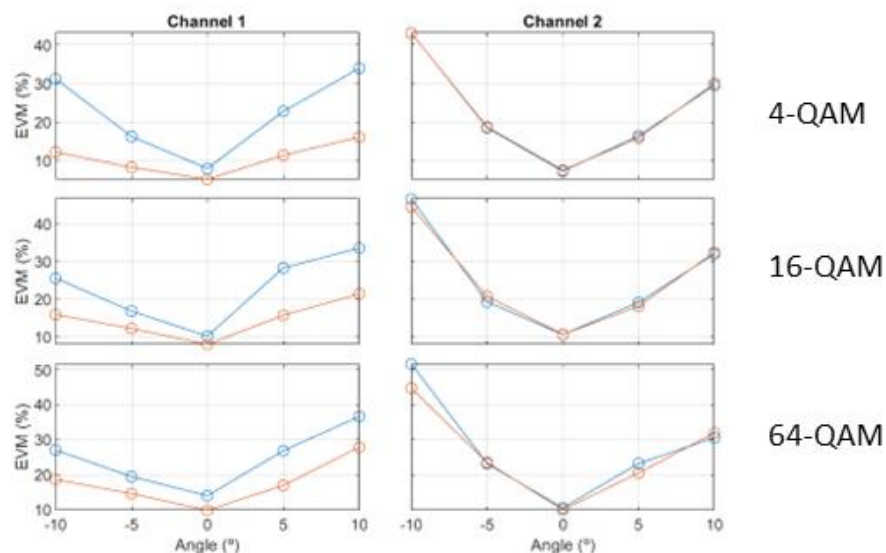
Looking to Fig. 25, we can see how the constellation degrades when we have a bigger order and more distance with respect to Fig. 24. Now, the EVM values are 23.8% for Channel 1 and 13.5% for Channel 2. The degradation here, is much stronger than the one we had in the previous scenario of measure (Fig. 17).



**Figure 25. Constellation for shaped pulse and 250 kbaud of symbol rate with 64-QAM, at 70 cm, 0° and a carrier frequency of 250 kHz.**

On the other hand, now we can try to see how the effect of changing the angle affects our system. In this type of system, it should be critical, so the change in angle would have to affect much more heavily than the change in distance.

In Fig. 26, we can notice what is happening in the last charts: for Channel 1 the bigger carrier frequency (500 kHz) is always better than the small one (250 kHz), but for Channel 2 they both perform the same. This can be because noise affects much more Channel 2 so both carrier frequencies perform 'as bad'. Also, we can see that the change in the angle makes very big changes in our system performance, only Channel 1 with 4- or 16-QAM and 500 kHz could be still somehow feasible in the central parts (around 5° of deviation). In we make bigger the angle deviation we obtain too high EVM values, such as more than 40% for Channel 2 at -10° and 4-QAM.



**Figure 26. EVM in terms if angle for shaped pulse and 250 kbaud of symbol rate.**

If we look at the BER (Fig. 27), we see exactly the same behavior that with EVM. For Channel 1 500 kHz performs better than 250 kHz and for Channel 2 both perform in the same way. Looking

at the values we can agree that the only times the system would be usable would be using 500 kHz for 4-QAM, the rest are not functional in the whole range of angles. Also, we can observe that when we change to positive angles Channel 1 degrades more than Channel 2, and the opposite thing when we change to negative angles. For example, at  $10^\circ$  and 4-QAM we obtain a BER of  $3.9 \times 10^{-3}$  for 250 kHz of carrier frequency for Channel 1 and with the same conditions we obtain  $2.5 \times 10^{-4}$  for Channel 2.

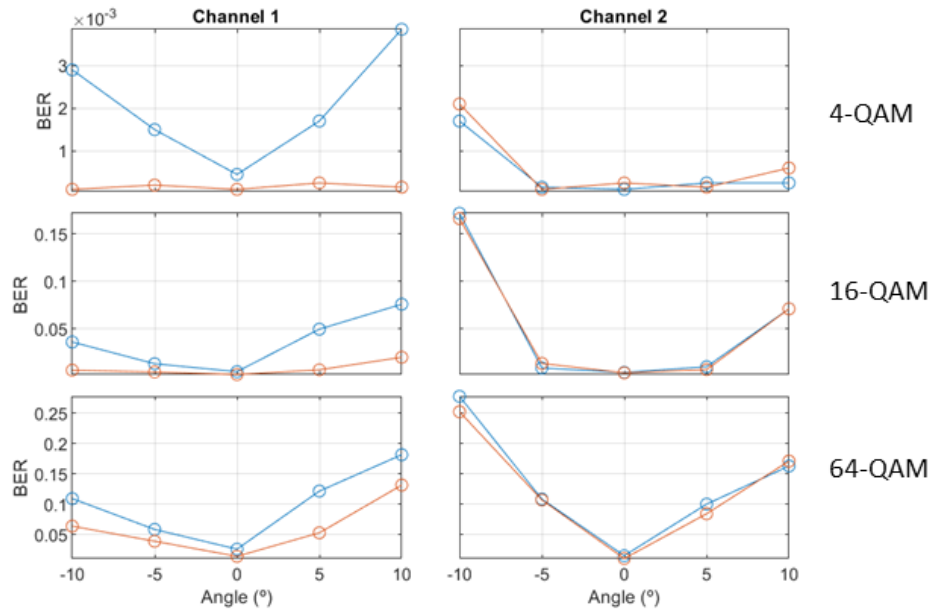


Figure 27. BER in terms if angle for shaped pulse and 250 kbaud of symbol rate.

In Fig. 28, we have a graph that represents how SNR changes when we modify the angle. As we can see, both carrier frequencies behave almost in the same way. For each one of the three different modulation orders (4-, 16- and 64-) we notice that it produces a triangular-shaped line with a maximum at  $0^\circ$ . It makes sense that as we increase the angle in both directions, the SNR decreases. Also, as we increase the modulation order, the SNR values keep very similar, except from the 4-QAM one, where slightly better SNR is obtained. For 4-QAM at  $0^\circ$  we obtain 42.5 dB and for 16-QAM and 64-QAM 40 dB and 39 dB. Also, it is noticeable that SNR decreases more when we move towards positive angles than to negative ones.

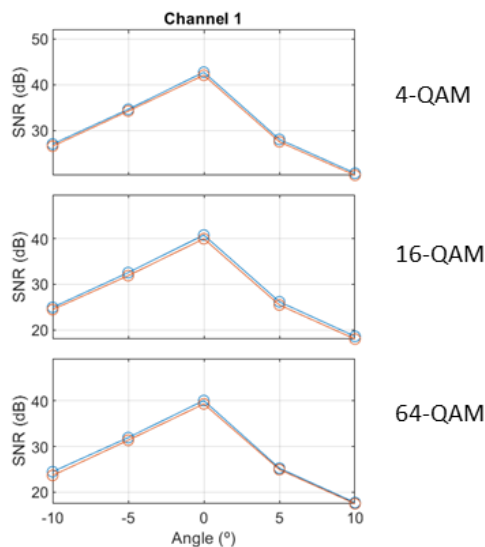
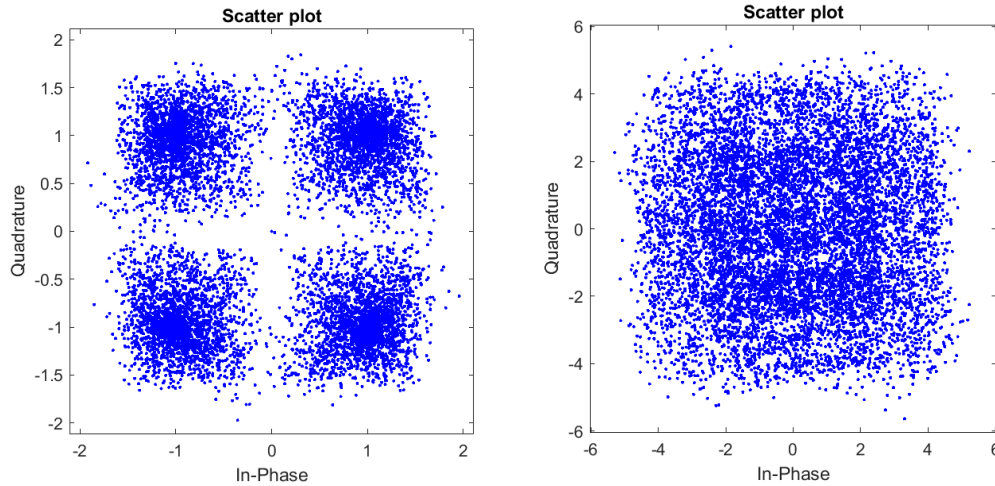


Figure 28. SNR in terms if angle for shaped pulse and 250 kbaud of symbol rate.

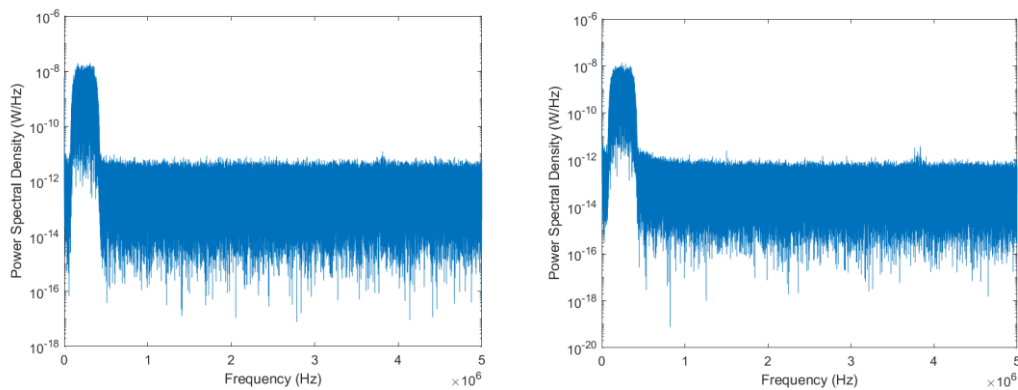
When we look to Fig. 29 we can see how much, changing from  $0^\circ$  to  $-10^\circ$  affects the performance of our system, because Channel 1 constellation has become very noisy, and Channel 2 is completely unrecognizable. Here, we have EVM values of 31.2% for Channel 1 and 43% for Channel 2, which are very big values.



**Figure 29. Constellation for shaped pulse and 250 kbaud of symbol rate with 4-QAM, at 50 cm,  $-10^\circ$  and a carrier frequency of 250 kHz.**

Finally, we have to look how the change from rectangular pulse to raised cosine affected our frequency spectrum.

In Fig. 30, we can see that we have cleaned a lot our frequency spectrum, because now we have got rid of all the secondary lobes we had when we were using the rectangular pulse. Now we can see only one lobe, kind of rectangular-shaped. If we checked its span is almost 500 kHz, which is what we expected because we are using a symbol rate of 250 kbaud ( $2 \times 250 = 500$ ), and we also can see how it is centered around 250 kHz, our carrier frequency.



**Figure 30. Electrical spectrum for shaped pulse and 250kbaud of symbol rate with 4-QAM, at 100 cm,  $0^\circ$  and a carrier frequency of 250 kHz.**

Then, we have the same approach but this time using a bigger symbol rate of 500 kbaud. Here we made our measurements for carrier frequencies of 500 kHz and 750 kHz. Also, as in every past scenario we made it for 4-, 16- and 64-QAM and changing the distance from 50 cm to 100 cm with steps of 10 cm and for a distance of 50 cm changing the angle from  $-10^\circ$  to  $10^\circ$  with steps of  $5^\circ$ .

Regarding Fig. 31, we can see that our system behaves worse than with 250 kbaud of symbol rate, what is normal because we are pushing our system further from its limits of good behavior. We

can notice that, as in the EVM for 250 kbaud (Fig. 21), for Channel 2 both carrier frequencies behave the same, but for Channel 1 one behaves better than the other. The special thing about this is that this time the lower carrier frequency behaves better than the bigger one. Our system works better with 500 kHz of carrier than with 750 kHz. For example, for 4-QAM in Channel 1 at 50 cm we obtain a value of 9.45% for 500 kHz of carrier frequency and 13.12% for 750 kHz of carrier frequency. So, combined with the result of our previous EVM chart we can say that maybe the carrier frequency where our system behaves the best is at 500 kHz.

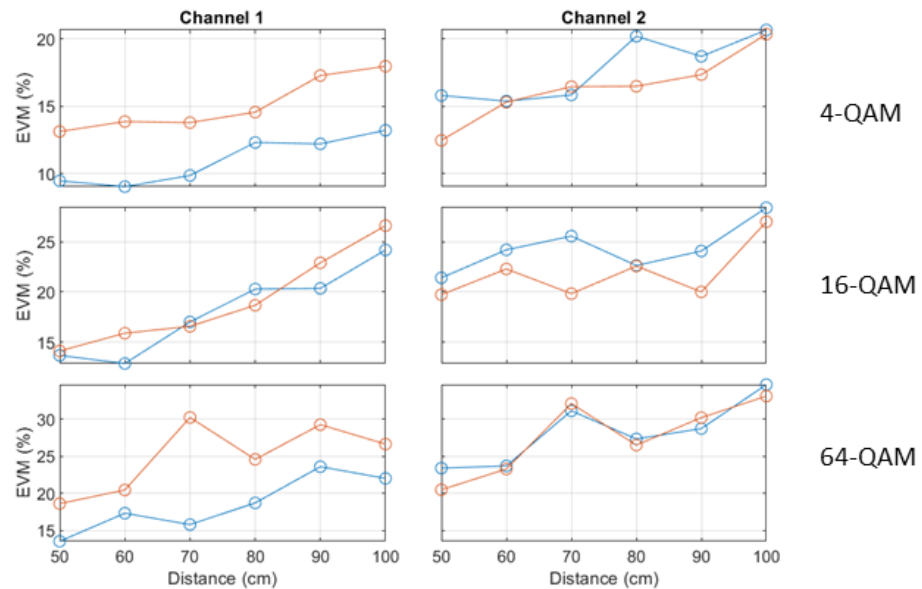


Figure 31. EVM for shaped pulse and 500 kbaud of symbol rate.

In Fig. 32, with BER we notice the same facts as with EVM: for Channel 2 both carriers behave the same and for Channel 1 500 kHz works slightly better than 750 kHz, but this time the difference is much more subtle. In Channel 1, for 16-QAM both carriers behave quite similarly, because the maximum difference for them is at 100 cm, where 500 kHz has a value of  $3.3 \times 10^{-2}$  and 750 kHz  $4 \times 10^{-2}$ . However, both values are bad (more than the threshold of  $3.8 \times 10^{-3}$ ) so we do not really care much about that difference. Also, as happened with EVM, we see that the system works worse than with 250 kbaud of symbol rate.



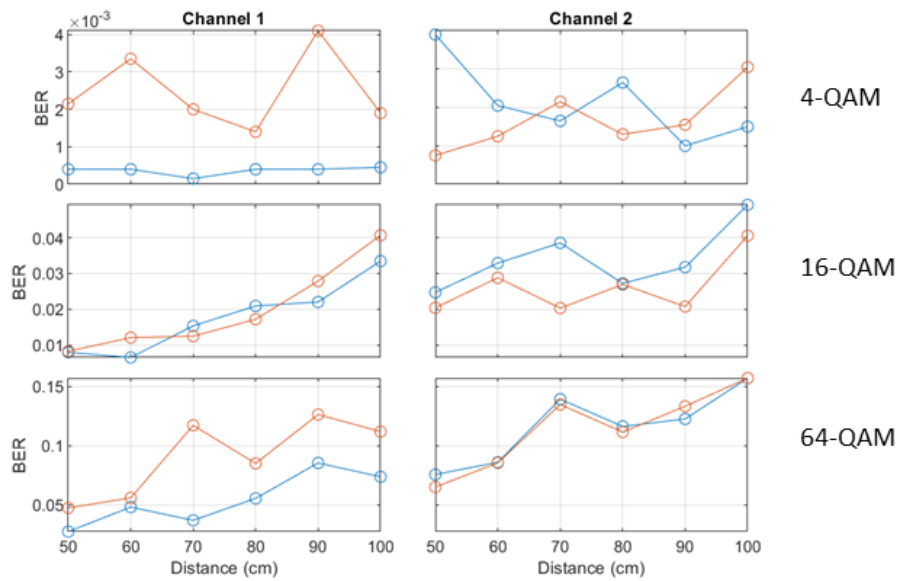


Figure 32. BER for shaped pulse and 500 kbaud of symbol rate.

In Fig. 33, we have a graph that represents how SNR changes when we increase the distance. We can see that it is extremely similar for both symbol rates, but this one has a little bit worse value. As we can see, both carrier frequencies behave almost in the same way. For each one of the three different modulation orders (4-, 16- and 64-) we notice that it produces a, more or less, linear descendent line. It makes sense that as we increase the distance, the SNR decreases. Also, as we increase the modulation order, the SNR values keep very similar, except from the 4-QAM one, where we a little bit better SNR. For example, for 4-QAM at 50 cm we measure a value of almost 40 dB and for 16- and 64-QAM we have 36.1 dB.

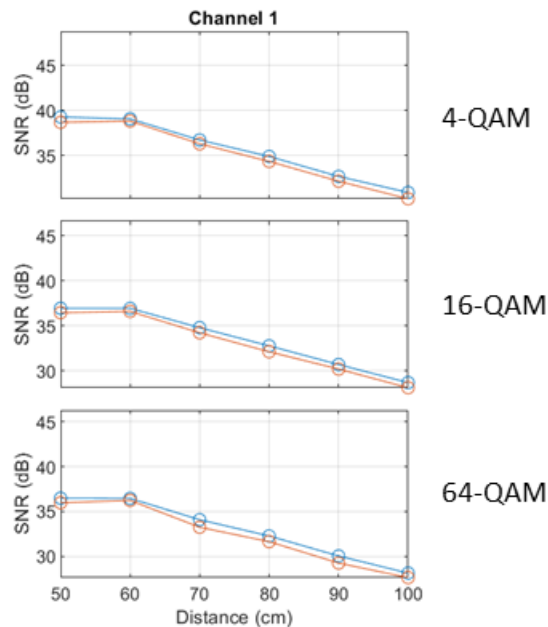
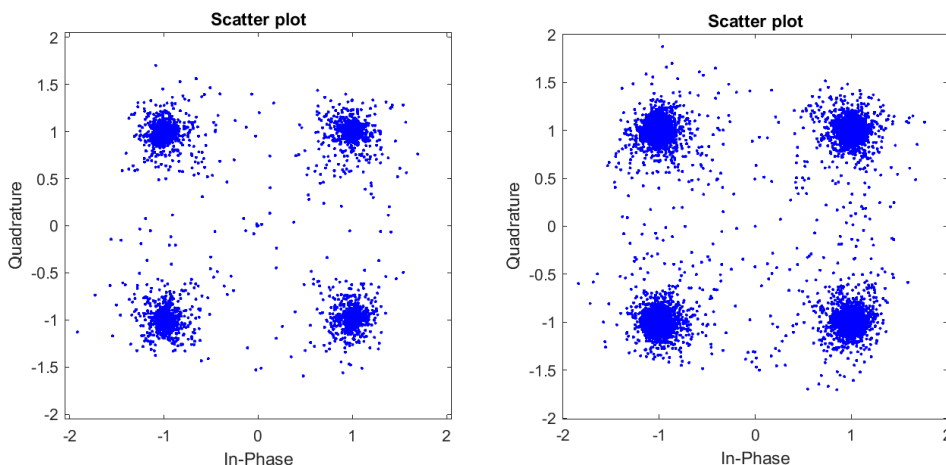


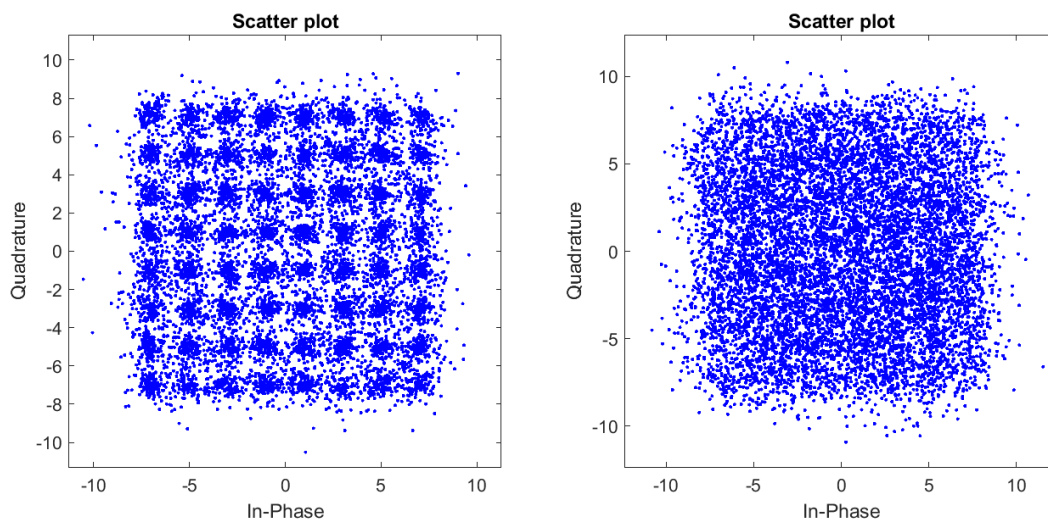
Figure 33. SNR for shaped pulse and 500 kbaud of symbol rate.

If we look to Fig. 34 obtained when using 500 kbaud of symbol rate, 500 kHz of carrier frequency, 4-QAM, 50cm and an angle of  $0^\circ$ , we can see that there is not much difference between this one and the same one from 250 kbaud of symbol rate. Here we have obtained EVM values of 9.45% for Channel 1 and 15.8% for Channel 2. This one is a little bit noisier than the other one, what is normal if we compare EVM values. Also, obviously they look worse than the ones from the rectangular shape pulse scenario.



**Figure 34.** Constellation for shaped pulse and 500 kbaud of symbol rate with 4-QAM, at 50 cm,  $0^\circ$  and a carrier frequency of 500 kHz.

In Fig. 35, we have the constellation for 64-QAM, 70 cm and  $0^\circ$ . We have augmented the distance and the order of the modulation if we compare it with the last one. The EVM values have increased to 15.78% for Channel 1 and 31.1% for Channel 2. As expected, we see a degradation in the constellation, but for Channel 2 the degradation is massive, we can't distinguish anything from it. In this same scenario but for 250 kbaud of symbol rate, we could still see something in the Channel 2's constellation, but here is much worse. We could expect this, as we saw in the EVM and BER charts (Fig. 31 and 32), where the system in this scenario worked worse than for 250 kbaud.



**Figure 35.** Constellation for shaped pulse and 500 kbaud of symbol rate with 64-QAM, at 70 cm,  $0^\circ$  and a carrier frequency of 500 kHz.

On the other hand, we have to check how the system is behaving with this new symbol rate when we modify the angle between the transmitter and the receiver, just as we did with 250 kbaud.

Fig. 36 is the EVM chart with respect to the angle. We see the same patron as we saw of Channel 2 behaving the same for both carrier frequencies and Channel 1 behaving better for 500 kHz than for 750 kHz. Also, this time the system is also working worse than for 250 kbaud. But this is pretty much what we expected, because if the system worked worse when we changed the distance, we should expect it to work also worse when we change the angle that is a much critical parameter. Subtle changes in the angles make the system totally unfeasible. For example, for Channel 2 in 4-QAM at  $-10^\circ$  we obtain values of EVM around 47%, which is too big.

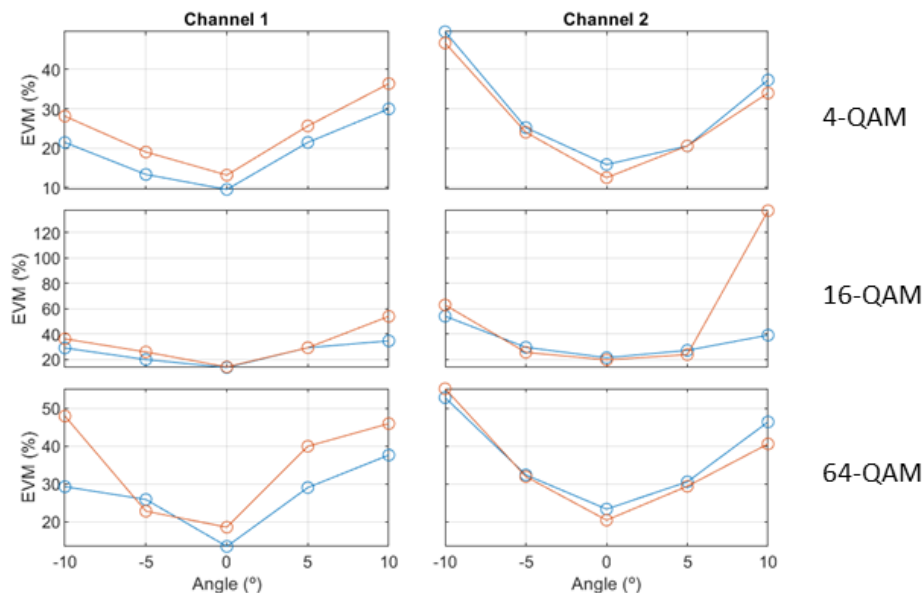


Figure 36. EVM in terms of angle for shaped pulse and 500 kbaud of symbol rate.

The same happens when we see BER with respect to the angle (Fig. 37). The same situation that happened with EVM with respect to distance, BER with respect to distance and EVM with respect to angle happens here. For example, we reach values of  $2.5 \times 10^{-2}$  for Channel 2 in 4-QAM at  $-10^\circ$ .

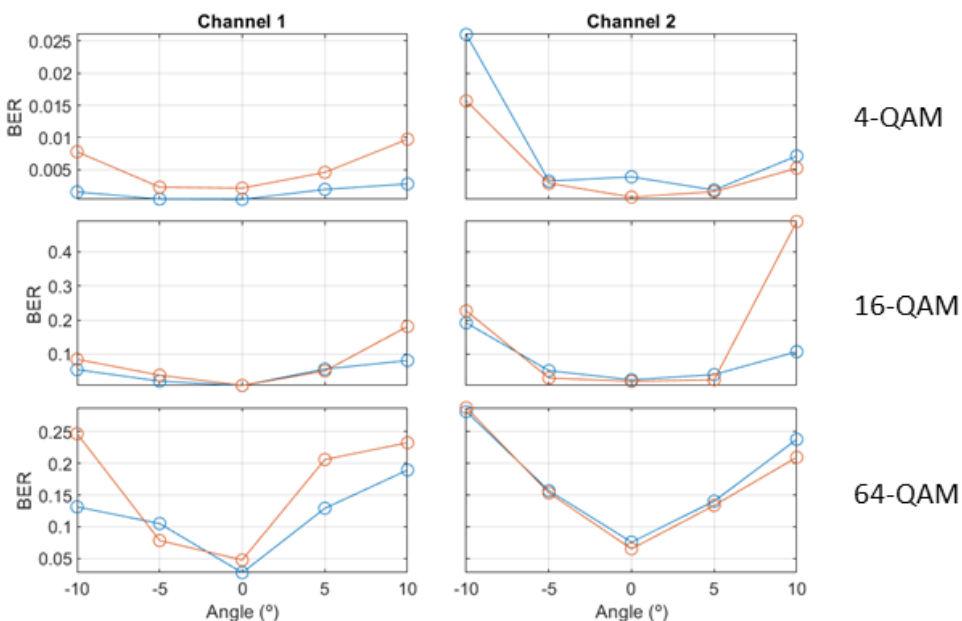


Figure 37. BER in terms of angle for shaped pulse and 500 kbaud of symbol rate.

In Fig. 38, we have a graph that represents how SNR changes when we modify the angle. It is extremely similar to the one for the other symbol rate. As we can see, both carrier frequencies behave almost in the same way. For each one of the three different modulation orders (4-, 16- and 64-) we notice that it produces a triangular-shaped line with a maximum at  $0^\circ$ . It makes sense that as we increase the angle in both directions, the SNR decreases. Also, as we increase the modulation order, the SNR values keep very similar, except from the 4-QAM one, where we have a little bit better SNR. At  $0^\circ$  for 4-QAM we obtain a value of almost 40 dB and for 16- and 64-QAM we obtain a value of 37.6 dB. Also, it is noticeable that SNR decreases more when we move to positive angles than to negative ones.

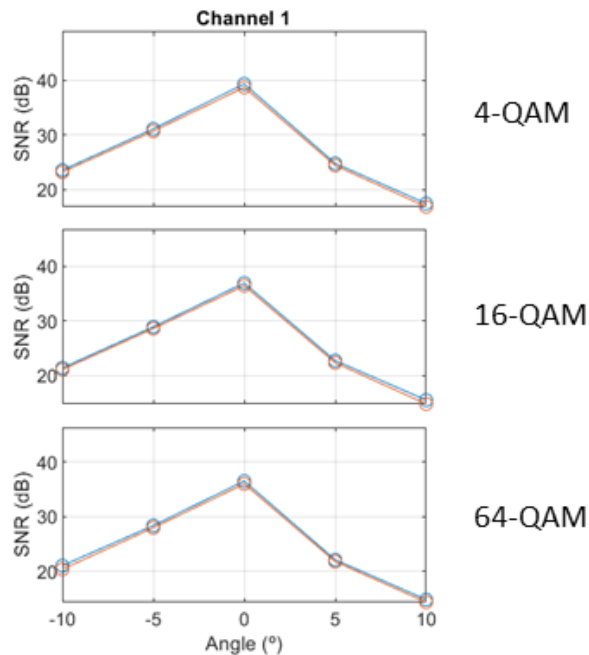
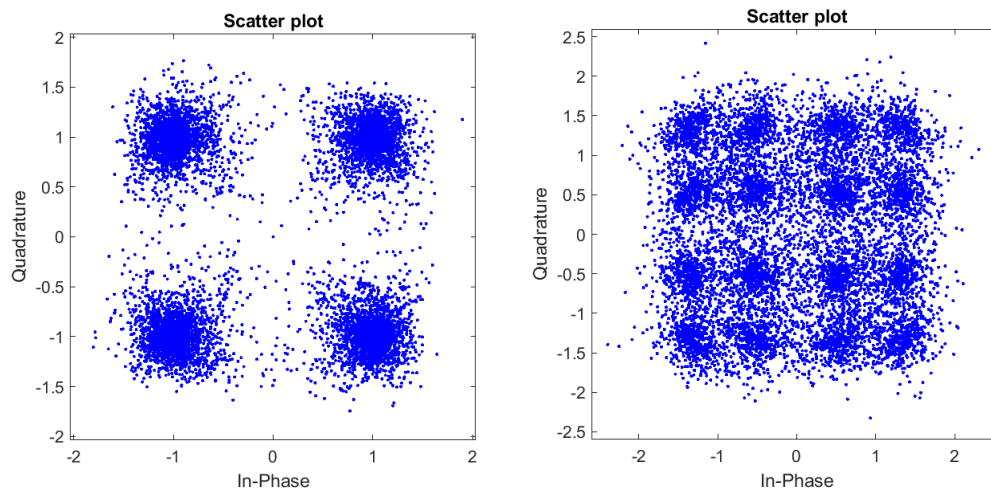


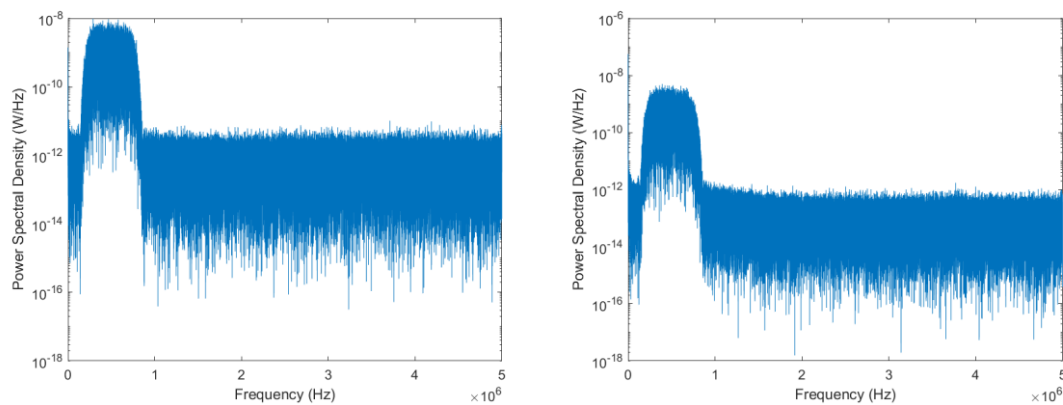
Figure 38. SNR in terms of angle for shaped pulse and 500 kbaud of symbol rate.

As we saw with the same constellations but for 250 kbaud of symbol rate (Fig. 29), Fig. 39 looks degraded if we compare them with the same one in  $0^\circ$  of angle deviation (Fig. 34), but surprisingly this time, Channel 2 constellation looks somehow better than for 250 kbaud. Here, we have EVM values of 30% for Channel 1 and 37.3% for Channel 2. We see a strange group of 4 circles where it should be one but, at least is better than the mess we had for 250 kbaud.



**Figure 39. Constellation for shaped pulse and 500 kbaud of symbol rate with 4-QAM, at 50 cm,  $-10^\circ$  and a carrier frequency of 500 kHz.**

Finally, in Fig. 40 we have the frequency spectra from the electrical signal from both channels. We see the flat spectrum that we wanted to achieve using the raised cosine, without secondary lobes. We notice that the main lobe spans approximately 500 kHz (from 250 kHz to 750 kHz) which corresponds to our symbol rate of 500 kbaud and also, we can see that this lobe is centered around 500 kHz which corresponds to our carrier frequency, in this case, of also 500 kHz.



**Figure 40. Electrical spectrum for shaped pulse and 500 kbaud of symbol rate with 4-QAM, at 100cm,  $0^\circ$  and a carrier frequency of 500 kHz.**

To conclude this section, we can say that we tried to figure out a way of improving our system and we thought about using the raised cosine to flatten our spectrum and thus made us able to use higher symbols rates like 500 kbaud, and consequently achieve higher speeds. But we notice that could work for 4-QAM, just like in rectangular pulse, but for 16-QAM we could only work in closer distances. However, we were able to work at 64-QAM at 50 cm for 250 kbaud of symbol rate and 500 kHz of carrier frequency, where we could achieve a higher speed than with rectangular shape.

#### 4.4 Shaped pulse with EQ

Finally, we reach the final scenario of our measurements. Our goal now is to make better the performance of our system while keeping the good spectral qualities of using a raised cosine pulse. We saw in the las scenario that we underperformed when we used this type of pulse with

respect when we used a rectangular pulse, so we need to try to figure out a way to make this performance better.

The solution we thought about was implementing an EQ that made us able to diminish BER and EVM of our received signals. Also, it was a pretty easy solution because we could implement it in MATLAB, so we just had to add it to our post-processing in MATLAB. We used `comm.LinearEqualizer`, an object from MATLAB's Communication Toolbox. We need to pass it three values: number of equalizer taps, reference tap and constellation. The last one, constellation, is quite easy, we only needed to pass the ideal constellation of the order of QAM we were using, so the equalizer knew what to look for. Then, the number of taps and the reference taps was mostly experimental, and we tested different values until we reached the ones that suited us the most. Those were 15 for number of taps and 8 for reference tap.

Then, when we used it, we needed to pass it the signal we wanted to equalize and some training values so the equalizer could learn how to equalize properly the signal. So, we created, apart from the signal, some training symbols for the equalizer. In this case we created 2000 training symbols and then the signal was 20000 symbols long.

So, now we can look at the results of modifying the angle between the transmitter and the receiver for 250 kbaud of symbol rate, just as we did in the previous scenario:

In Fig. 41, we can see how for 4-QAM are quite good for Channel 1, because the quality remains consistent if we change the angle, while for Channel 2 it changes heavier. For example, for 4-QAM at  $10^\circ$  for Channel 1 we obtain a value of 9% but for Channel 2 the value is 28%, what is a very big difference. For 16-QAM, we see a much bigger change because Channel 2 only is usable at  $0^\circ$ , and Channel 1 gets very degraded if we turn to positive angles. One special thing we can see is that both carrier frequencies (250 kHz and 500 kHz) behave almost the same for every case.

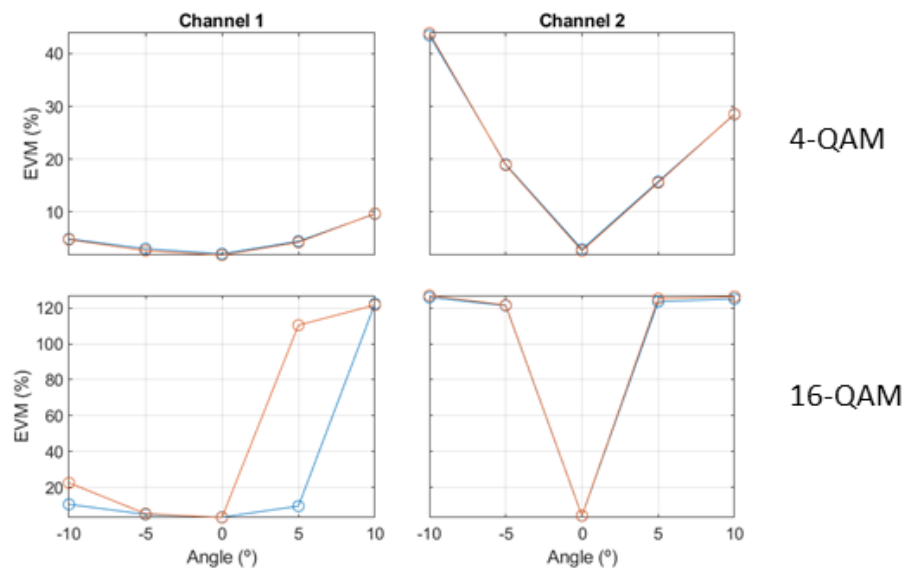


Figure 41. EVM in terms if angle for shaped pulse with EQ and 250 kbaud of symbol rate.

In terms of BER (Fig. 42) we can see almost the same behavior as with EVM: for 4-QAM the system works very well, the only exception would be Channel 2 at  $-10^\circ$  where it has value of  $3.2 * 10^{-3}$  which will be below the threshold ( $3.8 * 10^{-3}$ ), and then for 16-QAM both channel graphics are almost equal as EVM ones, where Channel 2, except at  $0^\circ$ , performs very bad and Channel 1 performs bad at positive angles. We see again how both carrier frequencies (250 kHz and 500 kHz) behave very similarly.

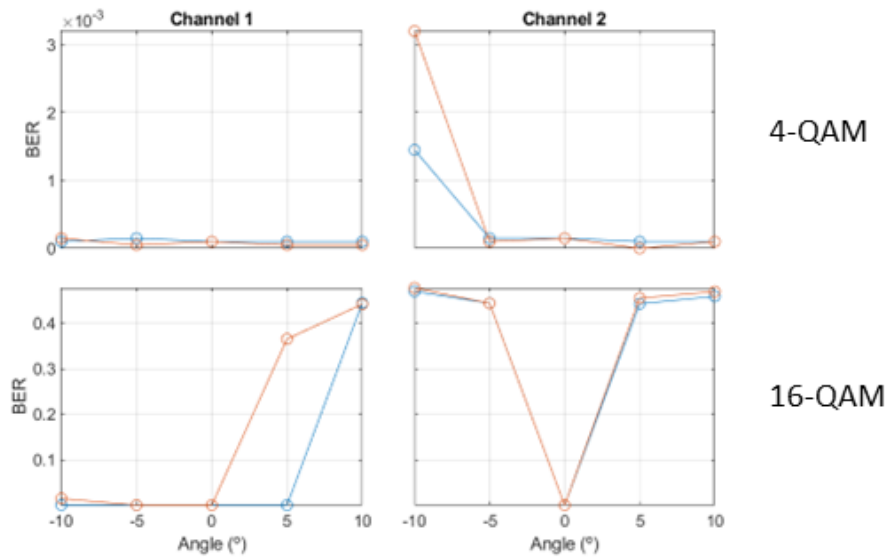


Figure 42. BER in terms if angle for shaped pulse with EQ and 250 kbaud of symbol rate.

In Fig. 43, we have a graph that represents how SNR changes when we modify the angle. We can see that the values we achieve here as the best ones from all the scenarios in terms of angle change. As we can see, both carrier frequencies behave almost in the same way. For both modulation orders (4 and 16) we notice that it produces a triangular-shaped line with a maximum at 0°. It makes sense that as we increase the angle in both directions, the SNR decreases. Also, as we increase the modulation order, the SNR values stay very similar, but the 4-QAM one has a little bit better SNR. At 0° for 4-QAM we have a value of 44.1 dB, and for 16-QAM we obtain a value OF 42.5 dB. Also, it is noticeable that SNR decreases more when we move to positive angles than to negative ones.

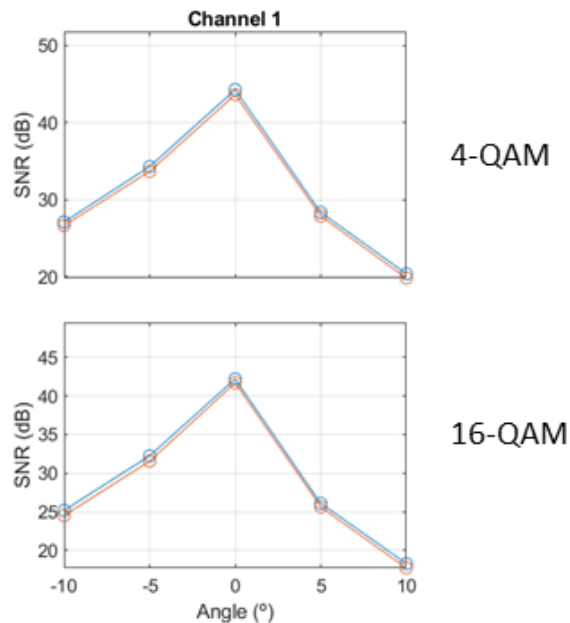
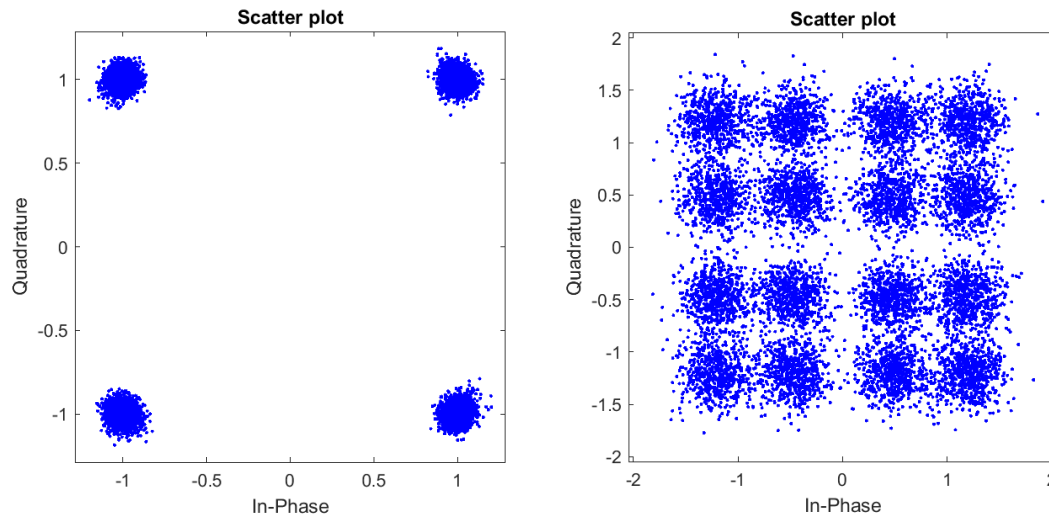


Figure 43. SNR in terms if angle for shaped pulse with EQ and 250 kbaud of symbol rate.

In Fig. 44, we have the constellations for 250 kbaud of symbol rate, carrier frequency of 250 kHz, 4-QAM, 50 cm and -10°. We can notice how good is the Channel 1 constellation. It is a very clean

constellation, and it is for  $-10^\circ$  of angle, which in the other scenarios would have been extremely noisy and barely recognizable. It is almost the same as the same constellation for  $0^\circ$  for rectangular shape. In other hand, Channel 2 constellation is much worse, we can see groups of four circles where there should be one. This creates a big EVM because almost every point is very far from where it ideally should be, but the disposition of this circles makes that still it is somehow able to decode the symbol because it is 4-QAM. That is why we have a quite good BER knowing how bad the EVM is. Here, for Channel 1 we obtain an EVM value of 4.87% while for Channel 2 we have 43.48%, what is a very big difference.



**Figure 44. Constellation for shaped pulse with EQ and 250 kbaud of symbol rate with 4-QAM, at 50 cm,  $-10^\circ$  and a carrier frequency of 250 kHz.**

Then, we have the same measurements but for 500 kbaud of symbol rate:

In Fig. 45, we have the EVM chart, and this time is not as good as with 250 kbaud. For 4-QAM the curves still have some reasonable shape; as further you go from  $0^\circ$  the more it degrades, but for 16-QAM it goes completely crazy and we have very bad EVM values for every angle, our system here does not work at all. We have to say that still, for 4-QAM in Channel 1 the EVM values are quite good (for example 5% at  $0^\circ$  or around 10% at  $-5^\circ$ ) and better than the previous scenario, so we have made an upgrade there.



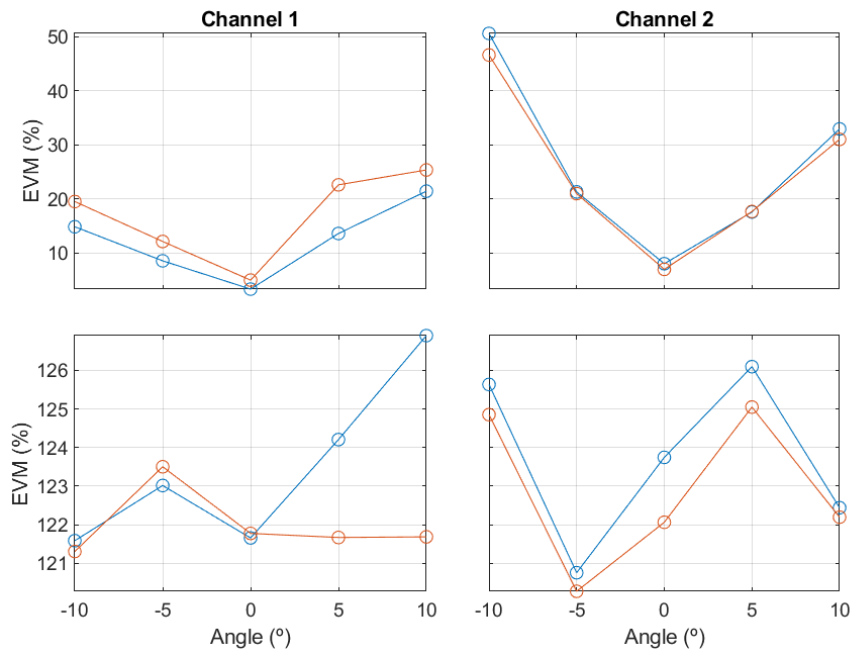


Figure 45. EVM in terms if angle for shaped pulse with EQ and 500 kbaud of symbol rate.

Regarding BER (Fig. 46), we see how good it is for 4-QAM, the only bad point is for Channel 2 at  $-10^\circ$ , but the rest is a completely usable system. As it happened with EVM, for 16-QAM the system does not work properly, and we get these strange shapes of the BER curves with very bad BER of almost 0.5.

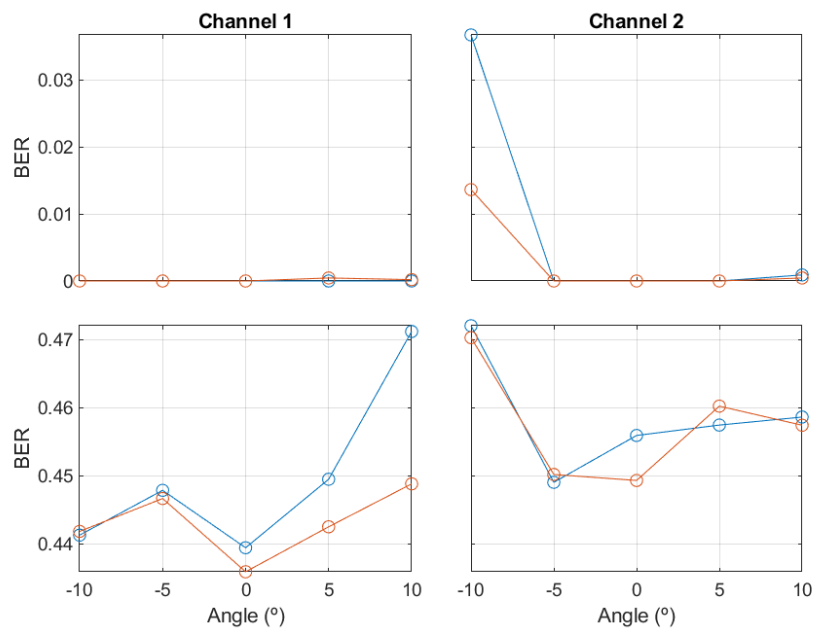


Figure 46. BER in terms if angle for shaped pulse with EQ and 500 kbaud of symbol rate.

In Fig. 47, we have a graph that represents how SNR changes when we modify the angle. We can see that the values we achieve here are the second-best ones from all the scenarios in terms of angle change, the best one is for the other symbol rate. As we can see, both carrier frequencies behave almost in the same way. For both modulation orders (4- and 16-) we notice that it produces a triangular-shaped line with a maximum at  $0^\circ$ . It makes sense that as we increase the angle in both directions, the SNR decreases. Also, as we increase the modulation order, the SNR values

stay very similar, but the 4-QAM one has a little bit better SNR. For example, at  $0^\circ$  4-QAM obtains a value of 40 dB and 16-QAM a value of 38 dB. Also, it is noticeable that SNR decreases more when we move to positive angles than to negative ones.

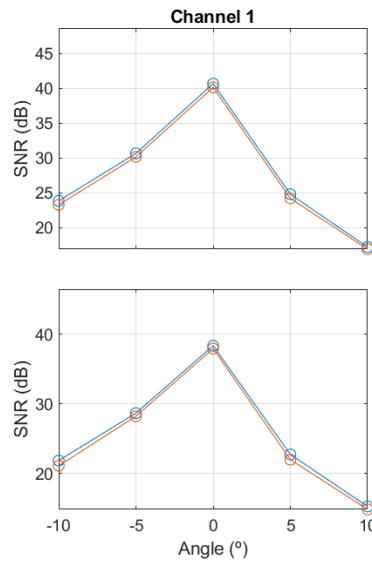


Figure 47. SNR in terms if angle for shaped pulse with EQ and 500 kbaud of symbol rate.

In Fig. 48 we have the constellations for 500 kbaud of symbol rate, 500 kHz of carrier frequency, 4-QAM, 50 cm and  $-10^\circ$  of angle. We can see that even being  $-10^\circ$  we get a very good constellation for Channel 1. But, for Channel 2 we get this kind of noisy constellation that has very bad EVM, but, if we pay attention to it, we can distinguish four areas of points that make up the four points the constellation should have. That is why we still get some bad, but reasonably good BER compared to the EVM. Here, for Channel 1 we obtain an EVM value of 14.88% and for Channel 2 50.64%, what is a very big difference again.

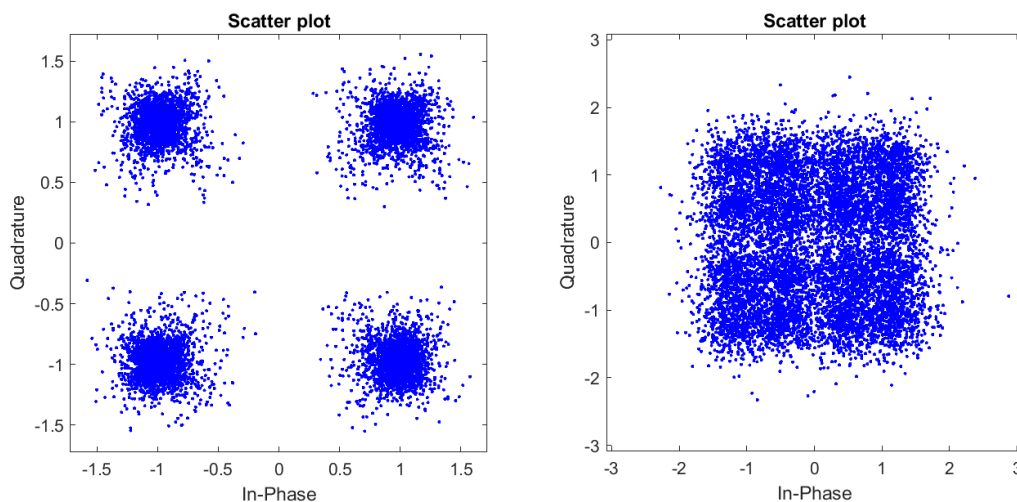


Figure 48. Constellation for shaped pulse with EQ and 500 kbaud of symbol rate with 4-QAM, at 50 cm,  $-10^\circ$  and a carrier frequency of 500 kHz.

## Chapter 5. Conclusion and future research lines

### 5.1 Conclusion

In conclusion, we have been able to create a VLC link (point-to-point) and implement a polarization multiplexing scheme on it and it has been possible to send data through it and receive it properly, so we achieved the goal of providing a Proof of Concept of PDM (Polarization Division Multiplexing) schemes in VLC. This has been possible by using the Instrumentation Control paradigm in MATLAB to control the equipment that has enabled us to send, simple and complex, signals to the LEDs and then receive them from the PDs. Also, thanks to correctly mount and implement the link using the components available on the laboratory that can be seen on Appendix B.

Then, we have to talk about which one of the results we have obtained give us the best performance in terms of transmission speed and which are the work conditions where our system is suitable to perform well. In order to say which values of the measurements we consider that are suitable for a good performance of the system, we have considered that BER values lower than the threshold of  $3.8 * 10^{-3}$  provide correct performance. Also, we have to look at EVM for which we consider good any value below 18.5%, 13.5% and 9% for 4-QAM, 16-QAM and 64-QAM. Then, finally we check if we have a reasonably good value of SNR. When these three conditions meet, we consider that the system is able to perform well under those conditions.

**Table 5. Suitable conditions for variable distance measurements.**

	250 kHz	500 kHz	750 kHz
Rectangular pulse			
250 kbaud	-	4-QAM, 16-QAM	4-QAM, 16-QAM
Shaped pulse			
250 kbaud	4-QAM, 16-QAM at 50 cm	4-QAM, 16-QAM until 70 cm, 64-QAM at 50cm	-
500 kbaud	-	4-QAM	4-QAM

**Table 6. Suitable conditions for variable angle measurements.**

	250 kHz	500 kHz	750 kHz
Shaped pulse			
250 kbaud	4-QAM from -5° to 0°, 16-QAM at 0°	4-QAM from -5° to 5°, 16-QAM at 0°, 64-QAM at 0°	-
500 kbaud	-	4-QAM at 0°	4-QAM at 0°
Shaped pulse with EQ			
250 kbaud	4-QAM from -5° to 5°, 16-QAM at 0°	4-QAM from -5° to 5°, 16-QAM at 0°	-
500 kbaud	-	4-QAM from -5° to 5°	4-QAM from -5° to 0°

In Table 5 and 6, we can see in which cases for each scenario the system performs well.

Regarding the rectangular shape scenario, we obtained good measurements for all distances with 4 and 16-QAM, but not with 64-QAM, independently of the carrier frequency (500 and 750 kHz). So, knowing that our symbol rate is 250 kbaud, using 16-QAM (4 bits per symbol) and two polarizations, we achieve a total speed of  $(250000 * 4 * 2 =) 2$  Mb/s.

Then, in the second scenario (shaped pulse), we see that the system can work in 4-QAM for both symbol rates but only in 16-QAM and 64-QAM at 50cm for 250 kbaud of symbol rate. The system here is more sensible to changes in distance (mainly Channel 2, the noisier one). In this case, our maximum speed case would be to use 250 kbaud of symbol rate, 64-QAM, a carrier frequency of 500 kHz and a maximum distance of 50 cm, with which we would achieve a total speed of  $(250000 * 6 * 2 =) 3$  Mb/s.

So, in terms of distance change, if we need to vary our distance more than 50 cm, we would use the rectangular-shaped pulse achieving 2 Mb/s, but if we would not need to go further than 50 cm we would use the shaped-pulse achieving 3 Mb/s.

Then, in the shaped pulse scenario, if we look to angle change values, for 250 kbaud of symbol rate, we can work at 4-QAM from  $-5^\circ$  to  $0^\circ$  and 16-QAM at  $0^\circ$  for 250 kHz of carrier frequency and at 4-QAM from  $-5^\circ$  to  $5^\circ$ , 16-QAM at  $0^\circ$  and 64 at  $0^\circ$  for 500 kHz. Also, for a symbol rate of 500 kbaud we could work at 4-QAM for both carriers (500 and 700 kHz). So, the best scenario if angle changes is for 250 kbaud of symbol rate, with a carrier frequency of 500 kHz, 4-QAM and from  $-5^\circ$  to  $5^\circ$  achieving a total speed of  $(250000 * 2 * 2 =) 1$  Mb/s. If we did not need angle change, we could use the other configuration that achieved 3Mb/s.

In the last experiments (shaped pulse with EQ), we only collected data for different values of angles. For 250 of symbol rate, we can work at 4-QAM from  $-5^\circ$  to  $5^\circ$  and at 16-QAM at  $0^\circ$  for both carriers. For 500 kbaud of symbol rate, we can work at 4-QAM from  $-5^\circ$  to  $5^\circ$  for 500 kHz of carrier frequency and from  $-5^\circ$  to  $0^\circ$  for 750kHz of carrier frequency. Thus, the best option for angle change is 500 kbaud of symbol rate, 500 kHz of carrier frequency, 4-QAM for an angle from  $-5^\circ$  to  $5^\circ$  achieving a total speed of  $(500000 * 2 * 2 =) 2$  Mb/s.

So, in terms of angle change the best option is the shaped pulse with EQ with a total speed of 2 Mb/s, over the shaped pulse option with a total speed of 1 Mb/s.

## 5.2 Future research lines

Finally, the last part of our project is going to talk about how we can keep improving our system in several different ways. One of them could be combining polarization multiplexing with other multiplexing techniques. For example, if we would implement a CDM (Color Division Multiplexing) scheme for each polarization we would be able to send three different signals in each polarization so, we could triple our speed rates.

Also, this project is only focused on VLC point-to-point for the sake of simplicity, but it could be also interesting to see how a VLC multipoint system using also a polarization multiplexing scheme would be affected by other effects due to its multipoint nature (reflections, ...), or implementing MIMO techniques to the system.

Moreover, we could study in which ways we can try to mitigate the fast degradation that our system suffers when we increment the modulation order or other parameters such as the distance and the angle. Maybe by using more advanced equalizers that can improve our symbol detection.

Finally, one interesting research line would be to try to implement the modulation and reception of the signal into an FPGA, to avoid using the equipment. This way we would be able to easily deploy these systems in real installations.

## References

- [1] P. A. Hoeher, *Visible Light Communications Theoretical and Practical Foundations*. Hanser Publications, 2019.
- [2] A.-M. C̃ and M. Dimian, “Current Challenges for Visible Light Communications Usage in Vehicle Applications: A Survey,” *IEEE COMMUNICATIONS SURVEYS & TUTORIALS*, vol. 19, no. 4, p. 2017, doi: 10.1109/COMST.2017.2706940.
- [3] X. Lu *et al.*, “Memory-controlled deep LSTM neural network post-equalizer used in high-speed PAM VLC system,” *Optics Express*, Vol. 27, Issue 5, pp. 7822–7833, vol. 27, no. 5, pp. 7822–7833, Mar. 2019, doi: 10.1364/OE.27.007822.
- [4] Y. Wang, L. Tao, X. Huang, J. Shi, and N. Chi, “Enhanced Performance of a High-Speed WDM CAP64 VLC System Employing Volterra Series-Based Nonlinear Equalizer”, doi: 10.1109/JPHOT.2015.2436911.
- [5] X. Li, Z. Ghassemlooy, S. Zvánovec, and P. A. Haigh, “A 40 Mb/s VLC System Reusing an Existing Large LED Panel in an Indoor Office Environment †,” 2021, doi: 10.3390/s21051697.
- [6] M. Obeed, A. M. Salhab, M. S. Alouini, and S. A. Zummo, “On Optimizing VLC Networks for Downlink Multi-User Transmission: A Survey,” *IEEE Communications Surveys and Tutorials*, vol. 21, no. 3, pp. 2947–2976, Jul. 2019, doi: 10.1109/COMST.2019.2906225.
- [7] S. Rajbhandari, *Spatial and wavelength division multiplexing for high-speed VLC systems: An overview; Spatial and wavelength division multiplexing for high-speed VLC systems: An overview*. 2016. doi: 10.1109/CSNDSP.2016.7574005.
- [8] I. Siddique, M. Z. Awan, M. Y. Khan, and A. Mazhar, “Li-Fi the Next Generation of Wireless Communication through Visible Light Communication (VLC) Technology,” *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, pp. 30–37, Jan. 2019, doi: 10.32628/cseit1838108.
- [9] G. Corbellini, K. Aksit, S. Mangold, and T. R. Gross, “Connecting Networks of Toys and Smartphones with Visible Light Communication,” 2014.
- [10] M. Maheepala, A. Z. Kouzani, and M. A. Joordens, “Light-Based Indoor Positioning Systems: A Review,” *IEEE Sensors Journal*, vol. 20, no. 8. Institute of Electrical and Electronics Engineers Inc., pp. 3971–3995, Apr. 15, 2020. doi: 10.1109/JSEN.2020.2964380.
- [11] L. U. Khan, “Visible light communication: Applications, architecture, standardization and research challenges,” *Digital Communications and Networks*, vol. 3, no. 2, pp. 78–88, May 2017, doi: 10.1016/J.DCAN.2016.07.004.
- [12] H. Burchardt, N. Serafimovski, D. Tsonev, S. Videv, and H. Haas, “VLC: Beyond point-to-point communication,” 2014.
- [13] X. Guo and N. Chi, “Superposed 32QAM Constellation Design for  $2 \times 2$  Spatial Multiplexing MIMO VLC Systems,” *JOURNAL OF LIGHTWAVE TECHNOLOGY*, vol. 38, no. 7, 2020, doi: 10.1109/JLT.2019.2961146.
- [14] J. zhi Deng and X. hui Cheng, “An experimental scheme for cordwood color division multiplexing VLC system,” *Optoelectron Lett*, vol. 10, no. 6, pp. 459–462, Nov. 2014, doi: 10.1007/s11801-014-4152-1.

## Appendix A. MATLAB code

### Waveform Generator API

```
classdef GeneratorAPI
    properties
        vgen
    end
    methods
        function obj = GeneratorAPI(vendor, address)
            obj.vgen = visa(vendor, address);
        end

        function config(obj,y,fs,samples,element,channel,high,low,offset)
            handler = fopen(strcat("D:\Alejandro\bits",num2str(channel),".txt"),
            'w');
            for i = 1:length(y)
                fprintf(handler, '%d \r\n', y(i));
            end
            fclose(handler)

            copyfile(strcat("D:\Alejandro\bits",num2str(channel),".txt"),'\\AWG3000\AWG_I
            mportedSignals','f');

            fopen(obj.vgen)
            fprintf(obj.vgen, strcat('WLIST:WAV:IMP
            "bits',num2str(channel),"","\\AWG3000\AWG_ImportedSignals\bits',num2str(chann
            el),'.txt'))
            fprintf(obj.vgen,
            strcat('SEQ:ELEM',num2str(element),':WAV',num2str(channel),'
            "bits',num2str(channel),''))

            fprintf(obj.vgen, strcat("AWGC:SRAT"," ",num2str(fs)))
            fprintf(obj.vgen, strcat('SEQ:ELEM',num2str(element),':LENG',"
            ",num2str(samples)))
            fprintf(obj.vgen,
            strcat('SEQ:ELEM',num2str(element),':VOLT:HIGH',num2str(channel),"
            ",num2str(high)))
            fprintf(obj.vgen,
            strcat('SEQ:ELEM',num2str(element),':VOLT:LOW',num2str(channel),"
            ",num2str(low)))
            fprintf(obj.vgen,
            strcat('SEQ:ELEM',num2str(element),':OFF',num2str(channel),"
            ",num2str(offset)))

            fprintf(obj.vgen, 'MARK:LEV1 2')
            fprintf(obj.vgen, 'MARK:MODE1 AUTO')
            fprintf(obj.vgen, strcat('OUTP',num2str(channel)," ON"))
            fclose(obj.vgen)
        end

        function run(obj)
            fopen(obj.vgen)
            fprintf(obj.vgen, 'AWGC:RUN')
```



```

        fclose(obj.vgen)
    end
    function reset(obj)
        fopen(obj.vgen)
        fprintf(obj.vgen, '*RST')
        fprintf(obj.vgen, 'WLIST:WAV:DEL ALL')
        fclose(obj.vgen)
    end
end
end
end

```

## Oscilloscope API

```

classdef OscilloscopeAPI

    properties
        socket
    end

    methods
        function obj = OscilloscopeAPI(ip,bufferSize)
            obj.socket = tcpip(ip, 5025)
            obj.socket.InputBufferSize = bufferSize
        end

        function config(obj,fs,samples,time,channel,endianess)
            fopen(obj.socket)

            fprintf(obj.socket, strcat(':CHAN',num2str(channel),':INP AC'))
            fprintf(obj.socket, strcat(':CHAN',num2str(channel),':PROB:COUP
AC'))
            fprintf(obj.socket, strcat(':CHAN',num2str(channel),':DISP ON'))
            fprintf(obj.socket, ':AUT')
            while(strcmp(query(obj.socket, '*OPC?'),"1")) pause(0.1); end
            fprintf(obj.socket, ':TRIG:LEV AUX,1')
            fprintf(obj.socket, ':TRIG:SWE TRIG')
            fprintf(obj.socket, ':TRIG:MODE EDGE')
            fprintf(obj.socket, ':TRIG:EDGE:SOUR AUX')
            fprintf(obj.socket, ':ACQ:SRAT:AUTO OFF')
            fprintf(obj.socket, strcat(':ACQ:SRAT:ANAL'," ",num2str(fs)))
            pause(0.2)
            query(obj.socket, ':ACQ:SRAT:ANAL?')
            fprintf(obj.socket, strcat(':ACQ:POIN'," ", num2str(samples)))
            fprintf(obj.socket, strcat(':TIM:POS'," ", num2str(time/2)))
            fprintf(obj.socket, strcat(':TIM:RANG'," ",num2str(time)))
            fprintf(obj.socket, ':WAV:FORM WORD')
            fprintf(obj.socket, strcat(':WAV:BYT'," ",endianess))
            while(strcmp(query(obj.socket, '*OPC?'),"1")) pause(0.1); end
            fclose(obj.socket)
        end

        function [data,t] = readdata(obj,channel,samples)
            fopen(obj.socket)
            fprintf(obj.socket, strcat(':WAV:SOUR CHAN',num2str(channel)))

            Q = query(obj.socket, ':WAV:PRE?');

```



```
strQ = string(split(Q, ','));  
  
while(strcmp(query(obj.socket, '*OPC?'), "1")) pause(0.1); end  
  
fprintf(obj.socket, ':WAV:DATA?')  
  
data = fread(obj.socket, samples, 'int16');  
fclose(obj.socket)  
  
t = (0:length(data)-1)*str2double(strQ(5));  
t = t + str2double(strQ(6));  
  
data = data.*str2double(strQ(8)) + str2double(strQ(9));  
end  
  
end  
end
```

## OOK

```
ipgen = '192.168.1.88'  
iposc = '192.168.1.172'  
  
bits = randi([0,1],100,1)  
y = [rectpulse(bits,10); zeros(50,1)];  
  
osc = OscilloscopeAPI(iposc, 40E6);  
gen = GeneratorAPI('ni', 'TCPIP::192.168.1.88::INSTR');  
  
fsgen = 1E6;  
fsosc = 10E6;  
duration = 1.05E-3;  
samples = duration * fsosc;  
  
gen.config(y, fsgen, length(y), 1, 1, 2.5, 1.75, 0);  
osc.config(fsosc, samples, duration, 1);  
data = osc.readdata(samples,1);  
  
h = rectpulse(1,100);  
posfilter = upfirdn(data(75:end),h,1,100);  
posfilter(posfilter < 0) = 0;  
posfilter(posfilter > 0) = 1;  
  
res = posfilter(1:100);  
  
biterr(res,bits)
```

## Simple 4-QAM

```
ipgen = '192.168.1.88'  
iposc = '192.168.1.172'  
  
fsgen = 1E6;  
fsosc = 10E6;  
duration = 1.05E-3;
```





```
samples = duration * fsosc;
samples2 = 30000;

f = 0.2E6;
t = (0:1049)*1/fsgen;

bits = randi([0,3],100,1);
simbolos = qammod(bits,4);
y = real([rectpulse(simbolos,10); zeros(50,1)].*exp(1i*2*pi*f*t.));

osc = OscilloscopeAPI(iposc, 40E6);
gen = GeneratorAPI('ni', 'TCPIP::192.168.1.88::INSTR');

gen.config(y, fsgen, length(y), 1, 1, 2.5, 1.75, 0);
osc.config(fsosc, samples, duration, 1);
[data,ttest] = osc.readdata(samples2,1);
fsosc2 = round(1/(ttest(2)-ttest(1)));
fsimb = fsgen/10;

data = data(205:end);
ttest = ttest(205:end);

t2 = (0:length(data)-1)*1/fsosc2;

data = data.*exp(-1i*(2*pi*f*ttest.' + 0.44 + 3*pi/2));
h = rectpulse(1,fsosc2/fsimb);
posfilter = upfirdn(data,h,1,fsosc2/fsimb);
scatterplot(posfilter)
simbolos2 = qamdemod(posfilter,4);

res = simbolos2(1:100);

symerr(res,bits)
```

### Rectangular pulse measurements

```
ipgen = '192.168.1.88';
iposc = '192.168.1.172';

fsgen = 10E6;
fsosc = 10E6;
num_simb = 20000;

i = 1;
BER = zeros(1,12);
SNR1 = zeros(1,12);
SNR2 = zeros(1,12);
EVM1 = zeros(1,12);
EVM2 = zeros(1,12);

for f = [0.2E6, 1E6]
    for fsimb = [0.1E6, 0.4E6]
        for m = [4, 16, 32]

            t = (0:num_simb*fsgen/2/fsimb+50-1)/fsgen;
```



```
duration = num_simb/fsimb;
samples = duration * fsosc;

code_words = randi([0,m-1],num_simb,1);

simbolos = qammod(code_words,m);
simbolos1 = simbolos(1:2:end);
simbolos2 = simbolos(2:2:end);

y1 = real([rectpulse(simbolos1,fsgen/fsimb);
zeros(50,1)].*exp(1i*2*pi*f*t.));
y2 = real([rectpulse(simbolos2,fsgen/fsimb);
zeros(50,1)].*exp(1i*2*pi*f*t.));

osc = OscilloscopeAPI(iposc, 40E6);
gen = GeneratorAPI('ni', 'TCPIP::192.168.1.88::INSTR');

gen.reset();
gen.config(y1, fsgen, length(y1), 1, 1, 2.125, 2.125, 0);
gen.config(y2, fsgen, length(y2), 1, 2, 2.125, 2.125, 0);
gen.run();
osc.config(fsosc, samples, duration, 1);
osc.config(fsosc, samples, duration, 2);

[noise1,~] = osc.readdata(1);
[noise2,~] = osc.readdata(2);

pot_noise1 = bandpower(noise1,fsosc,[f-fsimb,f+fsimb]);
pot_noise2 = bandpower(noise2,fsosc,[f-fsimb,f+fsimb]);

%only measure noise once, out of the loop

gen.reset();
gen.config(y1, fsgen, length(y1), 1, 1, 2.5, 1.75, 0);
gen.config(y2, fsgen, length(y2), 1, 2, 2.5, 1.75, 0);
gen.run();
osc.config(fsosc, samples, duration, 1);
osc.config(fsosc, samples, duration, 2);

[data1,ttest1] = osc.readdata(1);
[data2,ttest2] = osc.readdata(2);

fsosc2 = round(1/(ttest1(2)-ttest1(1)));

data1 = data1(102:end);
ttest1 = ttest1(102:end);

data2 = data2(100:end);
ttest2 = ttest2(100:end);

pot1 = bandpower(data1,fsosc2,[f-fsimb,f+fsimb]);
pot2 = bandpower(data2,fsosc2,[f-fsimb,f+fsimb]);

data1 = data1.*exp(-1i*(2*pi*f*ttest1.' + 0.95 + 3*pi/2));
data2 = data2.*exp(-1i*(2*pi*f*ttest2.' + 1 + 3*pi/2));
```



```
h = rectpulse(1,fsosc2/fsimb);
posfilter1 = upfirdn(data1,h,1,fsosc2/fsimb);
posfilter2 = upfirdn(data2,h,1,fsosc2/fsimb);

posfilter1 = posfilter1*std(simbolos1)./std(posfilter1);
posfilter2 = posfilter2*std(simbolos2)./std(posfilter2);

scatterplot(simbolos1)
scatterplot(posfilter1)
scatterplot(simbolos2)
scatterplot(posfilter2)

simbolos_demod1 = qamdemod(posfilter1,m);
simbolos_demod2 = qamdemod(posfilter2,m);
code_words1 = code_words(1:2:end);
code_words2 = code_words(2:2:end);

biterr(simbolos_demod1(1:num_simb/2),code_words1)
biterr(simbolos_demod2(1:num_simb/2),code_words2)

res = zeros(1,num_simb);
res(1:2:end) = simbolos_demod1(1:num_simb/2);
res(2:2:end) = simbolos_demod2(1:num_simb/2);

finddelay(res.',code_words)
[~,BER(i)] = biterr(res,code_words.');
```

```
SNR1(i) = pot1/pot_noise1;
SNR2(i) = pot2/pot_noise2;

EVM1(i) = sqrt((sum(abs(simbolos1 -
posfilter1(1:num_simb/2)).^2)/length(simbolos1))/(sum(abs(simbolos1).^2)/length(simbolos1)));
EVM2(i) = sqrt((sum(abs(simbolos2 -
posfilter2(1:num_simb/2)).^2)/length(simbolos2))/(sum(abs(simbolos2).^2)/length(simbolos2)));

    i = i + 1;
end
end
end

save('medidas.mat','BER','SNR1','SNR2','EVM1','EVM2','pot1','pot2')
```

### Shaped pulse (raised cosine) measurements

```
ipgen = '192.168.1.88';
iposc = '192.168.1.172';

pos = '100';
ang = '0';

fsgen = 10E6;
fsosc = 10E6;
num_simb = 20000;
num_train = 2000;
```



```
i = 1;
BER = zeros(1,6);
BER1 = zeros(1,6);
BER2 = zeros(1,6);
SNR1 = zeros(1,6);
SNR2 = zeros(1,6);
EVM1 = zeros(1,6);
EVM2 = zeros(1,6);

osc = OscilloscopeAPI(iposc, 40E6);
gen = GeneratorAPI('ni', 'TCPIP::192.168.1.88::INSTR');

tau1 = (0.68 + 3*pi/2 )./(2*pi*0.2E6);
tau2 = (0.61 + 3*pi/2 )./(2*pi*0.2E6);

for f = [0.5E6, 0.75E6]
    for fsimb = 0.5E6

        endianess = 'MSBF';

        t = (0:num_simb*fsgen/2/fsimb+50-1)/fsgen;

        duration = (num_simb)/2/fsimb;
        samples = duration * fsosc;

        code_words = randi([0,3],num_simb,1);

        simbolos = qammod(code_words,4);
        simbolos1 = simbolos(1:2:end);
        simbolos2 = simbolos(2:2:end);

        rolloff = 0.4;

        y1 = real([rectpulse(simbolos1,fsgen/fsimb);
zeros(50,1)].*exp(1i*2*pi*f*t.));
        y2 = real([rectpulse(simbolos2,fsgen/fsimb);
zeros(50,1)].*exp(1i*2*pi*f*t.));

        gen.reset();
        gen.config(y1, fsgen, length(y1), 1, 1, 2.125, 2.125, 0);
        gen.config(y2, fsgen, length(y2), 1, 2, 2.125, 2.125, 0);
        gen.run();
        osc.config(fsosc, samples, duration, 1, endianess);
        osc.config(fsosc, samples, duration, 2, endianess);

        [noise1,~] = osc.readdata(1,samples);
        [noise2,~] = osc.readdata(2,samples);

        pot_noise1 = bandpower(noise1,fsosc,[f-
fsimb*(1+rolloff)/2,f+fsimb*(1+rolloff)/2]);
        pot_noise2 = bandpower(noise2,fsosc,[f-
fsimb*(1+rolloff)/2,f+fsimb*(1+rolloff)/2]);

        for m = [4, 16, 64]
```

```

t = (0:(num_simb+num_train)*fsgen/2/fsimb+50-1)/fsgen;

duration = (num_simb+num_train++40)/2/fsimb;
samples = duration * fsosc;

code_words = randi([0,m-1],num_simb+num_train,1);

simbolos = qammod(code_words,m);
simbolos1 = simbolos(1:2:end);
simbolos2 = simbolos(2:2:end);

%
txfilter =
comm.RaisedCosineTransmitFilter('OutputSamplesPerSymbol',fsgen/fsimb,'Rolloff
Factor',0.4);

y1 = real([txfilter(simbolos1);
zeros(50,1)].*exp(1i*2*pi*f*t.));
y2 = real([txfilter(simbolos2);
zeros(50,1)].*exp(1i*2*pi*f*t.));

pot_tx_1 = bandpower(y1,fsgen,[f-
fsimb*(1+rolloff)/2,f+fsimb*(1+rolloff)/2]);
pot_tx_2 = bandpower(y2,fsgen,[f-
fsimb*(1+rolloff)/2,f+fsimb*(1+rolloff)/2]);

%only measure noise once, out of the loop

gen.reset();
gen.config(y1, fsgen, length(y1), 1, 1, 2.5, 1.75, 0);
gen.config(y2, fsgen, length(y2), 1, 2, 2.5, 1.75, 0);
gen.run();
osc.config(fsosc, samples, duration, 1, endianness);
osc.config(fsosc, samples, duration, 2, endianness);

[data1,ttest1] = osc.readdata(1,samples);
[data2,ttest2] = osc.readdata(2,samples);

fsosc2 = 1/(ttest1(2)-ttest1(1));

[pxx1,f1] =
periodogram(data1,rectwin(length(data1)),max(256,2^nextpow2(length(data1))),f
sosc2);
[pxx2,f2] =
periodogram(data2,rectwin(length(data2)),max(256,2^nextpow2(length(data2))),f
sosc2);

per_pre1 = figure;
semilogy(f1,pxx1);
xlabel("Frequency (Hz)")
ylabel("Power Spectral Density (W/Hz)")
xlim([0 5E6])

saveas(gcf,strcat('periodograms/pre/per_pre1_f_',num2str(f),'_rs_',num2str(fs
imb),'_m_',num2str(m),'_pos_',pos,'_ang_',ang,'.png'))
per_pre2 = figure;
semilogy(f2,pxx2);

```

```
xlabel("Frequency (Hz)")
ylabel("Power Spectral Density (W/Hz)")
xlim([0 5E6])

saveas(gcf, strcat('periodograms/pre/per_pre2_f_', num2str(f), '_rs_', num2str(fs_
imb), '_m_', num2str(m), '_pos_', pos, '_ang_', ang, '.png'))

carrier_sync =
comm.CarrierSynchronizer('SamplesPerSymbol', 1, 'Modulation', 'QAM');
symbol_sync1 =
comm.SymbolSynchronizer('TimingErrorDetector', 'Zero-Crossing (decision-
directed)', 'SamplesPerSymbol', 10, 'NormalizedLoopBandwidth', 0.01, 'DampingFacto
r', 1);
symbol_sync2 =
comm.SymbolSynchronizer('TimingErrorDetector', 'Zero-Crossing (decision-
directed)', 'SamplesPerSymbol', 10, 'NormalizedLoopBandwidth', 0.01, 'DampingFacto
r', 1);

pot1 = bandpower(data1, fsosc2, [f-
fsimb*(1+rolloff)/2, f+fsimb*(1+rolloff)/2]);
pot2 = bandpower(data2, fsosc2, [f-
fsimb*(1+rolloff)/2, f+fsimb*(1+rolloff)/2]);

data1 = data1.*exp(-1i*(2*pi*f*(ttest1.')));
data2 = data2.*exp(-1i*(2*pi*f*(ttest2.')));

[pxx3, f3] =
periodogram(data1, rectwin(length(data1)), max(256, 2^nextpow2(length(data1))), f_
sosc2);
[pxx4, f4] =
periodogram(data2, rectwin(length(data2)), max(256, 2^nextpow2(length(data2))), f_
sosc2);
per_pos1 = figure;
f3 = linspace(-5E6, 5E6, length(f3));
semilogy(f3, fftshift(pxx3));
xlabel("Frequency (Hz)")
ylabel("Power Spectral Density (W/Hz)")
xlim([-5E6 5E6])

saveas(gcf, strcat('periodograms/pos/per_pos1_f_', num2str(f), '_rs_', num2str(fs_
imb), '_m_', num2str(m), '_pos_', pos, '_ang_', ang, '.png'))
per_pos2 = figure;
f4 = linspace(-5E6, 5E6, length(f4));
semilogy(f4, fftshift(pxx4));
xlabel("Frequency (Hz)")
ylabel("Power Spectral Density (W/Hz)")
xlim([-5E6 5E6])

saveas(gcf, strcat('periodograms/pos/per_pos2_f_', num2str(f), '_rs_', num2str(fs_
imb), '_m_', num2str(m), '_pos_', pos, '_ang_', ang, '.png'))

rxfilter =
comm.RaisedCosineReceiveFilter('InputSamplesPerSymbol', round(fsosc2/fsimb), 'D_
ecimationFactor', round(fsosc2/fsimb)/10, 'RolloffFactor', 0.4);

posfilter1 =
2*rxfilter(data1(num_train*round(fsosc2/fsimb)/2+1:end));
```

```
posfilter2 =
2*rxfilter(data2(num_train*round(fsosc2/fsimb)/2+1:end));
%
datasync1 =
symbol_sync1(posfilter1(1+(txfilter.FilterSpanInSymbols+rxfilter.FilterSpanIn
Symbols)*round(fsosc2/fsimb)/10:end));
datasync2 =
symbol_sync2(posfilter2(1+(txfilter.FilterSpanInSymbols+rxfilter.FilterSpanIn
Symbols)*round(fsosc2/fsimb)/10:end));

constelacion_possimb1 = scatterplot(datasync1);

saveas(gcf, strcat('constelations/possimb/constelacion_possimb1_f_', num2str(f)
, '_rs_', num2str(fsimb), '_m_', num2str(m), '_pos_', pos, '_ang_', ang, '.png'))
constelacion_possimb2 = scatterplot(datasync2);

saveas(gcf, strcat('constelations/possimb/constelacion_possimb2_f_', num2str(f)
, '_rs_', num2str(fsimb), '_m_', num2str(m), '_pos_', pos, '_ang_', ang, '.png'))

poscar1 = carrier_sync(datasync1);
poscar2 = carrier_sync(datasync2);

poscar1 = poscar1*std(simbolos1)./std(poscar1);
poscar2 = poscar2*std(simbolos2)./std(poscar2);

scatterplot(simbolos1)
constelacion1 = scatterplot(poscar1(1:(num_simb)/2));

saveas(gcf, strcat('constelations/eq/constelacion1_f_', num2str(f), '_rs_', num2s
tr(fsimb), '_m_', num2str(m), '_pos_', pos, '_ang_', ang, '.png'))
scatterplot(simbolos2)
constelacion2 = scatterplot(poscar2(1:(num_simb)/2));

saveas(gcf, strcat('constelations/eq/constelacion2_f_', num2str(f), '_rs_', num2s
tr(fsimb), '_m_', num2str(m), '_pos_', pos, '_ang_', ang, '.png'))

code_words1 = code_words(1:2:end);
code_words2 = code_words(2:2:end);
j = 1;
phis = [0:pi/2:3*pi/2];

if fsimb == 0.25E6
    delay = 2;
else
    delay = 6;
end

for phi = phis

    [~, BERtest(j)] = biterr(qamdemod(
exp(1i*phi)*poscar1(1:(num_simb)/2), m), code_words1(num_train/2+1-delay:end-
delay));
    j = j + 1;
end

[~, index] = min(BERtest);
```



```

        simbolos_demod1 =
qamdemod(exp(1i*phis(index))*poscar1(1:(num_simb)/2),m);
        simbolos_demod2 =
qamdemod(exp(1i*phis(index))*poscar2(1:(num_simb)/2),m);

        [~,BER1(i)] = biterr(simbolos_demod1,code_words1(num_train/2+1-
delay:end-delay));
        [~,BER2(i)] = biterr(simbolos_demod2,code_words2(num_train/2+1-
delay:end-delay));

        res = zeros(1,num_simb);
        res(1:2:end) = simbolos_demod1;
        res(2:2:end) = simbolos_demod2;

        finddelay(res.',code_words);
        [~,BER(i)] = biterr(res,code_words(num_train+1-2*delay:end-
2*delay).');

        SNR1(i) = pot1/pot_noise1;
        SNR2(i) = pot2/pot_noise2;

        EVM1(i) = sqrt((sum(abs(simbolos1(num_train/2+1-delay:end-delay)
-
exp(1i*phis(index))*poscar1(1:(num_simb)/2)).^2)/length(simbolos1(num_train/2
+1-delay:end-delay)))/(sum(abs(simbolos1(num_train/2+1-delay:end-
delay)).^2)/length(simbolos1(num_train/2+1-delay:end-delay))));
        EVM2(i) = sqrt((sum(abs(simbolos2(num_train/2+1-delay:end-delay)
-
exp(1i*phis(index))*poscar2(1:(num_simb)/2)).^2)/length(simbolos2(num_train/2
+1-delay:end-delay)))/(sum(abs(simbolos2(num_train/2+1-delay:end-
delay)).^2)/length(simbolos2(num_train/2+1-delay:end-delay))));

        i = i + 1;

        close("all")
    end
end
end

save(strcat('medidas/medidas_rs_',num2str(fsimb),'_pos_',pos,'_ang_',ang,'.ma
t'),'pos','ang','BER','BER1','BER2','SNR1','SNR2','EVM1','EVM2','pot1','pot2'
)

```

### Shaped pulse (raised cosine) with EQ measurements

```

ipgen = '192.168.1.88';
iposc = '192.168.1.172';

pos = '50';
ang = '0';

fsген = 10E6;
fsosc = 10E6;
num_simb = 20000;
num_train = 2000;

i = 1;

```





```
BER = zeros(1,6);
BER1 = zeros(1,6);
BER2 = zeros(1,6);
SNR1 = zeros(1,6);
SNR2 = zeros(1,6);
EVM1 = zeros(1,6);
EVM2 = zeros(1,6);

osc = OscilloscopeAPI(iposc, 40E6);
gen = GeneratorAPI('ni', 'TCPIP::192.168.1.88::INSTR');

tau1 = (0.68 + 3*pi/2 )./(2*pi*0.2E6);
tau2 = (0.61 + 3*pi/2 )./(2*pi*0.2E6);

for f = [0.5E6, 0.75E6]
    for fsimb = 0.5E6

        endianess = 'MSBF';

        t = (0:num_simb*fsgen/2/fsimb+50-1)/fsgen;

        duration = (num_simb)/2/fsimb;
        samples = duration * fsosc;

        code_words = randi([0,3],num_simb,1);

        simbolos = qammod(code_words,4);
        simbolos1 = simbolos(1:2:end);
        simbolos2 = simbolos(2:2:end);

        rolloff = 0.4;

        y1 = real([rectpulse(simbolos1,fsgen/fsimb);
zeros(50,1)].*exp(1i*2*pi*f*t.));
        y2 = real([rectpulse(simbolos2,fsgen/fsimb);
zeros(50,1)].*exp(1i*2*pi*f*t.));

        gen.reset();
        gen.config(y1, fsgen, length(y1), 1, 1, 2.125, 2.125, 0);
        gen.config(y2, fsgen, length(y2), 1, 2, 2.125, 2.125, 0);
        gen.run();
        osc.config(fsosc, samples, duration, 1, endianess);
        osc.config(fsosc, samples, duration, 2, endianess);

        [noise1,~] = osc.readdata(1,samples);
        [noise2,~] = osc.readdata(2,samples);

        pot_noise1 = bandpower(noise1,fsosc,[f-
fsimb*(1+rolloff)/2,f+fsimb*(1+rolloff)/2]);
        pot_noise2 = bandpower(noise2,fsosc,[f-
fsimb*(1+rolloff)/2,f+fsimb*(1+rolloff)/2]);

        for m = [4, 16]

            t = (0:(num_simb+num_train)*fsgen/2/fsimb+50-1)/fsgen;
```

```
duration = (num_simb+num_train++40)/2/fsimb;
samples = duration * fsosc;

code_words = randi([0,m-1],num_simb+num_train,1);

simbolos = qammod(code_words,m);
simbolos1 = simbolos(1:2:end);
simbolos2 = simbolos(2:2:end);

txfilter =
comm.RaisedCosineTransmitFilter('OutputSamplesPerSymbol',fsgen/fsimb,'Rolloff
Factor',0.4);

y1 = real([txfilter(simbolos1);
zeros(50,1)].*exp(1i*2*pi*f*t.));
y2 = real([txfilter(simbolos2);
zeros(50,1)].*exp(1i*2*pi*f*t.));

pot_tx_1 = bandpower(y1,fsgen,[f-
fsimb*(1+rolloff)/2,f+fsimb*(1+rolloff)/2]);
pot_tx_2 = bandpower(y2,fsgen,[f-
fsimb*(1+rolloff)/2,f+fsimb*(1+rolloff)/2]);

%only measure noise once, out of the loop

gen.reset();
gen.config(y1, fsgen, length(y1), 1, 1, 2.5, 1.75, 0);
gen.config(y2, fsgen, length(y2), 1, 2, 2.5, 1.75, 0);
gen.run();
osc.config(fsosc, samples, duration, 1, endianness);
osc.config(fsosc, samples, duration, 2, endianness);

[data1,ttest1] = osc.readdata(1,samples);
[data2,ttest2] = osc.readdata(2,samples);

fsosc2 = 1/(ttest1(2)-ttest1(1));

[pxx1,f1] =
periodogram(data1,rectwin(length(data1)),max(256,2^nextpow2(length(data1))),f
sosc2);
[pxx2,f2] =
periodogram(data2,rectwin(length(data2)),max(256,2^nextpow2(length(data2))),f
sosc2);
per_pre1 = figure;
semilogy(f1,pxx1);
xlabel("Frequency (Hz)")
ylabel("Power Spectral Density (W/Hz)")
xlim([0 5E6])

saveas(gcf,strcat('periodograms/pre/per_pre1_f_',num2str(f),'_rs_',num2str(fs
imb),'_m_',num2str(m),'_pos_',pos,'_ang_',ang,'.png'))
per_pre2 = figure;
semilogy(f2,pxx2);
xlabel("Frequency (Hz)")
ylabel("Power Spectral Density (W/Hz)");
```

```
xlim([0 5E6])

saveas(gcf, strcat('periodograms/pre/per_pre2_f_', num2str(f), '_rs_', num2str(fs
imb), '_m_', num2str(m), '_pos_', pos, '_ang_', ang, '.png'))

    carrier_sync =
comm.CarrierSynchronizer('SamplesPerSymbol', 1, 'Modulation', 'QAM');
    symbol_sync1 =
comm.SymbolSynchronizer('TimingErrorDetector', 'Zero-Crossing (decision-
directed)', 'SamplesPerSymbol', 10, 'NormalizedLoopBandwidth', 0.01, 'DampingFacto
r', 1);
    symbol_sync2 =
comm.SymbolSynchronizer('TimingErrorDetector', 'Zero-Crossing (decision-
directed)', 'SamplesPerSymbol', 10, 'NormalizedLoopBandwidth', 0.01, 'DampingFacto
r', 1);

lineq1 =
comm.LinearEqualizer('NumTaps', 15, 'ReferenceTap', 8, 'Constellation', qammod(0:m
-1, m));
    lineq2 =
comm.LinearEqualizer('NumTaps', 15, 'ReferenceTap', 8, 'Constellation', qammod(0:m
-1, m));

    pot1 = bandpower(data1, fsosc2, [f-
fsimb*(1+rolloff)/2, f+fsimb*(1+rolloff)/2]);
    pot2 = bandpower(data2, fsosc2, [f-
fsimb*(1+rolloff)/2, f+fsimb*(1+rolloff)/2]);

    data1 = data1.*exp(-1i*(2*pi*f*(ttest1.')));
    data2 = data2.*exp(-1i*(2*pi*f*(ttest2.')));

    [pxx3, f3] =
periodogram(data1, rectwin(length(data1)), max(256, 2^nextpow2(length(data1))), f
sosc2);
    [pxx4, f4] =
periodogram(data2, rectwin(length(data2)), max(256, 2^nextpow2(length(data2))), f
sosc2);
    per_pos1 = figure;
    f3 = linspace(-5E6, 5E6, length(f3));
    semilogy(f3, fftshift(pxx3));
    xlabel("Frequency (Hz)")
    ylabel("Power Spectral Density (W/Hz)")
    xlim([-5E6 5E6])

saveas(gcf, strcat('periodograms/pos/per_pos1_f_', num2str(f), '_rs_', num2str(fs
imb), '_m_', num2str(m), '_pos_', pos, '_ang_', ang, '.png'))
    per_pos2 = figure;
    f4 = linspace(-5E6, 5E6, length(f4));
    semilogy(f4, fftshift(pxx4));
    xlabel("Frequency (Hz)")
    ylabel("Power Spectral Density (W/Hz)")
    xlim([-5E6 5E6])

saveas(gcf, strcat('periodograms/pos/per_pos2_f_', num2str(f), '_rs_', num2str(fs
imb), '_m_', num2str(m), '_pos_', pos, '_ang_', ang, '.png'))
```

```
rxfilter =
comm.RaisedCosineReceiveFilter('InputSamplesPerSymbol',round(fsosc2/fsimb),'D
ecimationFactor',round(fsosc2/fsimb)/10,'RolloffFactor',0.4);

posfilter1 = 2*rxfilter(data1);
posfilter2 = 2*rxfilter(data2);

datasync1 =
symbol_sync1(posfilter1(1+(txfilter.FilterSpanInSymbols+rxfilter.FilterSpanIn
Symbols)*round(fsosc2/fsimb)/10:end));
datasync2 =
symbol_sync2(posfilter2(1+(txfilter.FilterSpanInSymbols+rxfilter.FilterSpanIn
Symbols)*round(fsosc2/fsimb)/10:end));

constelacion_possimb1 = scatterplot(datasync1);

saveas(gcf,strcat('constelations/possimb/constelacion_possimb1_f_',num2str(f)
,'_rs_',num2str(fsimb),'_m_',num2str(m),'_pos_',pos,'_ang_',ang,'.png'))
constelacion_possimb2 = scatterplot(datasync2);

saveas(gcf,strcat('constelations/possimb/constelacion_possimb2_f_',num2str(f)
,'_rs_',num2str(fsimb),'_m_',num2str(m),'_pos_',pos,'_ang_',ang,'.png'))

poscar1 = carrier_sync(datasync1);
poscar2 = carrier_sync(datasync2);

poscar1 = poscar1*std(simbolos1)./std(poscar1);
poscar2 = poscar2*std(simbolos2)./std(poscar2);

scatterplot(simbolos1)
constelacion1 = scatterplot(poscar1(1:(num_simb)/2));

saveas(gcf,strcat('constelations/eq/constelacion1_f_',num2str(f),'_rs_',num2s
tr(fsimb),'_m_',num2str(m),'_pos_',pos,'_ang_',ang,'.png'))
scatterplot(simbolos2)
constelacion2 = scatterplot(poscar2(1:(num_simb)/2));

saveas(gcf,strcat('constelations/eq/constelacion2_f_',num2str(f),'_rs_',num2s
tr(fsimb),'_m_',num2str(m),'_pos_',pos,'_ang_',ang,'.png'))

code_words1 = code_words(1:2:end);
code_words2 = code_words(2:2:end);
j = 1;
phis = [0:pi/2:3*pi/2];

delay = 7;
for phi = phis

    [~,BERtest(j)] = biterr(qamdemod(
exp(1i*phi)*poscar1(num_train/2+1:num_train/2+(num_simb)/2),m),code_words1(num
_train/2+1-delay:end-delay));
    j = j + 1;
end

[~,index] = min(BERtest);

poseq1 = lineq1(datasync1,simbolos1(1:num_train));
poseq2 = lineq2(datasync2,simbolos2(1:num_train));
```



```
        scatterplot(simbolos1)
        constelacion1 =
scatterplot(poseq1(num_train/2+1:num_train/2+1+(num_simb)/2));

saveas(gcf, strcat('constelations/eq/constelacion1_f_', num2str(f), '_rs_', num2s
tr(fsimb), '_m_', num2str(m), '_pos_', pos, '_ang_', ang, '.png'))
        scatterplot(simbolos2)
        constelacion2 =
scatterplot(poseq2(num_train/2+1:num_train/2+1+(num_simb)/2));

saveas(gcf, strcat('constelations/eq/constelacion2_f_', num2str(f), '_rs_', num2s
tr(fsimb), '_m_', num2str(m), '_pos_', pos, '_ang_', ang, '.png'))

        simbolos_demod1 =
qamdemod(exp(1i*phis(index))*poseq1(num_train/2+1:num_train/2+(num_simb)/2),m
);
        simbolos_demod2 =
qamdemod(exp(1i*phis(index))*poseq2(num_train/2+1:num_train/2+(num_simb)/2),m
);

        [~,BER1(i)] = biterr(simbolos_demod1,code_words1(num_train/2+1-
delay:end-delay));
        [~,BER2(i)] = biterr(simbolos_demod2,code_words2(num_train/2+1-
delay:end-delay));

        res = zeros(1,num_simb);
        res(1:2:end) = simbolos_demod1;
        res(2:2:end) = simbolos_demod2;

        finddelay(res.',code_words);
        [~,BER(i)] = biterr(res,code_words(num_train+1-2*delay:end-
2*delay).');

        SNR1(i) = pot1/pot_noise1;
        SNR2(i) = pot2/pot_noise2;

        EVM1(i) = sqrt((sum(abs(simbolos1(num_train/2+1-delay:end-delay)
-
exp(1i*phis(index))*poscar1(1:(num_simb)/2)).^2)/length(simbolos1(num_train/2
+1-delay:end-delay)))/(sum(abs(simbolos1(num_train/2+1-delay:end-
delay)).^2)/length(simbolos1(num_train/2+1-delay:end-delay))));
        EVM2(i) = sqrt((sum(abs(simbolos2(num_train/2+1-delay:end-delay)
-
exp(1i*phis(index))*poscar2(1:(num_simb)/2)).^2)/length(simbolos2(num_train/2
+1-delay:end-delay)))/(sum(abs(simbolos2(num_train/2+1-delay:end-
delay)).^2)/length(simbolos2(num_train/2+1-delay:end-delay))));

        i = i + 1;

        close("all")
    end
end
end
save(strcat('medidas/medidas_rs_', num2str(fsimb), '_pos_', pos, '_ang_', ang, '.ma
t'), 'pos', 'ang', 'BER', 'BER1', 'BER2', 'SNR1', 'SNR2', 'EVM1', 'EVM2', 'pot1', 'pot2'
)
```

## Appendix B. Component datasheets

### PDA36A and PDA10A (Photodetectors)

**Light Analysis**


▼ CHAPTERS

- Power Meters
- Detectors**
- Beam Characterization
- Polarimetry
- Electronics Accessories

▼ SECTIONS

- Biased Photodetectors
- Amplified Photodetectors**
- Photon Counter
- Integrating Spheres
- Photomultiplier Tubes
- Balanced Detectors
- Position-Sensing Detectors
- Photodiodes
- Photocurrent Amplifiers
- Cameras

### 150 nm to 4.8 μm Amplified Photodetectors (Page 1 of 2)



**PDA36A**  
Power Supply  
Included

The PDA series of amplified photodetectors have a thin profile to allow access to light paths where there is minimal space. All connections and controls are located perpendicular to the light path. In addition, the PDA series includes a low-noise transimpedance or voltage amplifier capable of driving 50 Ω loads.

The housing features both external SM1 (1.035"-40) threads and internal SM05 (0.535"-40) threads, each of which are compatible with a large array of our threaded accessories, allowing convenient mounting of external optics, fibers, and apertures. Each housing provides two 8-32 tapped mounting holes (M4 x 0.7 for -EC versions). An internally SM1-threaded adapter ring is also included, as is a switchable power supply (120 VAC, 230 VAC).


Thorlabs' line of switchable gain detectors provide gain adjustment over a 70 dB range to take full advantage of the photodiode response range. Gains are adjustable from 1.5 kV/A to 4.7 MV/A in eight 10 dB steps. Our selection of wideband detectors increase the bandwidth range from DC to 150 MHz, while still maintaining low noise.

**All Detectors Come Complete with a Power Supply**

**PDA Series General Specifications**

ITEM #	SENSOR	BANDWIDTH <sup>1</sup>	WAVELENGTH RANGE	ACTIVE AREA	GAIN
PDA25K	GaP	7.5 MHz	150 – 550 nm	6.5 mm <sup>2</sup> (2.54 mm x 2.54 mm)	1.5 x 10 <sup>3</sup> to 4.75 x 10 <sup>6</sup> V/A <sup>2</sup>
PDA10A	Si	150 MHz	200 – 1100 nm	0.8 mm <sup>2</sup> (Ø1 mm)	1 x 10 <sup>4</sup> V/A
PDABA	Si	50 MHz	320 – 1000 nm	0.5 mm <sup>2</sup> (Ø0.8 mm)	1 x 10 <sup>5</sup> V/A
PDA36A	Si	17 MHz	350 – 1100 nm	13 mm <sup>2</sup> (3.6 mm x 3.6 mm)	1.5 x 10 <sup>3</sup> to 4.75 x 10 <sup>6</sup> V/A <sup>2</sup>
PDA100A	Si	1.5 MHz	400 – 1100 nm	75.4 mm <sup>2</sup> (Ø9.8 mm)	1.5 x 10 <sup>3</sup> to 4.75 x 10 <sup>6</sup> V/A <sup>2</sup>
PDA10CF	InGaAs	150 MHz	700 – 1800 nm	0.2 mm <sup>2</sup> (0.5 mm)	1 x 10 <sup>5</sup> V/A
PDA10CS	InGaAs	17 MHz	700 – 1800 nm	0.8 mm <sup>2</sup> (Ø1 mm)	1.5 x 10 <sup>3</sup> to 4.75 x 10 <sup>6</sup> V/A <sup>2</sup>
PDA50B	Ge	400 kHz	800 – 1800 nm	19.6 mm <sup>2</sup> (Ø5 mm)	1.5 x 10 <sup>3</sup> to 4.75 x 10 <sup>6</sup> V/A <sup>2</sup>
PDA10D	InGaAs	15 MHz	1200 – 2600 nm	0.8 mm <sup>2</sup> (Ø1 mm)	1 x 10 <sup>4</sup> V/A
PDA30G	PbS	0.2 kHz to 1 kHz <sup>3</sup>	1000 – 2900 nm	9 mm <sup>2</sup> (3.0 mm x 3.0 mm)	100X
PDA20H	PbSe	0.2 kHz to 10 kHz <sup>4</sup>	1500 – 4800 nm	4 mm <sup>2</sup> (2.0 mm x 2.0 mm)	100X

**SM1FC**  
SM1-Threaded Fiber Adapters



<sup>1</sup> Same specs apply to the -EC versions  
<sup>2</sup> Applies to lowest gain setting on switchable gain versions  
<sup>3</sup> Switchable gain, 8 steps, 70 dB total adjustment  
<sup>4</sup> AC-coupled output only

**PDA25K Responsivity**



**PDA10A, PDA36A, PDA100A Responsivity**



**PDABA Responsivity**



**PDA10CF, PDA10CS, PDA50B Responsivity**



**PDA10D Responsivity**



**PDA30G, PDA20H Responsivity**



## LB1761-ML (Lens for LEDs)

(The photo is not exactly the same model but the lens inside (N-BK7) is the same but the focal length is 25 mm)

Products Home / Optical Elements / Optical Lenses / Spherical Singlet Lenses / Bi-Convex Spherical Lenses / N-BK7 Bi-Convex Spherical Lenses / Mounted N-BK7 Bi-Convex Lenses (Uncoated)

**Mounted N-BK7 Bi-Convex Lenses (Uncoated)**

- ▶ Positive Focal Length for Use at Finite Conjugates
- ▶ Uncoated Wavelength Range: 350 - 2000 nm
- ▶ SM05-Threaded or SM1-Threaded Mounts

LB1869-ML (Ø1")

LB1187-ML (Ø1/2")

**Related Items**

- Other Bi-Convex Lenses
- Quick Release Optic Mount
- Bi-Convex Lens Kits
- Lens Tube Systems

Overview | Graphs | Tutorial | Feedback

### Features

- Material: N-BK7
- Wavelength Range: 350 - 2000 nm (Uncoated)
- Available Sizes: Ø1/2" and Ø1"
- Focal Lengths Available from 15.0 mm to 1.0 m
- Well Suited for Many Finite Imaging Applications
- Offers Excellent Transmittance Throughout the Visible and Near Infrared
- Mounted in SM-Compatible Lens Cells

Thorlabs' Mounted Uncoated N-BK7 Bi-Convex Lenses are popular for many finite imaging applications. This type of lens is best suited for use in situations where the object and image are on opposite sides of the lens and the ratio of the image and object distances (conjugate ratio) is between 0.2 and 5.

N-BK7 is probably the most common optical glass used for high-quality optical components. It is typically chosen whenever the additional benefits of UV fused silica (i.e., good transmission further into the UV and a lower coefficient of thermal expansion) are not necessary. These Ø1/2" and Ø1" lenses are mounted in SM05- (0.535"-40) and SM1- (1.035"-40) compatible mounts, respectively.

Although this page features our uncoated bi-convex lenses, these lenses are also available with one of three antireflection coatings (-A, -B, or -C) deposited on both sides to reduce the amount of light reflected from each surface. Links to each of these pages can be found in the table to the right. Please see the Graphs tab for coating details.

Lens kits are available. Please click [here](#) for information.

**Zemax Files**

Click on the red Document icon next to the item numbers below to access the Zemax file download. Our entire [Zemax Catalog](#) is also available.

**Optic Handling and Cleaning Tutorial**

**Optical Coatings Guide**

N-BK7 Bi-Convex Spherical Singlets	
Lens Shape	Convex/Convex
Material	N-BK7 (Grade A)
Wavelength Range <sup>a</sup>	350 nm - 2.0 µm (Uncoated)
Design Wavelength	633 nm
Index of Refraction	1.515 (@ 633 nm)
Surface Quality	40-20 Scratch-Dig
Surface Irregularity	M/4
Spherical Surface Power	3λ/2
Abbe Number	$v_d=84.17$
Centration	≤3 arcmin
Clear Aperture	Ø1/2" lenses: >Ø11.04 mm Ø1" lenses: >Ø22.85 mm
Diameter Tolerance	+0.00 mm / -0.10 mm
Focal Length Tolerance	±1%

<sup>a</sup> Also available with a [-A coating](#) for the 350 - 700 nm range, a [-B coating](#) for the 650 - 1050 nm range, or a [-C coating](#) for the 1050 - 1700 nm range.

Bi-Convex Lens Selection Guide		
N-BK7	Uncoated	-B (850 - 1050 nm)
	-A (350 - 700 nm)	-C (1050 - 1700 nm)
Mounted N-BK7	Uncoated	-B (850 - 1050 nm)
	-A (350 - 700 nm)	-C (1050 - 1700 nm)
UV Fused Silica	Uncoated	-B (850 - 1050 nm)
	-UV (245 - 400 nm)	-C (1050 - 1700 nm)
	-A (350 - 700 nm)	
CaF <sub>2</sub>	Uncoated	-E (3 - 5 µm)
ZnSe		-F (8 - 12 µm)

Custom Coatings are also available. Please contact our [technical support staff](#) for a quote.

## ACL2520-A (Lens for Photodetectors)



**ACL2520, ACL2520-A, ACL2520-B - July 15, 2015**

Item #s ACL2520 and ACL2520-A, ACL2520-B were discontinued on July 15, 2015. For informational purposes, this is a copy of the website content at that time and is valid only for the stated product.

- ▶ Diameters Ranging from 10 to 75 mm
- ▶ Numerical Apertures Ranging from 0.488 to 0.79
- ▶ Substrate Transmission Range: 380 - 2100 nm
- ▶ Optimized For Condenser Applications

**ACL1210**  
Ø12 mm, f = 10.5 mm

**ACL1815-A**  
Ø18 mm, f = 15 mm

**ACL25416U-B**  
Ø25.4 mm (Ø1"), f = 16.0 mm

**ACL5040**  
Ø50 mm, f = 40 mm

**ACL7560-A**  
Ø75 mm, f = 60 mm

**Condenser Application**

[Hide Overview](#)

### OVERVIEW

#### Features

- 13 Different Diameters Available
- Available Uncoated or with One of Two AR Coatings
- Offers Higher NA (0.488 to 0.79) and Less Spherical Aberration than Spherical Lenses

#### High-Efficiency Illumination Applications

- Light Collection
- Projection
- Detection
- Condensing

These Aspheric Condenser Lenses are ideal for high-efficiency illumination applications. Compared to spherical lenses, our aspheric condenser lenses introduce less aberration, offer larger apertures, and provide lower f/# ratios. These aspheric condenser lenses are designed for collimating light from a lamp, LED, or similar light source; for best performance in this application, the flatter side of the lens should face the source. Shorter focal lengths and a low f/# also allows these lenses to be used in close proximity to one another or other optical elements. This makes them well suited for focusing light onto a detector or light collection.

To offer more flexibility for the design of your optical system, these lenses are available in diameters ranging from 10 to 75 mm. The aspheric surface is

#### Zemax Files

Click on the red Document icon next to the item numbers below to access the Zemax file download. Our entire Zemax Catalog is also available.

#### Common Specifications<sup>a</sup>

Design Wavelength	Visible
Glass Type	B270 Optical Crown Glass
Wavelength Range	Uncoated: 380 - 2100 nm A: 350 - 700 nm B: 650 - 1050 nm
Reflectance Over AR Coating Range for Coated Optics (Avg., AOI = 0°)	<0.5%
Uncoated Transmission Graph (Click Here to Download Raw Data)	
A Coating Reflectance Graph (Click Here to Download Raw Data)	
B Coating Reflectance Graph (Click Here to Download Raw Data)	
Diameter Tolerance	+0.0 mm / -0.5 mm
Center Thickness Tolerance	±0.3 mm
Centration	<30 arcmin
Clear Aperture	>90%
Maximum Temperature	250 °C

<sup>a</sup> Please see the Specs tab for more specifications.



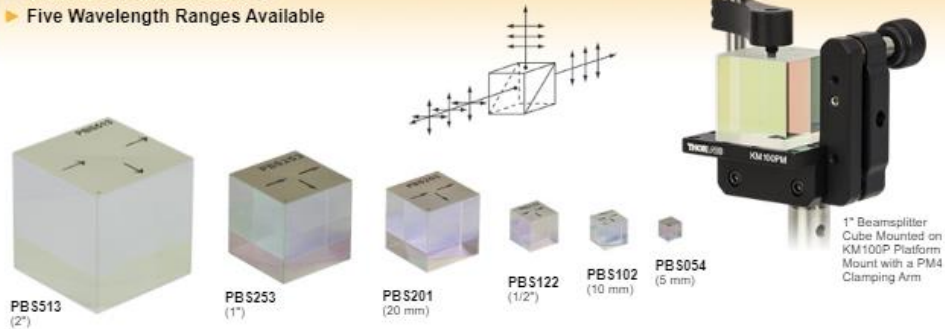
## PBS251 (Polarizing Beamsplitter Cube)

Products Home / Polarization Optics / Polarizers / Polarizing Beamsplitters / Polarizing Beamsplitting Cubes / Broadband Polarizing Beamsplitter Cubes



### Broadband Polarizing Beamsplitter Cubes

- Available in Sizes Ranging From 5 mm to 2" (50.8 mm)
- Reflects S Polarization by 90°
- Five Wavelength Ranges Available



#### Related Items



Overview Specs Graphs Damage Thresholds LIDT Calculations BS Cube Mounting BS Selection Guide Polarizer Guide Insights Feedback

#### Features

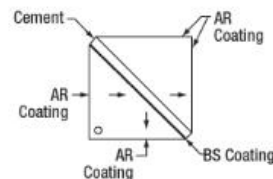
- 5 mm, 10 mm, 1/2" (12.7 mm), 20 mm, 1" (25.4 mm), and 2" (50.8 mm) Cubes
- 5 Wavelength Ranges Available:
  - 420 - 680 nm
  - 620 - 1000 nm
  - 700 - 1300 nm (1" Only)
  - 900 - 1300 nm
  - 1200 - 1600 nm
- Extinction Ratio
  - PBS519: Average  $T_P/T_S > 1000:1$
  - All Other Beamsplitters Below:  $T_P/T_S > 1000:1$

Thorlabs' Polarizing Beamsplitting Cubes are offered in six sizes and with five beamsplitting coating ranges. These cubes separate the s- and p-polarization components by reflecting the s component with the dielectric beamsplitter coating, while allowing the p component to pass. These cubes are designed to be used with the transmitted beam, which offers an extinction ratio of  $T_P/T_S > 1000:1$ , except the PBS519 2" 420 - 680 nm cube, which offers an average extinction ratio of  $> 1000:1$  over the wavelength range. The reflected beam will only have an extinction ratio of roughly 20:1 to 100:1, depending on the beamsplitter.

Thorlabs offers polarizing beamsplitter cubes in 5 mm, 10 mm, 1/2" (12.7 mm), 20 mm, 1" (25.4 mm), and 2" (50.8 mm) sizes. These cubes are made from N-SF1 or H-ZF3 glass and are offered in five different coatings for the following wavelength ranges: 420 - 680 nm, 620 - 1000 nm, 700 - 1300 nm, 900 - 1300 nm, and 1200 - 1600 nm. Please see the Specs tab for more information on each cube, including its damage threshold, or see the Graphs tab for S- and P-polarization transmission graphs.

The dielectric beamsplitting coating is applied to the hypotenuse of one of the two prisms that make up the cube. Then, cement is used to bind the two prism halves together (refer to the diagram shown above). The engraved dot on the top of the cube indicates the prism with the beamsplitting coating. Light can be input into any of the polished faces to separate the S and P polarizations. Cubes larger than 5 mm also feature engraved arrows indicating one possible orientation.

Please refer to the BS Cube Mounting tab above for information on mounting options and compatibility. Alternatively, our 1" cubes are available pre-mounted in cage cubes. Custom beamsplitter cubes can be ordered by contacting Technical Support. For high power applications, we also offer high power polarizing beamsplitting cubes. We also offer polarizing beamsplitter cubes at laser line wavelengths, which have a higher extinction ratio of 3000:1 ( $T_P/T_S$ ).



Cube Beamsplitter Diagram (Coating and Cement Layer Not to Scale)



[Click to Enlarge](#)  
12.7 mm Beamsplitter Cube Mounted in the CCM1-4ER Compact Cage Cube Using the BS127CAM Beamsplitter Adapter  
(Refer to the BS Cube Mounting tab for Other Options)



### 1" (25.4 mm) Polarizing Beamsplitter Cubes



Zoom

Item #	PBS251	PBS252	PBS255	PBS253	PBS254
Cube Size	1" x 1" x 1" (25.4 mm x 25.4 mm x 25.4 mm)				
Wavelength Range	420 - 680 nm	620 - 1000 nm	700 - 1300 nm	900 - 1300 nm	1200 - 1600 nm
Material	N-SF1				
Transmission	$T_P > 90\%$				
Reflection	$R_{S, Avg} > 95\%$				
Surface Quality	40-20 Scratch-Dig				

## LPVISE100-A (Polarizer)

Products Home / Polarization Optics / Polarizers / Film Polarizers / Economy Film Polarizers with Windows

### Economy Film Polarizers with Windows

- Polarizing Film Between Two AR-Coated N-BK7 Windows
- Designed for 400 - 700 nm, 600 - 1100 nm, or 1050 - 1700 nm
- Three Sizes: Ø1/2", Ø1", and Ø2"

Application Idea: LPVISE050-A Ø1/2" Polarizer in a PRM05 Optic Mount

Related Items:

- Rotation Mounts
- Wire Grid Polarizers
- Nanoparticle-Based Linear Polarizers
- Polarimetry

Overview | Damage Thresholds | LIDT Calculations | Feedback | **Polarizer Guide**

#### Features

- Dichroic Polarizing Film Between Two AR-Coated N-BK7 Windows
- Three Operating Wavelength Ranges Available:
  - 400 - 700 nm (A-Designation)
  - 600 - 1100 nm (B-Designation)
  - 1050 - 1700 nm (C-Designation)
- High Acceptance Angle:  $\pm 30^\circ$
- Custom Sizes Available; Contact [Tech Support](#) for Details

These glass polarizers have polarization efficiencies in excess of 99% (see the tables below for extinction ratio specifications) and are available in Ø1/2", Ø1", and Ø2" sizes. Featuring a dichroic film sheet, which allows transmission and absorption of polarized light over a specified wavelength range, these polarizers provide high absorption of the rejected polarization, making them ideal for low-power applications. A protective N-BK7 window is epoxied onto each side of the film. In addition, each window has one of three AR coatings deposited on the glass-to-air interface (see the tables below for AR coating specifications). The output polarization direction is marked with a line on the side of each polarizer (see schematic to the right). We also offer [2" x 2" sheets of the dichroic polarizer for 400 - 700 nm](#) without protective windows that are ideal for cutting custom sizes.

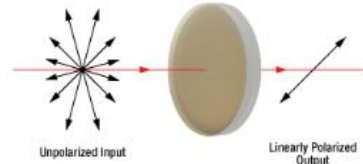
These polarizers are sensitive to stress when mounting. Overtightening the retaining ring can cause stress-induced birefringence in both the glass and the assembly and can also reduce the extinction ratio of the optic. To ensure that the polarizer is not loose in the housing, we recommend the use of our [Stress-Free Retaining Rings](#). The use of setscrew-based mounts is not recommended for these polarizers. While the polarizer's surfaces can be cleaned with normal solvents, take care to avoid the polarizer's edge.



Click to Enlarge  
One Ø2" polarizer linearly polarizes incoming light before a Ø2" Polymer Zero-Order Wave Plate converts it to an elliptical polarization state. Each optic is mounted in an LCRM2 cage rotation mount.

Common Specifications <sup>a</sup>	
Polarizing Material	Dichroic Film
Window Material	N-BK7 <sup>b</sup>
Clear Aperture	>90% of Diameter
Surface Quality	80-40 Scratch-Dig
Acceptance Angle	$\pm 30^\circ$
Operating & Storage Temperature	0 to 40 °C
Beam Deviation	<20 arcmin

a. For item-specific specifications, see the tables below.  
b. Click Link for Detailed Specifications on the Substrate



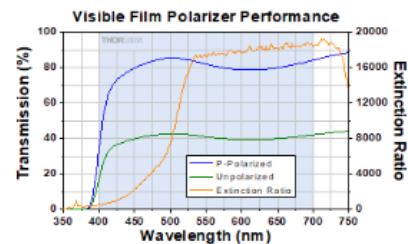
### Economy Film Polarizers, 400 - 700 nm



Item #	LPVISE050-A	LPVISE100-A	LPVISE200-A
Operating Wavelength Range		400 - 700 nm	
AR Coating Range		350 - 700 nm	
Reflectance over Coating Range (Avg.)		<0.5% at 0° AOI	
AR Coating Curve			
Extinction Ratio <sup>a,b</sup>		>100:1 (400 - 500 nm) >1000:1 (500 - 700 nm) >5000:1 (530 - 690 nm)	
Size	Ø1/2"	Ø1"	Ø2"
Thickness	2.1 mm	3.3 mm	6.5 mm
Dimensional Tolerance	$\pm 0.2$ mm		
Damage Threshold	1 W/cm <sup>2</sup> (532 nm, CW, Ø0.471 mm) <sup>c</sup> 0.07 J/cm <sup>2</sup> (532 nm, 10 Hz, 8 ns, Ø200 µm)		
Transmitted Wavefront Distortion	$\lambda$ at 633 nm		

- a. The extinction ratio (ER) is the ratio of maximum to minimum transmission of a sufficiently linearly polarized input. When the transmission axis and input polarization are parallel, the transmission is at its maximum; rotate the polarizer by  $90^\circ$  for minimum transmission.
- b. The extinction ratio is specified at 0° AOI and will vary slightly over the acceptance angle of the optic.
- c. The power density of your beam should be calculated in terms of W/cm<sup>2</sup>. For an explanation of why the linear power density provides the best metric for long pulse and CW sources, please see the [Damage Thresholds](#) tab.

These thin film polarizers, which are optimized for use in the 400 - 700 nm range, have an AR coating for the 350 - 700 nm range deposited on the air-to-glass interface of each window. They offer an average transmission of 38% for unpolarized light over their operating wavelength range. The direction of the output polarization is marked on the edge of each polarizer.



The graph above shows the measured transmission of unpolarized light as well as that of polarized light aligned with the polarization axis of the optic. The shaded region represents the specified operating wavelength range of the polarizer.