UNIVERSITÀ DEGLI STUDI DI TORINO

CORSO DI LAUREA MAGISTRALE IN FISICA DELLE TECNOLOGIE AVANZATE

VISIBLE LIGHT COMMUNICATION

Relatore:

Stefano Argirò

Co-relatore:

Antonio Orlando

Candidato:

Stefano Truzzi

Anno Accademico 2015/2016

Abstract:

The need to be always connected to the network and to have access to ever larger volumes of data has led to the improvement of the existing transmission technologies and the creation of new transmission techniques. One of these new techniques is the Visible Light Communication (VLC). It is based on the transmission of light pulses that belong to the visible spectrum. VLC employs a part of the electromagnetic spectrum that is currently unused for transmission purposes, this provides a huge bandwidth (390 THz). This technology is based on LEDs and photodiodes, devices which, although present for decades in the global market, are seeing only recently improved performance and wide spread. These considerations are leading many researchers to approach the VLC. In this work we first made a study to understand what the level of technology has been reached until now. In general the VLC systems could be divided in two macrocategories: the indoor systems and the outdoor systems. Both technologies have different possible applications, in particular the outdoor system can be derived from the indoor system. Furthermore in December 2011 the IEEE (institute of electrical and electronics engineers) completed the standardization of the physical layer. The physical layer is the first layer of the ISO/OSI model. It describes the system physical features like: means of transmission, construction of transmitter and receiver device and what is the modulation used to transmit data. One can conclude that in indoor systems the most used modulation for VLC are currently the ON-OFF Keying (OOK) and the Pulse Position Modulation (PPM). Furthermore with simple LOS (line of sight) channel model is possible understand, approximately, how the VLC channel propagates through the room. The second part of thesis is based on Matlab simulations, by which one can understand the quantities involved. The simulation involve the On-Off Keying and the Pulse Position modulation. For this two modulations the BER (Bit Error Ratio) curves were derived. A second simulation involve the LOS (line of sight) channel. The simulation estimate the light power received from a photodiode using the signal transmitted by a LED inside a room. Finally, to test and get feedback with simulated and analyzed data we preparec a prototype using Arduino. We built a simple model, which transmits data using a OOK modulation.

1 Introduction

In the last years the wireless technology has spread widely. One of the most important reasons lies in the need of people to stay always connected at high speed to the network. Whether for work or for fun more and more people have the necessity to be connected in any location at high speed. Furthermore with the increase of web technology the amount of transmitted data increased drastically.

Until today radiofrequency-based technology has been able to meet these requirements, but in the next future there will be a problem: the volume of transmitted data increases exponentially but the radiofrequency band is limited.

A possible solution to this problem is the optical wireless technology (OWC). Optical wireless technologies are new communication technologies where the information is transported through modulated beams of visible or infrared light.

OWC has different properties with respect to radiofrequency transmission [1].

	Property	Radio	Optical wireless
1	Bandwidth regulated	Yes	No
2	Pass through opaque object	Yes	No
3	Multipath fading	Yes	No
4	Multipath propagation	Yes	Yes
5	Path loss	High	High
6	Input X(t) represents	Amplitude	Power

Table 1.1: Radio – OWC comparison [1].

As one can see from table 1.1 the main advantage of OWC with respect to radio communications is that the OWC is not regulated, the entire OWC bandwidth is free and not licensed. That is a notable advantage for anyone that would design and commercialize OWC systems. Furthermore if one compares with the radio bandwidth with the VLC bandwidth finds that the VLC bandwidth is much bigger respect to radio as show in table 1.2.

Radio bandwidth:	VLC bandwidth (table 1.2.1):	Ratio : $\frac{VLC}{Radio}$
$Band \cong 300\ GHz$	$Band = 869\text{ THz} - 400\text{ THz} = \cong 470\ THz$	$\frac{490\text{ THz}}{300\text{ GHz}} = \frac{490 * 10^3\text{GHz}}{300\ \text{ GHz}} \cong \cong 1600$

Table 1.2: Radio – VLC comparison.

The VLC bandwidth is about 1600 times wider than the radio Bandwidth.
OWC, differently from radio, doesn't pass through opaque objects. This is an advantage in terms of security because the signal can be restricted to a confined area.
Another advantage is that OWC doesn't suffer from multipath fading which is the attenuation of the signal due to reflections and refractions (chapter 1.2.6).The last difference from radio is that the transmission in OWC is based on the wave power and not on the amplitude of the electromagnetic field. Therefore in OWC the received signal can be only positive meanwhile in radio signal the signal can be positive or negative.
The remaining properties in table 1.1 are in common: both radio and optical waves suffer from multipath distortion due to reflection of the signal; furthermore the path loss for OWC is proportional to distance square as in radio communications.

1.1 Work synthesis:

This thesis is structured as follows:

1) In the theoretical section (chapter 2) we analyze the theory of VLC indoor communication system, We start from the block diagram (chapter 2) and we describe every blocks of that, the transmitter block (2.1), the receiver block (chapter 2.2), the channel block (chapter 2.3) and the modulation block (chapter 2.4).
2) In the simulation section (chapter 3) we simulate the VLC direct (LOS) channel and two ways to modulate the VLC transmission.
3) In the prototype section (chapter 4) we build a small VLC prototype with Arduino to transmit Ascii code and to analyze BER (bit error rate)

4) In the applications section (chapter 9 and 10) we list some experimental and existing VLC application

1.2 Visible light communication

Visible light communications is a branch of optical wireless communication that involves electromagnetic waves in the visible spectrum. Many VLC communication applications that are described in the following are valid for OWC systems and vice versa.
As mentioned above VLC is a subset of OWC that uses visible light spectrum to communicate. With respect to OWC, VLC applications consider only the visible spectrum and not the infrared or ultraviolet spectrum.
In general VLC use commercial LEDs to communicate. As one can see in chapter 3, LEDs have a wide white wavelength emission spectrum that involves all the visible. In the following chapter the VLC LEDs, unless otherwise specified, are treated as devices transmitting monochromatic waves. This may seem as a very restrictive simplification but in practice it is not, because, thanks to the receiver photodiode (chapter 4) and lenses filter, one can consider the transmitter light as a sum of many single monochromatic waves.
To treat VLC systems one should introduce some general features:

1.2.1 Involved frequencies

All transmission systems use waves to communicate. In particular radio and optical communication use electromagnetic waves to transport information. The electromagnetic spectrum is very broad and is categorized by wavelength or by frequency. These quantities are closely linked by the formula:

$$\boldsymbol{\lambda = c/f}$$

Equation 1.2.1

Where c is the speed of light, $\boldsymbol{\lambda}$ is the wavelength an f is the frequency.
The VLC electromagnetic spectrum is in general divided in wavelength intervals that match approximately to colors perceived by the human eye as show in table 1.2.1.

Color	Frequency (THz)	Wavelength (nm)	
Ultraviolet (Near UV)	789 – 869	300 – 380	OWC
Violet	668 – 789	380 – 450	VLC
Blue	631 – 668	450 – 475	VLC
Cyan	606 – 631	476 – 495	VLC
Green	526 – 606	495 – 570	VLC
Yellow	508 – 526	570 – 590	VLC
Orange	484 – 508	590 – 620	VLC
Red	400 – 484	620 – 750	VLC
Infrared (Near IR)	400 – 214	750 - 1400	OWC

Table 1.2.1: OWC waves classification and value

1.2.2 Flickering

Flickering is a rapid variation of the intensity of the source. This phenomenon is always present in VLC light sources because the transmission method uses intensity variations to transmit the signal. However flickering denotes a light flashing that is perceivable from human eyes and causes distress

The flickering is perceptible from humans when the light source (chapter 2.1) does not meet certain conditions. Flickering is characterized by controllable factors like the light bulb blink frequency or the illumination intensity variation and uncontrollable factors like the people degree of light/dark adaptation, the age and fatigue [2].

1.2.3 Intensity modulation – Direct Detection (IM-DD)

In general, two methods of VLC transmission/detection are possible: intensity modulation – direct detection (IM-DD) and coherent transmission/detection.

The coherent transmission/detection is usually associated with laser diodes (LD). The IM-DD instead is associated with light emitting diodes (LED). This method is used because the LED, unless one uses complex and expensive systems, cannot generate coherent waves (chapter 3 examine LED).

IM-DD is a transmission scheme in which the intensity of the optical source is modulated by the signal, and the demodulation is achieved through direct detection of the optical carrier and conversion using a photo-detector (Chapter 4).
This work treats the IM-DD and does not consider the coherent transmission/detection.
The IM-DD VLC systems are cheaper and easiest to implement and the lasers on the other side are dangerous for human eyes. This is the reason which leads to consider LED IM-DD for commercial everyday applications.

1.2.4 Signal to noise ratio (SNR)

Signal to noise ratio (SNR) is the ratio between the signal power and the noise power. This dimensionless quantity is a very important parameter for all communication systems. This is a sort of quality factor for the communication system.
If the SNR is low the system will not transmit correctly, it is desirable to achieve the highest possible SNR but this implies the increase of the system cost, because raising the signal power means more power consumption and lowering the noise power means increased system complexity.
The best choice is in general a middle way, the SNR must be high as possible to transmit correctly but must be low enough to keep the cost of the system low.

1.2.5 Multipath propagation and fading

Both VLC and radio signals propagate in the medium and arrive at the receiver. The signal that arrives at the receiver is divided in two components: the direct component (line of sight, LOS) and the diffusive component (Not line of sight, NLOS). The diffusive component is due to the reflected part of the signal that is generated from the interaction with an obstacle. For example for indoor VLC a component of the signal hits the various walls and bounces to the transmitter.
Multipath propagation is due to the diffusive component of the signal.
A consequence of multipath propagation is fading, or the attenuation of the signal. Fading can also be due to shadowing form obstacles between the source and the receiver.

1.2.6 Peak to average ratio (PAPR)

Peak to average ratio (PAPR) is the ratio between the square of the peak and variance of the wave. The PAPR is a measure of how much the waveform is "large".

$$PAPR = \frac{{|x|_{peak}}^2}{x_{rms}^2}$$

Equation 1.2.6 Peak to average ratio

2 VLC physical layer

To characterize the communication system the International Organization for Standardization (ISO) defined the Open Systems Interconnection model(ISO/OSI model).
The ISO/OSI splits the communication system in a seven-level hierarchical structure listed in table 2.1 [3]:

OSI Model			
Layer		**Protocol data unit (PDU)**	**Function**
Host layers	7. Application	Data	High-level Apis, including resource sharing, remote file access
	6. Presentation		Translation of data between a networking service and an application; including character encoding, data compression, encryption/decryption
	5. Session		Managing communication sessions, i.e. continuous exchange of information in the form of multiple back-and-forth transmissions between two nodes
	4. Transport	Segment(TCP) /Datagram(UDP)	Reliable transmission of data segments between points on a network, including segmentation, acknowledgement and multiplexing.
Media layers	3. Network	Packet	Structuring and managing a multi-node network, including addressing, routing and traffic control.
	2. Data link	Frame	Reliable transmission of data frames between two nodes connected by a physical layer
	1. Physical	Bit	Transmission and reception of raw bit streams over a physical medium

Table 2.1: OSI model from wikipedia.

The first and lowest level of this structure PHY is the level on which is focalized this thesis. The physical level involves the modulation/demodulation, channel, transmitter and receiver definitions.

To characterize the physical level one must be able to draw a block diagram :

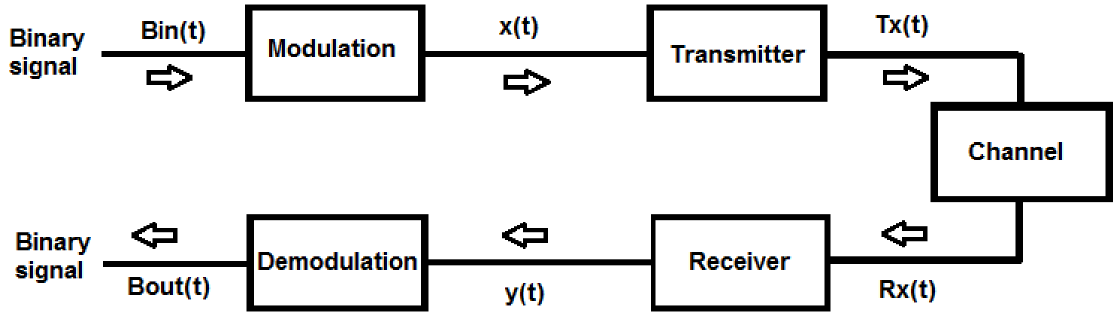


Figure 2.1: physical level block diagram

Four elements can be identified from figure 2.1:

a) Transmitter (Section 2.1)

b) Receiver (Section 2.2)

c) Channel (Section 2.3)

d) Modulation/Demodulation (Section 2.4)

Transmitters are devices that take the modulated signal and turn it into something that can be transmitted through the channel. In VLC systems transmitters are usually LEDs.
Receivers are the opposite of transmitters: these devices take the signal that arrives from the channel and convert it into signal that may be processed by the demodulator. In VLC systems the receivers are usually photodiodes.
The Channel is how physically the signal interacts and propagates with the environment. The channel takes the signal coming from the transmitter and calculates/simulates how the signal arrives at the receiver. The Channel isn’t associated with a device: it describes how the signal passes through the environment, for example an empty room, or from building to building.
Modulation and demodulation are complementary opposite blocks: the first takes the logical signal and transform it to something that can be transmitted through the transmitter. For example in VLC the modulation transforms the logical input signal in a current signal that can drive the LED.

Demodulation instead takes what comes from the receiver and reconverts it to a logical signal.

The above blocks will described in detail later in the following chapters.

VLC standard IEEE 802.15.7

Every technology has a standard with norm or requirement. In 2011 the Visible Light Communication Task Group completed PHY (physical) and MAC*(media access control) standards for VLC. The standard's name is IEEE 802.15.7.

The PHY is the level on which this thesis is focused, for VLC the IEEE has defined three standards: the PHY I, PHY II and PHY III. This are three types of physical standardization levels that are defined in IEEE standard table 2.1.1:

Standard	Modulation	Speed	Multi-Optical sources
PHY I	OOK,VPPM	$11.67\frac{kb}{s}\ to\ 266.6\frac{kb}{s}$	NO
PHY II	OOK,VPPM	$1.25\frac{Mb}{s}\ to\ 96\frac{Mb}{s}$	NO
PHY III	CSK	$12\frac{Mb}{s}\ to\ 96\frac{Mb}{s}$	YES

Table 2.2: Visible light communication system PHY standards.

**MAC is a lower sub-layer of data link layer that is the second level of the seven level mentioned in chapter 2 (Block diagram).*

2.1 Transmitter

In general transmitters are devices that are used to transmit the information through a medium. The transmitter takes the signal in input and creates, in output, something that can arrive at the receiver. The transmitter can be of many types, for example the radio techniques communicate with an antenna that create an electromagnetic field, the VLC communicate with a light source that can be varying its power, even the human throat could be considered as transmitter that generates an audible signal.

For VLC the transmitter should be a device that can manipulate the visible light to generate a signal that could be interpreted from a receiver device. In general as light transmitter the OWC technologies use two devices: light emitting diode (LED) and laser diode (LD). For VLC LEDs seems to be the best choice, respect to LDs the LEDs devices are cheapness and are not harmful for eyes.

2.1.1 LED

The LED is transducer: a device that is able to transform a form of input energy (i.e. Electrical) to another form of energy in output (i.e. Luminous).

In general a LED is a diode device composed of a p-n junction that emits optical radiations when it’s subjected to an electronic excitation that is induced by the application of a forward voltage bias across the junction (figures 2.1.1 and 2.1.2). The electrons acquire energy from the bias voltage and bounce to conduction band. Afterwards the electrons lose their energy combining with a hole. That energy generates optical radiation (photons) with energy proportional to energy gap of the semiconductor material used to build the junction. When the electron drops down to valence band, the phenomenon is called spontaneous emission because the de-excitement is not caused by an external factor but is just spontaneous.

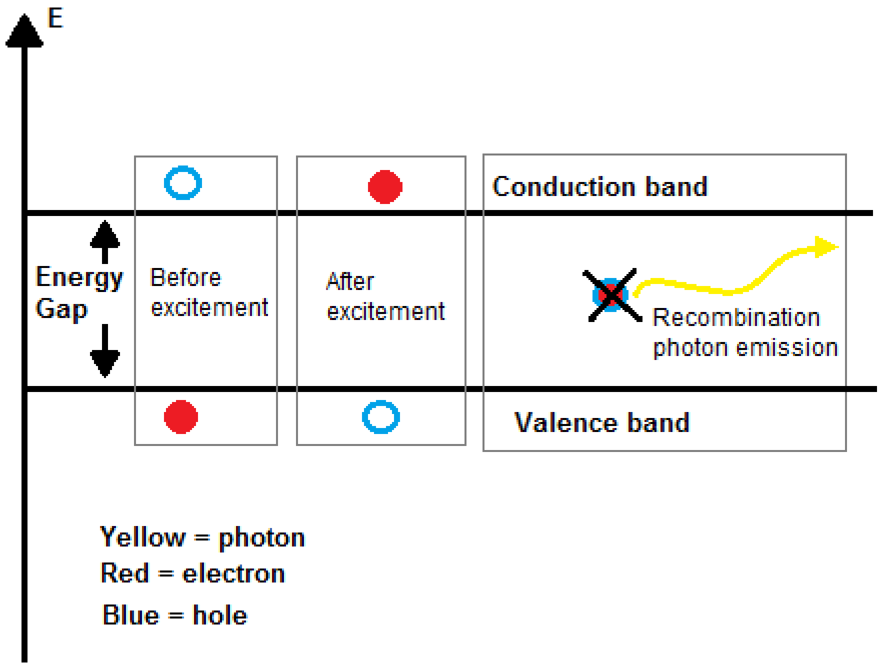


Figure 2.1.1 LED band model

$\Delta E = E_c - E_v = hf$ $\quad f = photon\ frequency \quad h = Planck\ constant$ **equation 2.1.1**

That optical radiation consists of electromagnetic waves that can be IR, UV or visible. LEDs in general are an incoherent source of light. Their output waves are spread through a big portion of EM spectrum (not monochromatic) and the direction of electric and magnetic fields that are generated from every photon are randomly distributed.

The LEDs have spread on market for their high efficiency in converting electrical energy into optical energy. Furthermore if the efficiency factor is high the heat dispersion in conversion should be very low.

Two big advantages of LED technologies are durability, LEDs has very long life cycle respect fluorescent lamps, and price, LEDs are very cheap and affordable. For all these reasons the LEDs are a good solution for future communications systems. All of these features are general properties of LEDs and are important not only for VLC technologies.

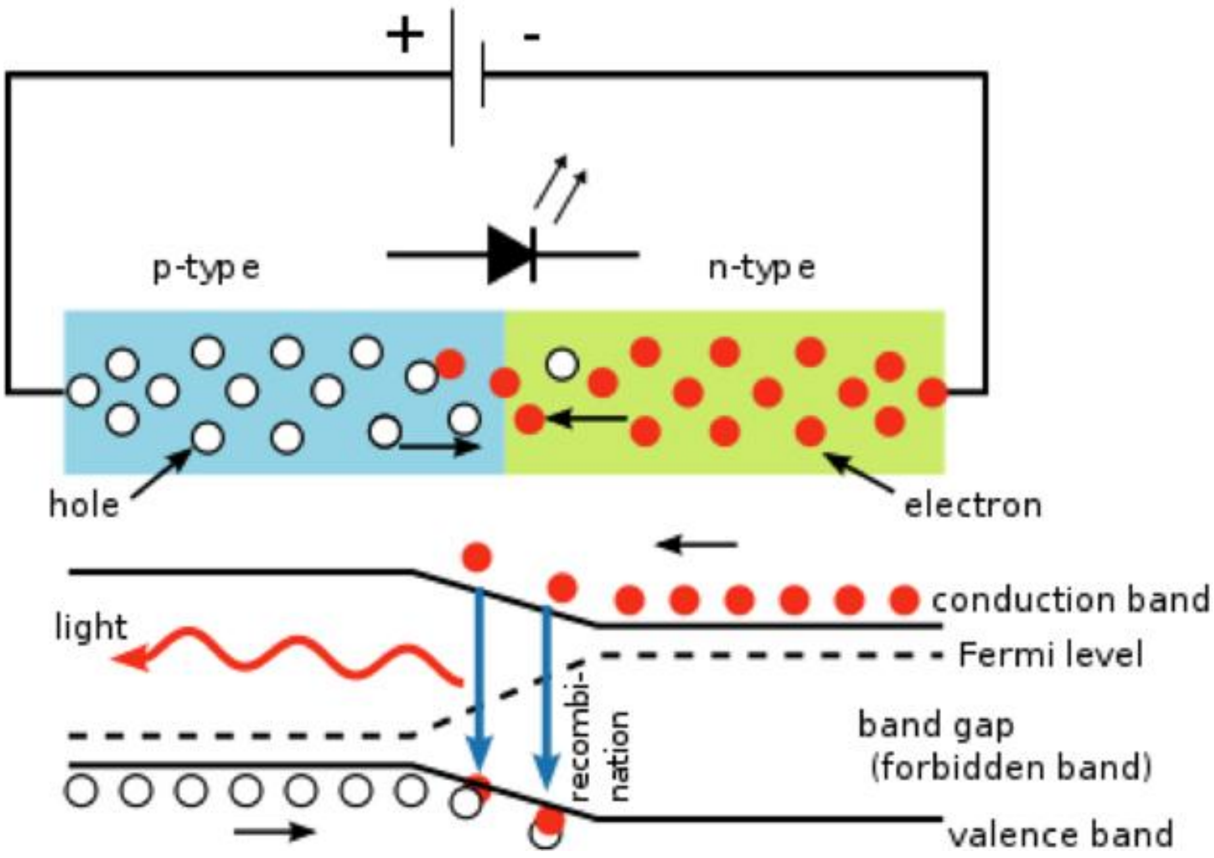


Figure 2.1.2 LED working principle

2.1.2 LED in OWC/VLC

LEDs have one more property that hasn't been mentioned in the previous: if desired a LED can be a fast intermittent source of light, this is the key to transform a LED into a transmitter.

The simplest transmission scheme can be achieved is by turning on and off the LED like a sort of Morse signal. If one for example associates the "on" light level to the number one and to the "off" light level the number zero one can represent a binary number 0 or 1 for all LED pulses.

For that example the speed with which the number is transmitted is directly related to the LED intermittence speed (or flashing frequency of LED).

One could say that if the LED transmits continuously this would cause a nuisance for the human eye in the form of flickering. However if the intermittence speed is sufficiently high, the human eye does not perceive it and high transmission speeds can be achieved.

Contemporary LEDs devices can reach few tens of Mhz of frequency. Furthermore this number is bound to improve because many of the companies that invested in VLC are attempting to improve the speed. This is a specific example the IM-DD technique.
LEDs also show a few downsides. A big issue for LEDs is that the light intensity response to a current signal is not linear.
This issue acquires particular importance with high peak to average ratio (PAPR) modulation/multiplexing techniques (ex. PAM, OFDM).
Figure 2.1.3 explains this point: the black lines down in the graph represent a modulation spectrum. If one of these lines cross in the LED non-linear zone the response is not linear and this causes an intermodulation interference.
This is inconvenient and can be avoided by not using modulation based on many amplitude level variations. Unfortunately this causes a loss of throughput.
A method to solve this problem is to use linearizer circuits but this greatly increases the entire system price and complexity.

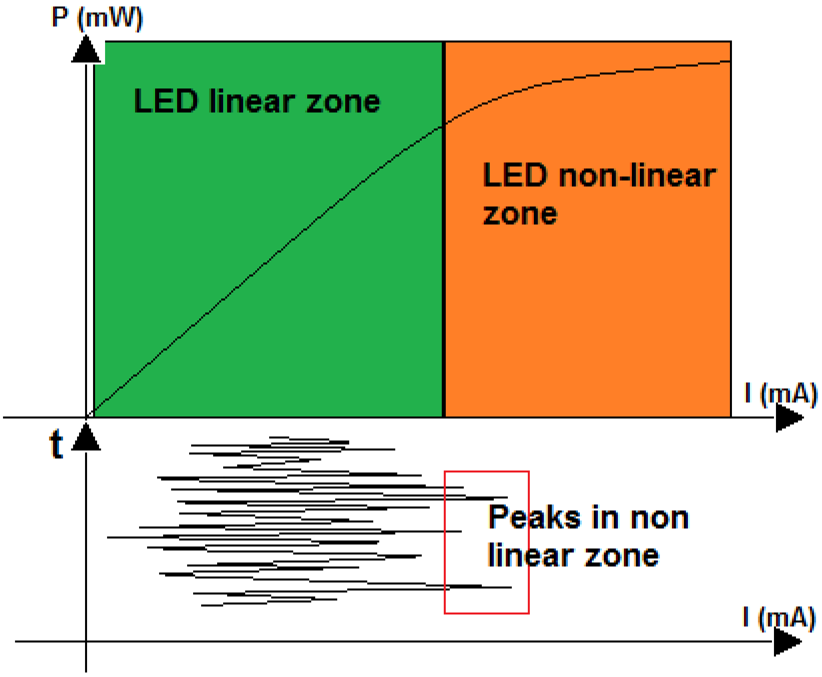


Figure 2.1.3 LED non linearity zone

For all the reasons listed before, in particular for simplicity, cheapness and for the incoherence of LEDs, these devices seem to be the best choice for the use of IM-DD transmission and vice versa the IM-DD seems the best choice to modulate the LED.

There are in general two types of LED that can be used: blue LEDs with yellow phosphorous, or three-colors LEDs.

1) Blue LEDs with yellow phosphorous are the most popular and economic white LED on the market. Unfortunately this LEDs type are the slowest one because the phosphorus reduces the response time. This LED can reach only few Mhz of bandwidth.
2) Three-colors LEDs (also known as RGB leds) are composed of three LEDs: red, green and blue. These LEDs are less widespread on the market then phosphorous-Blue LEDs and are more expensive. However the response time of each “sub-LED” is not slowed by the phosphorous. These LEDs can use each color as a single channel for a total of three channels. Each “sub-LED” can reach few tens Mhz.

The first types of LEDs are just used largely in everyday life. The second is less used and is more expensive.
The LEDs used for VLC in general are white LEDs because in general VLC technology designed to double use: environment illumination and communication. Theoretically one can use every LEDs colors to transmit with VLC but the environments users may be bothered.

2.1.3 VLC LED characteristics

If one wanted to design a LED IM-DD modulation circuit the most important characteristics (from [4]) that should be considered are:

1) Maximum frequency to which the LED can blink because it affects directly the bit rate and spectral efficiency (Sp_{eff}). For example if one transmits 2 bits at each pulse (0 is OFF and 1 is ON) and has a 10Mhz LED:
   $LED_{bitrate} = 10Mhz * bit = \frac{\mathbf{10Mbit}}{s}$ $\quad Sp_{eff} = \frac{\mathbf{10Mbit}}{s} * \frac{1}{Band\ Hz} = \frac{1bit}{s*Hz}$
2) Brightness is a very important parameter because it’s closely related to transmitted energy. Higher brightness means more transmitted energy. It can be found on LED’s datasheet.
3) Price. LEDs are cheaper than other light sources (light bulbes, fluorescent tubes, etc…)

4) Color is important to avoid interference. One can select the right color in a given environment. For example if you have solar light in your room you can avoid the yellow/orange color.
5) LED half angle power, the angle at which the optical power reduces to 50%. If this value was too small (less than 10 / 15 degrees), the light would be too localized and this would make the receiving too localized; meanwhile if this value was too big (greater than 75/80 degree) the light would be too widespread and the power received would be too small (see chapter 3 section 1).

Last small consideration is that is theoretically possible to transform the LED light into a monochromatic coherent source and use the coherent transmission/detection but is very expensive, for that reason this solution isn't taken into account.

2.1.4 VLC LED ideal equation and simple circuit

The Shockley diode equation: 2.1.2 represents the I (V) characteristics of an ideal diode in forward or reverse bias. This equation is an approximation and can be used to get a general idea of the diode generated current:

$$\mathbf{i_d} = \mathbf{I_s}\left(\mathbf{e}^{\frac{\mathbf{v_d}}{\mathbf{KT}}} - \mathbf{1}\right) \xrightarrow{KT<\mathrm{v_d}} \mathbf{i_d} \propto \begin{cases} v_d \to +\infty \ \textit{direct bias} \\ v_d \to -\infty \ \textit{reverse bias} \end{cases} \to \begin{cases} i_d \propto I_s e^{v_d} \\ i_d \propto -I_s \end{cases}$$

Equation 2.1.2: current generated by diode, Shockley formula. i_d is the current which is generated at the diode, I_s is the saturation current of the diode, K is the Boltzmann constant, T is the absolute temperature in Kelvin's grade, v_d is the voltage across the junction.

With that equation and some electronics knowledge one can draw the simplest circuit for LED diode shown in figure 2.1.2.

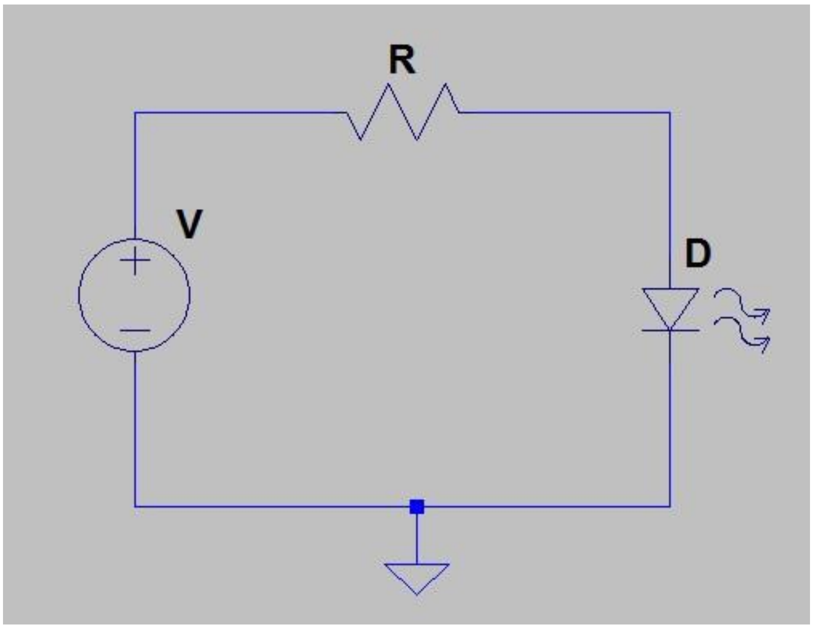


Figure 2.1.4 LED/diode example circuit

In 2.1.4 figure the diode is polarized directly. The expression of the current is derived in equation 2.1.3:

$$\mathbf{V = V_D + V_R \rightarrow \begin{cases} V_D = V_R = R * I_d \\ I_d = \dfrac{V - V_d}{R} \end{cases} \qquad I_d = I_s\left(e^{\frac{V_d}{KT}} - 1\right)}$$

Equation 2.1.3: V is voltage, I current, R resistance, the subscript letters are the corresponding circuit components

The LED current depends exponentially from the bias on p-n junction. Because of this, if one uses more than two light intensity levels to represent the symbols in order to increase the throughput (for example using N-PAM with N = 4 section 2.4.4) the effect of nonlinearity must be considered because this can generate an interference.

2.2 Receiver

To decode the transmitted signal one should be able to detect and discern its properties. In general this is possible with two methods: direct detection and coherent detection, both will be explained later.

For both techniques in VLC the transmitted signal is light wave. The device dedicated to detection of photon is called photodetector.

Photo-detectors are transducers that transform the received light (photons) into an electrical signal. Photo detector is a square-law detector because the current generated from it is proportional to the square of instantaneous optical field on its surface.

A particular type of photodetector that is used in VLC (and OWC) for its simplicity, dimension and for the low cost is the photodiode, that is an electronic device composed of a p-n junction. The photodiode general structure is the same as the LED, the difference is in the bias polarity: this time the bias applied isn’t direct but is zero or inverse. The reason is that the photodetector functioning is based on the depletion zone: the larger the depletion zone is, the larger the photo-detector sensitivity is. The depletion zone dimension is proportional to the inverse photodetector voltage bias starting from $V = 0$; at $V = 0$ the depletion zone dimension is based on the manufacture process. The choice of reverse bias is very important because it also increases the dark current that is one of the noise sources (see chapter on noise). Furthermore if a too larger bias is applied the diode goes in to breakdown region and no longer works as it should.

If a photon of sufficient energy hits the diode it creates an electron-hole pair. If this happens in the depletion zone the electron and hole are pushed out of the junction and give origin to a photocurrent. With this one can quantify the optical power through the measure of the current variation in the junction.

These devices are very popular on the market and, probably, their use will increase in the next future. Photodiodes are used in many field of common life. For example solar power, radiation detector, cellphone camera, etc.

For VLC systems there are several features of photodiode that make it particularly suitable [4]:

1) Fast response time: essential to receive and analyze the signal this quantity represents how long the photodiode employs to analyze the signal. This time

depends of the transit time of the photon/carriers in the depletion zone, on the photodiode circuit.

2) Low noise. In communication science the receiver is the place where the noise is analyzed. To simplify one can say that the noise depends on the photodiode quality but this isn't precisely correct. Noise considerations are many and very important because from that depends the correct reception, section 2.2.2 deals in detail with the various sources of noise and how to model it.
3) Low price is important because a receiver can be made of several photodiodes.
4) Wide wavelength range which must reach be at least equal to visible light wavelength range (400nm to 900nm).
5) Field of vision (FOV) must be larger than 25/35 degrees.
6) Small minimum resolution angle: this is how the PD can discern two light sources. If this value is too big, the PD cannot distinguish two neighbor LEDs. One can calculate this value with Rayleigh criterion: $\theta_R = 1.22\frac{\lambda}{d}$. Where λ is the incident wavelength, d is the diameter of the hole and θ_R is the minimum viewing angle.
7) High quantum efficiency. Representing how "good" is the photodiode, it's the ratio of how many electrons go out over how many photons come in: it is a conversion factor. Physically this parameter can be evaluated by a formula that contains a few parameters, illustrated in equation 2.2.1:

$$\boldsymbol{\eta_{qe} = \frac{electrons\ output}{photons\ input}} \qquad \boldsymbol{\eta_{qe} = (1 - R)\xi\left(1 - e^{-\alpha d}\right)}$$

Equation 2.2.1 quantum efficiency. R is the reflection coefficient between the air and the semiconductor, ξ is the e-h fraction that contributes to the photocurrent, α is the absorption coefficient of the PhD material and d is the absorption distance in the material.

8) Good Responsivity: this is another important parameter for the receiver photodiode. This parameter is similar to quantum efficiency in facts, it depends on it. Similar to quantum efficiency this parameter represent how "good" is the photodiode. The photodiode responsivity (equation 2.2.2) is the multiplicative factor that indicates how much light, expressed in luminous watt, are converted to electrical current, expressed in ampere.

$$\boldsymbol{R = \frac{\lambda q \eta_{qe}}{hc} \rightarrow R\left[\frac{A}{W}\right]}$$

Equation 2.2.2: definition of Responsivity. R is responsivity, λ is the wavelength, q is the electron charge, h is the planck constant, c is the light speed, η is the quantum efficiency.

2.2.1 Direct Detection

Direct detection is the detection of the transmitted envelope of optical power. Unlike common radio techniques this tech doesn't need a local oscillator and it's simple and economic. Unfortunately this technique has only one degree of freedom (less throughput): the only possible modulation variation to transmit the information is through alteration of light intensity (equation 2.2.3). The phase can't be considered because the transmitted light isn't coherent. Therefore if the detector is hit by an incoming optical radiation with P_r average power one can estimate the electric photocurrent generated by detector through equation 2.2.3:

$$\boldsymbol{i = \frac{\lambda q \eta_{qe}}{hc} P_r(t) = R P_r(t)}$$

Equation 2.2.3: Intensity law, photocurrent generated from photodiode. Where q is electron charge, h is Planck constant, c is light speed, η is the quantum efficiency, R is the responsivity and λ is the wavelength.

2.2.2 Noise

If one wants to transmit a message he must consider that the message that arrives at the receiver suffers of variations. In general these variations are due to various factors that can derive from different causes. All of these factors and variations are studied and modelled in noise theory.

In general every type of noise follows a stochastic law and usually this can be approximated as a distribution. In general the VLC noises are treated as shot or Gaussian, the first type follows a Poisson discrete distribution and the second type follow a normal Gaussian distribution. If the number of observables (λ) is high (λ>1000) the Poisson distribution can be approximated as normal Gaussian distribution [5].

Four source of noise can be identified as relevant for VLC, of which three of shot type and one of Gaussian type [6], as discussed in the following.

Photon fluctuation/Quantum noise

An ideal photodetector is affected by the quantum nature of light: the number of photons emitted from optical source in a fixed time is never constant. The rate of photons follows the Poisson distribution.

The noise generated from this takes the name of photon fluctuation or quantum noise.

The photon fluctuation noise is a shot (Poisson) noise present in all photon detectors. The shot noise variance is given by equation 2.2.4:

$$\boldsymbol{\sigma_q^2 = 2q\langle i\rangle B}$$

Equation 2.2.4: Photon fluctuation noise. q is the charge quantity in Coulomb (in general this is the electron charge), $\langle i\rangle$ is the mean current of arrival signal, B = Δf is the noise frequency bandwidth and σ_q^2 is the quantum noise variance.

Dark current noise

The dark current is current present in the photo-detector even when placed in a dark room, caused by thermal effect that can send electrons into conduction band. This current depends on the manufacture process, in particular it depends on the material and its energy gap. Furthermore the dark current depend also from reverse bias, if one increases the reverse bias the sensibility of photo-detector increase and more dark current can be detected.

The dark current can be evaluated with the shot noise variance as in equation 2.2.5:

$$\boldsymbol{\sigma_d^2 = 2qI_d\Delta f}$$

Equation 2.2.5: Dark current noise variance. Sigmad is the variance, q is the current charge (electron), Id is the dark current of device (on Datasheet), Δf is the occupied bandwidth.

Background radiation noise

The background noise is the noise due to the environment factor for example sun or sky luminosity. This noise in general depends on intensity of the external source, the intensity depend in general on wavelength spectrum: a good example for this is the sun that emits more power on yellow/orange wavelength or the sky that emits more power on sky-blue wavelength.

The background noise is a shot noise that can be estimated with 2.2.6.

$$\sigma_{bg}^2 = 2q\Delta f R I_{env} \quad I_{env} \propto W(\lambda)$$

Equation 2.2.6: Background noise. q in general is the electron charge, Δf is the bandwidth, R is the photodiode responsivity, I_{env} is the intensity of the environment source. W is spectral radiance of background noises sources (ex. Sun or sky).

Thermal noise

Thermal noise is noise due to particles thermal agitation, in electrical case, the particles are electrons that form current, the temperature influence the movement of that electrons. Thermal noise can be modelled as white Gaussian noise. This noise prevails at low frequency.

Thermal noise it known as Johnson noise and can be estimated through the formula 2.2.7:

$$\sigma_{th}^2 = \frac{4k_b T_e \Delta f}{R_L}$$

Equation 2.2.7: Where k_b is the Boltzmann constant, T_e is the environment temperature, Δf is the frequency bandwidth and R_L is the circuit resistor.

Noise considerations

To analyze VLC system one should consider the four noise sources previously described. The VLC waves frequency goes from about 400THz to 800THz, at this frequency one can neglect the thermal noise because the shot noise is much larger respect thermal at very high frequencies.

$$\begin{cases} \boldsymbol{\sigma_q^2 = 2q\langle i\rangle \Delta f} \\ \boldsymbol{\sigma_d^2 = 2qI_d\Delta f} \\ \boldsymbol{\sigma_{bg}^2 = 2qRI_{env}\Delta f} \end{cases} \xrightarrow{yields} \boldsymbol{\sigma_{tot}^2 = 2q\Delta f(\langle i\rangle + I_d + I_{env})} \qquad \boldsymbol{\sigma_{th}^2 = \frac{4k_bT_e\Delta f}{R_L}}$$

Equation 2.2.8: Noise considerations

The dark current noise can be neglected because in general: $\mathbf{I_d} \ll \langle \mathbf{i} \rangle$ e $\mathbf{I_d} \ll \mathbf{I}_{env}$To simplify the model, and without loss of generality, researchers tend to consider the background noise dominant. This can be a valid approximation in rooms with little background light, for example a room with one or more windows that are not directly exposed to sun. If this simplification is valid one can make other assumption: first the background noise can be considered constant respect the received light power, because it is very slowly variable, second one can approximate the shot background noise as white Gaussian for large number of events.

$$SNR = \frac{(RP_r)^2}{\sigma_q^2 + \sigma_d^2 + \sigma_{bg}^2 + \sigma_{th}^2} \xRightarrow{only\ background} \frac{(RP_r)^2}{\sigma_{bg}^2} = \frac{(RP_r)^2}{2qRI_{env}\Delta f}$$

Equation 2.2.9: R is the photodiode responsivity, P_r is the received power and every σ^2 represent different type of noise.

2.3 Channel

To simulate VLC systems one of the most important features is the channel block. The channel represents how the transmitter light ray moves through space to arrive at the receiver.
The two main components to analyze VLC channel are the line of sight transmission (LOS) and the not line of sight transmission (NLOS). The NLOS transmission generally is given by rays which are reflected on obstacles and, after some bounces, hit the detector.
As mentioned before to model the channel a starting approach is to consider the VLC wave as monochromatic.

2.3.1 General VLC channel considerations

The following table lists all the features that one should consider when trying to implement a channel simulation block for VLC.

1) System environment and location
2) Transmission/reflections emission
3) Wavelength dependence
4) Number of reflections

System environment and location

The location of the system and the surrounding environment are the first elements to be considered where investigating VLC technology.
VLC is not suitable for long distance communication because the transmitted signal can't pass the opaque objects. (e.g. walls, buildings etc…).
The fact that the VLC signal doesn't penetrate walls is not only a negative property because this permits the signal to be confined in a closed room. This capacity facilitates security procedures.
Second consideration about the system location is in that VLC systems are based on light intensity modulation. Therefore each environmental light may be the cause of serious interference.
The VLC light can provide illumination to the surrounding environment.
The easiest condition to analyze and simulate is case of a short range, indoor, closed room, system. Also closed room can be empty or filled with furniture and windows. It depends on

how one wants the system to be accurate and depends on how many calculation resources one has available. The systems just described will be analyzed later in detail.

Transmission/reflections emission

To model and simulate the VLC channel, one should be able to quantify the energy received from the detector. So one has to understand what is the intensity law (equation 2.2.3) that characterizes VLC systems.
To decide accuracy of the model and available computing resources one can choose the level of detail in describing:

1) how light generated by a source is emitted in space
2) what happens when light is reflected by an obstacle

At first order one can say that the energy that reach a receiver is proportional to the light generated by a source. Second one may note that the receiver, in a second time, may become a source of reflected light. The transmitter emission can be hence modeled in three different ways depending on the type of source/reflection surfaces:

1) Lambertian source propagation (ideal diffuse)
2) Specular source reflection
3) Mixed source propagation (diffuse with directional component)

Lambertian (diffuse ideal) emission law

The Lambertian emission law says that the luminous intensity emitted from an ideal diffusely source detected from an observer is directly proportional to the cosine of the angle between the surface normal and the direction of the observer. This means that the light does not have a preferred direction of reflection. Lambertian surface hence has the same radiance when viewed from any angle, where radiance is the radiant flux emitted, reflected, transmitted or received by a surface, per unit solid angle per unit projected area.
This process can be iterated for subsequent reflections: the new observer becomes an ideal diffuse light source with intensity equal to previous received power and, the new observer (receiver), can be considered in another place.
This seems to be the best solutions for closed room VLC systems for the simplicity and because the majority of the objects in rooms can be approximated as Lambertian surfaces.

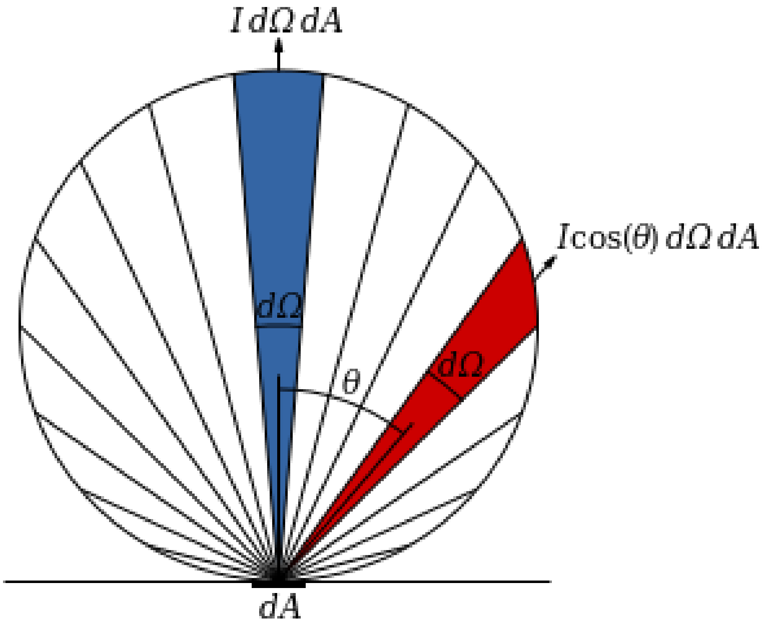


Figure 2.3.1: Reflection from a diffusively reflecting (Lambertian) [7].

Specular (ideal) emission

Specular reflection: is "the mirror-like reflection of light from a surface, in which light from a single incoming direction (a ray) is reflected into a single outgoing direction".

In simple terms all the light is reflected (generated) in a single direction that depends only from its arrival direction.

This approach is not good for general VLC closed room systems because object with mirror-like surface are very few.

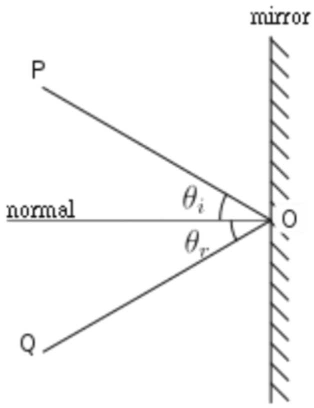


Fig. 2.3.2 specular,mirror-like reflection [8]

Mixed ("real") propagation

This is the most realistic propagation model. The light sources are usually both Lambertian and specular sources. When a ray of light hits an object the reflection direction has a preference (specular), but is not the only direction that the light will follow (Lambertian). Dealing with this mixed condition is very complex. In table XXX one can see that the only approach that uses this condition relies on a dedicated program.
Someone may think that this approach is better than the Lambertian, certainly is more accurate but it's very complex and it requires very high computing resources.
Therefore the best solution is to consider how one must be accurate in relation to the computing resources available.

Wavelength dependence (reflectance model)

When waves hit objects the intensity with which they are reflected depends on the wavelength of the arrival wave. For infrared systems the reflection is approximately constant because the IR communication experiments usually use monochromatic rays. VLC systems use white light rays with wavelength range between ≈ 350nm and ≈750nm. This should be considered for the simulations.
There are in general several ways to proceed:

1) Fixed reflectance: this type of model derive directly from infrared wireless system. Light is considered as monochromatic wave with fixed lambda and fixed reflectance. This model fits a very simplified scenario.
2) Three waves fixed reflectance: one considers light as a sum of three separated monochromatic colored waves and studies the three cases separately.
3) Wavelength-dependent reflection: this is the best and most accurate model that can be implemented. It considers the reflection depending on the value of the wavelength of the hitting wave. Very high computing resources are required.

Number of reflections

Analytical calculation of the reflections bounce is very complex. The algorithms which can calculate the reflections are few and are in general based on Monte Carlo or on recursive method of simulations. In general the algorithms that take in account several reflections use very high computing resources.

The best algorithm we found is from [9].

k	H_0
0	4.825×10^{-5}
1	6.030×10^{-5}
2	7.068×10^{-5}
3	7.673×10^{-5}
4	7.714×10^{-5}
5	7.714×10^{-5}
6	7.714×10^{-5}
7	7.714×10^{-5}
8	7.714×10^{-5}

Table 2.3.1 from article [8] estimated LOS and reflection alghoritm result. H0 is the DC gain for different k reflection. The column H0 is incremental: the second row H0 is the sum of the K=0 and K=1 contributions the third row is sum of K=0,1,2 etc…

Table 2.3.1 is obtained through a ray tracing algorithm based on Zemax, a commercial optical and illumination design software that costs about $4440 for standard license. That table is very significant because shows that the high order reflections (>4) are negligible for VLC and IR channel. Furthermore the difference from four and three order is very small and one can consider only three order of light reflection without lose generality.

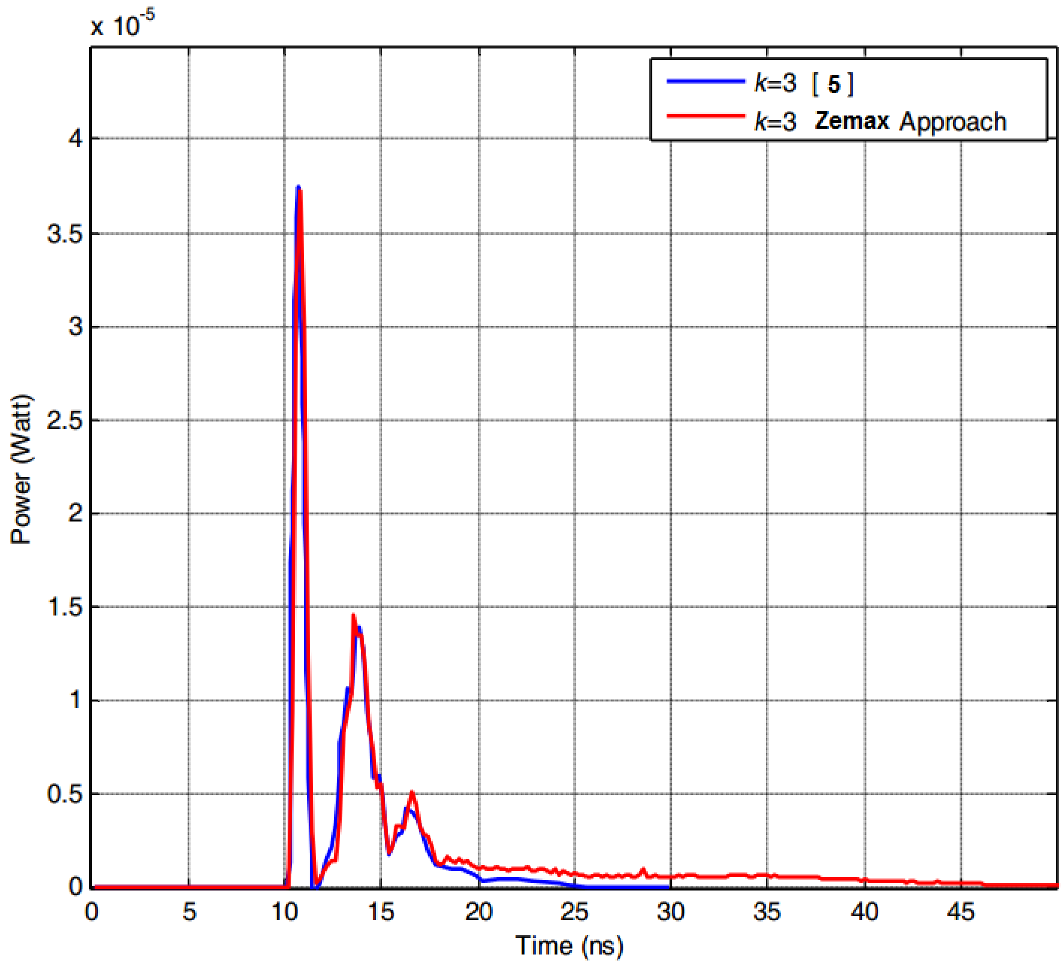


Figure 2.3.3 from article [9]. Differences between Zemax approach algorithm ray tracing and [7] ray tracing algorithm. [7] algorithm is based on [6] but for visible light. k is the number of bounces.

The figure 2.3.3 from [9] compares the Zemax and a general ray tracing ([10] and [11]) algorithm and shows that the differences are negligible.

Because differences between Zemax approach algorithm and [10] algorithm are negligible (picture 2.3.3), one can conclude that buying a Zemax license is useless and the algorithm used in articles [10] are sufficient.

2.3.2 Generality of indoor channel

The IM-DD technique is the most used method to communicate with VLC for its low cost and for its simplicity.

Typical VLC use LEDs to communicate with photodiodes: the modulating signal is the current signal $m(t)$ that passes through the LED that generates the optical power output $x(t)$. In simple words the modulation is based on the light power generated from LED.

The receiver is a photodetector that produces a photocurrent $y(t)$. The photocurrent (square law detector) is directly proportional to the optical power that impinges the photodetection surface.

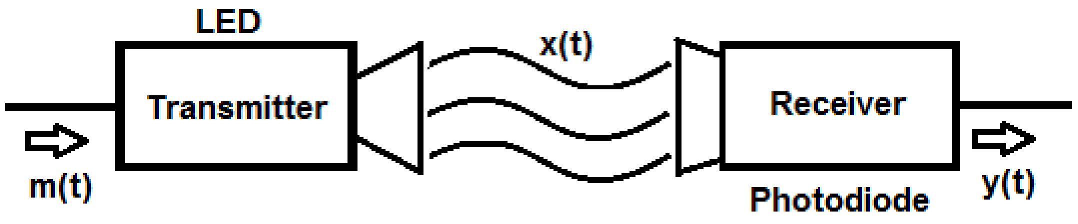


Figure 2.3.4 Block diagram of an optical intensity, direct detection communication channel. Where m(t) is the LED's current, x(t) is the optical power, y(t) is the photocurrente generated from photodiode.

IM-DD VLC system has an equivalent baseband model with which it's possible (Section 2.4.1) to introduce a few simplifications.

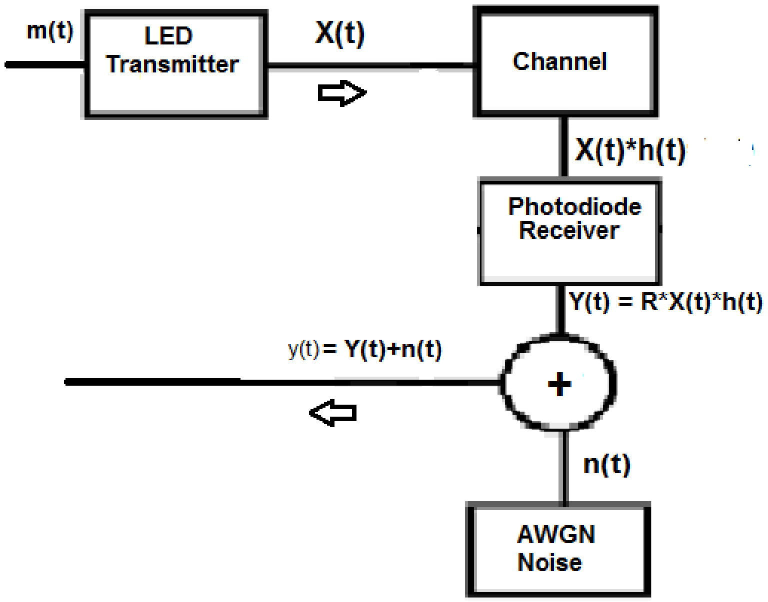


Figure 2.3.5 Equivalent baseband model of a Visible light communication using IM-DD. Here R is the photodiode responsivity, m(t) is the LED's current, h(t) is the baseband channel impulse response, X(t) is the optical power, Y(t) is the generated photocurrent, n(t) is the noise (see chapter 2 section 2.2).

As for the radio communication systems for VLC the noise is situated immediately before the receiver and this can be modelled as a double sided white Gaussian noise with power spectral density $PSD = \frac{N_0}{2}$.

An indoor VLC channel in general should work with two components: the LOS channel and the diffusive channel. The first is composed by all the line of sight rays that hit the photodiode without bouncing on other objects; the second is the sum of all light rays that bounce on the objects in the room (walls are consider as objects) and, in general, it's called non line of sight (NLOS) and is very important for very fast VLC systems.

$$P_r = \left(H_{los}(0) + H_{nlos}(0)\right)P_t = \left(H_{los}(0) + \sum_{refl} H_{refl}(0)\right)P_t$$

$$H_{VLC}(f) = H_{los} + H_{nlos}(f)$$

Equation 2.3.1 Power and channel for a VLC IM-DD LOS and NLOS channel.

NLOS links, in particular indoor VLC systems, are affected by the multipath propagation, similarly to RF, but in VLC this effect is more pronounced.

The electric field fading that is caused by the multipath propagation is instead irrelevant because the light wavelength is much smaller then the dimension of the detector. It creates a sort of antenna spatial diversity and in addition the photocurrent is proportional to the integral of the optical power on the detector surface, that contributes to the disappearance of the electrical field fading.

The baseband model for an IM/DD in mathematical formula can be expressed as in eq.2.3.2:

$$y(t) = Rx(t) \otimes h(t) + n(t) = \int_{-\infty}^{+\infty} R * x(\tau)h(t-\tau)d\tau$$

Equation 2.3.2 baseband model formula. R is the photodiode responsivity, x is the optical power and h is the channel impulse. ⊗ represent the convolution product.

Another significant difference from radio transmission is that in VLC the transmitted signal represents a power (optical) and not an amplitude as in the RF case (see chapter 1). This involves two considerations on the transmitted signal: first the signal x(t) must be positive and second, for eye safety, the maximum optical power must be limited.

$$x(t) \geq 0 \qquad\qquad P_{max} = \lim_{T\to\infty} \frac{1}{2T} \int_{-T}^{T} x(t)\,dt$$

When one designs a VLC system he must take in account this fact and the consequences: in radiofrequency system the SNR is proportional to average received power, in VLC system the SNR instead is proportional to the square of the average received optical power as in equation 2.3.3.

$$SNR_{VLC} = \frac{R^2 H^2(0) P_r^2}{N_0} \qquad H(0) = \int_{-\infty}^{\infty} h(t)dt$$

Equation 2.3.3 SNR for a VLC and OWC system see article [4]. Rb is the bitrate, N0 is the noise spectral density, Pr is the received power of the photodiode, H(0) is the channel DC gain and R is the photodiode responsivity.

Another important consideration for the indoor channel concerns the total time that the signal needs to reach the receiver. The pulses do not arrive all at once but in a certain time period, the time necessary for light to propagate from the first component of the LOS signal to the last component of the NLOS. The NLOS time can be modified according to the ISI that the design allows. Section 2.3.4 show some example of how the pulse arrives at the receiver.

Two important parameters for VLC channel consideration are the root mean square delay spread and the mean delay spread: these parameters are used to define the channel time-dispersion and they are define as in eq. 2.3.4.

$$D_{rms} = \left[\frac{\int (t-\mu)^2 h^2(t)dt}{\int h^2(t)dt}\right]^{\frac{1}{2}} \qquad \mu = \frac{\int t h^2(t)dt}{\int h^2(t)dt}$$

Equation 2.3.4 Root mean square delay spread on left and mean delay spread on right

Broadly speaking the mean delay spread for empty room VLC system oscillates from about 15ns to 35ns, and about 20ns to 40ns for IR system.

These parameters are connected to the room dimension, the wall material and the light wavelength; it is better to have a small value for these two parameters because these are a limit (together with photoreceiver sensibility) the speed of transmission.

2.3.3 Line of sight (LOS) VLC indoor channel model

Ref [12] derives the first and simplest case that one should consider is LOS (line of sight) channel, without the reflecting contribute, with a point Lambertian light source and constant received irradiance. This can be achieved by imposing the condition that the transmitter-receiver square distance (d) is larger than receiver surface (A_r): $d^2 \gg A_r$.

The LOS systems picture shown in Figure 2.3.6

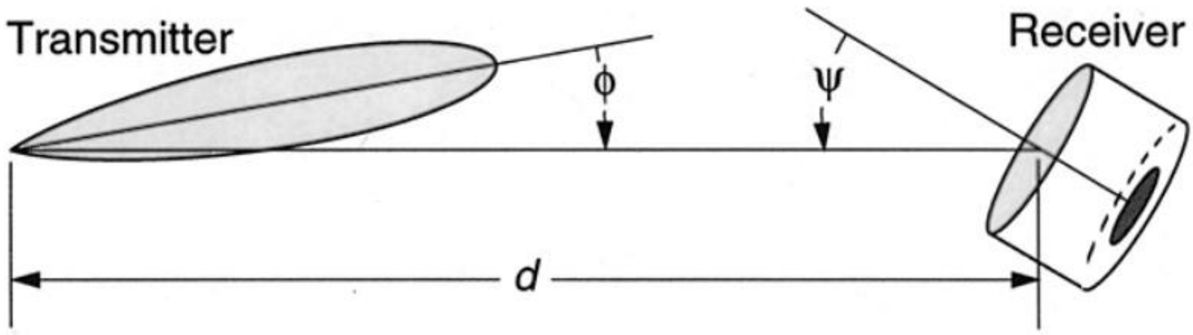


Figure 2.3.6: transmitter to receiver LOS path from [12]

The approximation used before brings to constancy of the luminous flux on A_r:

$$d^2 \gg A_r \rightarrow \int_{A_r} P_r * \hat{n}\, dA_r = P_r * A_r$$

Equation 2.3.5: constancy of the luminous flux implyes simplification of the area integral

The angular distribution of the optical radiation could be defined by Lambertian radiant intensity:

$$R_0(\phi) = \begin{cases} \frac{(m_l + 1)}{2\pi} cos^{m_l}(\phi) & for \quad \phi \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right] \\ 0 & for \quad \phi \geq \frac{\pi}{2} \end{cases}$$

Equation 2.3.6 where m_l is the Lambert's mode number a way to describe the directionality and ϕ is the angle between the led normal and d (vector that connect transmitter and receiver).

The Lambertian mode is related to LED semiangle, this parameter can be found on the LED datasheet:

$$m_l = \frac{-\ln 2}{\ln\left(cos\phi_{\frac{1}{2}}\right)}$$

Equation 2.3.7: calculation of Lambertian order

The effective area of receiver photodiode is based on the receiver surface inclination. One should project the surface on the axis d that connects transmitter and receiver.

$$A_{eff}(\psi) = \begin{cases} A_r cos\psi & 0 \leq \psi \leq \frac{\pi}{2} \\ 0 & \psi > \frac{\pi}{2} \end{cases}$$

Equation 2.3.8: receiver surface projection A_r is the area of the receiver photodiode, ψ is the angle beetween the photodiode normal and d.

To evaluate the LOS channel one should multiply all the factors found until now to obtain:

$$H_{LOS}(0) = A_r \frac{(m_l + 1)}{2\pi d^2} cos^{m_l}(\phi) T_s(\psi) cos(\psi)$$

Equation 2.3.9: $H_{LOS}(0)$ DC LOS gain. Where m_l is the Lambert's mode number, ψ is the angle beetween the photodiode normal and d, ϕ is the angle beetween the led normal and d.

This is the channel gain, now one can simply multiply the LOS channel gain by the input optical power to obtain the power on the receiver:

$$P_r = H_{los}(0) P_t$$

Equation 2.3.10: P_r is the received power, $H_{LOS}(0)$ is the DC LOS gain, P_t is the transmitted power.

To express the impulsive optical gain it is necessary multiply DC LOS gain with a Dirac's delta:

$$h_{los}(t) = A_r \frac{(m_l + 1)}{2\pi d^2} cos^{m_l}(\phi) T_s(\psi) cos(\psi) \delta\left(t - \frac{d}{c}\right)$$

Equation 2.3.11: impulsive optical power gain.

From the considerations in chapter 3 section 1 a numerical algorithm is derived.

2.3.4 Non line of sight indoor model

In indoor VLC systems the LOS path isn't the only path signal through which arrives at the receiver: there is another path that is generated from the light rays that bounce on the wall and furniture of the room.

A more complex model can include NLOS to increment the arrival power or to overcome the lack of LOS channel.

The NLOS channel is more complicated to model. In particular it requires significantly more computing resources.

The first consideration that has to be done for the NLOS channel regards the arrival time of the impulse. The impulses do not arrive all at the same time because the beam path isn't the same for all the rays. If one transmits a led pulse and measures the arrival time should obtain a response as the one shown in Figure 2.3.7:

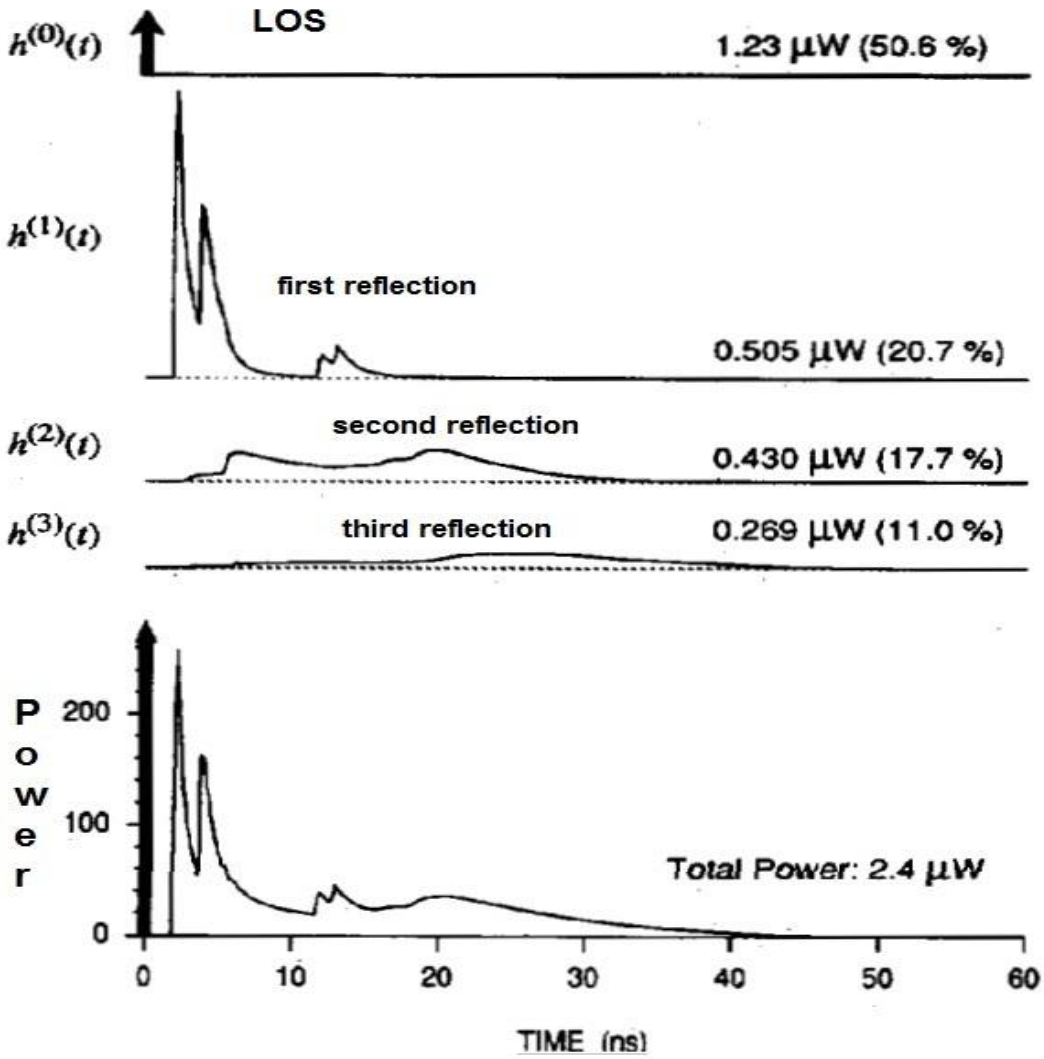


Figure 2.3.7:(LOS+NLOS channel as a function of time in a room of x = 5m, y = 5m and z = 3m. The led transmitter is in the centre of the ceiling, the receiver is at x = 0,5m, y = 1m , z = 0m (z starting from the ceiling). The x-axis is the time, the y-axis is the channel at istant t and the number on the area under the graphic is the power of the channel [10].

The graph 2.3.7 represents the sum of the LOS path (the thick black line on the left) plus one, two and three bounce channels.

The received power on the photodiode can then be expressed with the sum of the LOS and NLOS channel gain multiplied with the power transmitted:

$$P_r = \left(H_{los}(0) + H_{nlos}(0)\right)P_t = \left(H_{los}(0) + \sum_{refl} H_{refl}\,(0)\right)P_t$$

Equation 2.3.12 total power received from LOS + NLOS path.

One of the most used algorithm to model the NLOS path of VLC (OWC) system is presented in [10]. The condition of use of this algorithm involves a closed empty room with Lambertian reflective walls. For an approximation this condition is not so constrictive.
Now the key is to consider the light as a sum of various infinite rays that bounce through the room, the sum of these rays is the LOS plus the NLOS gain path.
One ray is generated from the source element and hits a receiver element, the source and receiver element can be not only a LED and a photoreceiver, but also the wall. For example if one takes one ray that is reflected from the wall,: in a first time the wall is considered as receiver and in a second time the wall receiver element can be considered as a transmitter with power proportional to received power multiply the wall reflection coefficient. This can be applied at every ray for every reflection.
To represent one ray one needs the path that this ray runs from the led to the photoreceiver, after that the path can be viewed as a sum of “sub-rays” (ex. From led to wall ray plus from wall to photodiode ray), each of these rays can be represented with a source, a receiver, a path and a pulse.
Therefore the impulse response can be modelled as the sum of all the sub-rays above mentioned:

$$h(t, S, R_x) = \sum_{k=0}^{\infty} h^k\,(t, S, R_x)$$

Equation 2.3.13 one ray path with infinite multiple bounces, k is the bounce numbers, t is the time, S is the source point, R_x is the receiver point.

If one considers only h^k one by one and considering that k = 0 is the line of sight LOS path that is calculated previously, the channel for $k > 0$ can be calculated recursively with equation 2.3.14.

$$h^{(k)}(t;S;R_x) = \int_S h^{(0)}\left(t;S,\left\{\vec{r},\hat{n},\frac{\pi}{2},d\vec{r}^2\right\}\right) \otimes h^{(k-1)}(t;\{\vec{r},\hat{n},1\},R)$$

Equation 2.3.14 rays path with multiple bounces

And if one calculates the convolution products obtain equation 2.3.15:

$$\boldsymbol{h^{(k)}(t;S;R_x) = \frac{n+1}{2\pi}\int_S \frac{\rho_r cos^n(\phi)(\psi)}{{d_{sj}}^2} \bullet rect\left(\frac{2\psi}{\pi}\right) h^{(k-1)}\left(t - \frac{d_{sj}}{c};\{\vec{r},\hat{n},1\},R\right) d\vec{r}^2}$$

Equation 2.3.15 rays path with multiple bounces

At this point to implement an algorithm one should transform the integral in a sum. To do this it is necessary to replace the integral with a summation, this process can be made with a simple consideration: the division of the wall area in many small finished tiles of area ΔA. Each of these tiles will be, as explained above, both a source and a receiver.

$$\begin{aligned} &{h_{nlos}}^{(k)}(t,S,R_x) = \\ &= \frac{m_l+1}{2\pi}\sum_{j=1}^{k} \rho_j cos^{\,m_l}(\phi_j)\frac{cos(\psi)}{d_{Sj}^2} rect\left(\frac{2\psi}{\pi}\right) * h_{los}^{(k-1)}\left(t-\frac{d_{Sj}}{c},S,R_x\right)\Delta A \end{aligned}$$

Equation 2.3.16 finite sum for one bounce.

From this one founds the ${h_{nlos}}^{(k)}$, to find the total path one should use the 5.4.1.1 and sum for all k in order to obtain the gain LOS + NLOS channel.

In [10] one can note that the execution time of this algorithm increments very quickly for k increment but in section 2.3.1 is showed that only three or four reflections are necessary to characterize perfectly the VLC NLOS channel.

2.3.5 Simulation Methods

The table below summarizes the main methods used to implement a VLC system. It is taken from [5].

Method	Modeling of Reflectance	Number of Reflections	Assumptions
Zemax approach	Wavelength Dependent	High Order (10 or more reflections are possible)	- Diffuse, specular and mixed reflections - Room with furniture or any other indoor environment - Realistic measured source
Monte Carlo Ray Tracing	Fixed Reflectance	Third Order	- Purely Lambertian reflections - Empty room - Ideal Lambertian source
Recursive	Fixed Reflectance	First Order	- Purely Lambertian reflections - Empty room - Ideal Lambertian source
Recursive	Fixed Reflectance	First Order	- Purely Lambertian reflections - Empty room - Ideal Lambertian source
Recursive	Averaged Reflectance	Fourth Order	- Purely Lambertian reflections - Room with furniture - Ideal Lambertian source
Recursive	Wavelength Dependent	Third Order	- Purely Lambertian reflections - Empty room - Ideal Lambertian source

Table 2.3.2 simulation methods from article [5]

This table is very interesting because it gives a panoramic view on existing simulation algorithms for VLC.

2.4 Modulation

Modulation is the set of transmission techniques aimed to transmit a signal called modulating signal by means of another signal called carrier signal.
For example with respect to figure 2.4.2 for VLC the carrier signal correspond to light wave (red) while the modulating signal correspond to On-Off square wave (blue).
The modulation schemes used in VLC are in general different from the ones used in radio communication modulation, because they use power to transmit information and not directly the wave amplitude. Power is in general proportional to the square of the amplitude.
Furthermore VLC is used for low cost systems and use the IM-DD technique that in general isn't used for radio because has only one degree of freedom and consequently low throughput.
VLC may be convenient with IM-DD techniques because the light frequency band is very large and unlicensed. Furthermore VLC is blocked by the walls and because of this in every room one can have a different VLC system without worrying about interference with other wireless networks.
The IM-DD is the most used contemporary technology but this does not mean that coherent transmission can't be used. In the following we discuss some of the most used modulation techniques for VLC systems: the on-off keying (OOK) modulation, the pulse position modulation (PPM) and some variants of that, the pulse amplitude modulation (PAM) and the color shift keying (CSK).

2.4.1 Modulation: important concepts and parameters

When one speaks about modulation a few important entities should be defined:

1. Baseband model
2. Symbol and Bit
3. Digital communication
4. Dimming
5. Spectral efficiency
6. Bit error ratio (BER) and error function
7. SNR (section 1.2.4)

Baseband model

The baseband model is a way to represent the signal. It is a mathematical way to simplify the transmission problem. The light can be represented like an electromagnetic wave with an electric and magnetic field, the electric field can be represented, neglecting the spatial component, with a wave according to the equation 2.4.1

$E(t) = A_0(t)cos\big(2\pi f_0 t + \phi_0(t)\big)$ Electric field plane wave

$E_a(t) = A_0(t)e^{i(2\pi f_0 t+\phi_0(t))} \Rightarrow E_a(t) = \big[\boldsymbol{A_0(t)e^{i(\phi_0(t))}}\big] * e^{i(2\pi f_0 t)}$

$E(t) = Re[E_a(t)] = Re\big[A_0(t)e^{i(2\pi f_0 t+\phi_0(t))}\big]$

$E_{env}(t) = \boldsymbol{A_0(t)e^{i(\phi_0(t))}} \equiv \boldsymbol{Baseband}$

$P_{env} = [|E_{env}(t)|]^2 = E_{env}(t)E_{env}{}^*(t) = A_0(t)e^{i(\phi_0(t))}A_0(t)e^{-i(\phi_0(t))} = |A_0(t)|^2$

Equation 2.4.1: Mathematical representation of envelope and baseband for electric field wave and power. Subscript a stands for analytical, env stands for envelope. A_0 is the wave amplitude, f_0 is the wave frequency, ϕ_0 is the wave phase, t is the time evolution, i is the imaginary unit.

The baseband model consist in considering only the envelope of the modulated wave at low frequency.

Symbol and Bit

The modulating wave state is called “symbol”, the symbol can vary according to modulation logic for example amplitude modulation puts into correspondence each amplitude state with a different symbol. Practically the symbol is the state of analogic modulating signal.

The bit instead is the state assumed by the digitized signal before entering in the modulator and after the demodulator.

Digital modulation

Digital modulations are “new” (respect to analogic) types of modulation in which the symbols are organized with binary logic, in this way the symbols can be represented with binary numbers and the mathematical manipulations are easy then the analogic counterpart. Digital modulations are in general less accurate then the analogic counterpart. The digital logic is based on the signal sampling that consist in the division of the analogical signal in many small pieces, every piece will be associated to its digital representation as illustrated in figure 2.4.1.

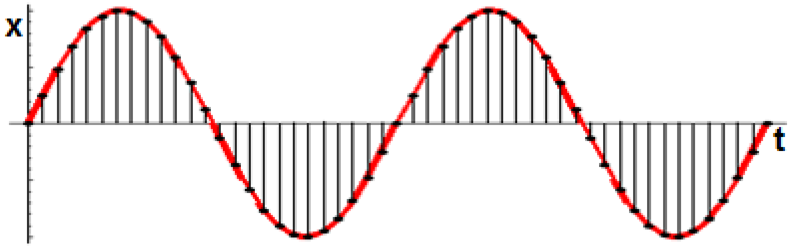


Figure 2.4.1 sampling example. The red line is the analogic signal, the dots are the digital sampling. Where t is the time and x represent a generic signal.

With digital logic it is possible to implement many types of operations that would be impossible for analogic counterpart, and solve some channel issues, like transmission error or bandwidth limitation. Essentially the digital logic is used to make the transmission more efficient and reliable.

The use of digital logic doesn't entail the transmission of a digital signal. In general the physical signal is an analog wave (or wave package). To use digital logic one need the opportunity to transform the analogic signal in digital signal and vice versa by sampling. An important theorem for sampling is the Nyquist-Shannon sampling theorem. Essentially this theorem states that the sampling frequency fs must be greater than twice the signal bandwidth, as represented in eq. 2.4.2:

$$f_s > 2f_M \qquad f_s = \frac{1}{\Delta t}$$

Equation 2.4.2 Nyquist-Shannon theorem. Here Δt is the sampling time: from figure 2.4.1 is the t distance between two dots, f_s is the sampling frequency and f_M is the frequency range occupied by the signal (bandwidth).

The theorem must be satisfied otherwise the reconstruction of the original signal, from the digitally sampled version, becomes impossible.

Dimming

VLC allows the communication through a beam of light generated from a light source, this should happen in various moment of the day. The user might want to change the light intensity for various reasons, this features can be satisfied with light dimming.

The dimming is a change in the average intensity of light to increase or to lower the ambient light.
Every modulation has its own way to realize the dimming, usually this is achieved by a variation of modulation duty cycle.

$$d = \frac{\tau}{T}$$ **d = duty cycle** **τ = one period active time** **T = period time** **equation 2.4.3**

Spectral efficiency

Spectral efficiency is a quality factor for modulation. This factor is nothing more than the ratio between the bitrate and the bandwidth. This is a dimensionless (bits are not a physical dimension) quantity that represents how the modulation occupies efficiently the bandwidth. The higher is this number more efficient is band occupation.
In the following this number will be calculated in some modulation scheme.
In general for VLC modulation the spectral efficiency is low with respect to the radio modulation but if one takes into consideration that the bandwidth for VLC can be approximately infinite (chapter 1) this isn't a problem.

Bit error ratio (BER) and error function

The bit error ratio is the ratio between the number of bits transmitted with the wrong value and the total number of transmitted bits. This is a very important dimensionless quantity because, to transmit with good efficiency, this number must be kept very low, otherwise the transmission will be compromised by the number of errors.
The bit error ratio is a very important parameter for communication systems. This parameter is difficult to calculate analytically, in particular with the complication of modulation. In chapter 3 section 2 and 3 this parameter is simulated for some type of modulations.
This thesis considers the error probability depending on an additive white Gaussian noise (WGN). This can be obtained considering the dominance of the background noise.
With a WGN dominants noise the BER is in general proportional to the Q function (Marcum), that represents the tail probability of the standard normal Gaussian distribution, or to erfc function (complementary error function), two very similar functions (with a simple substitution and a multiplication one can transform one in other) and in general are defined as in equation 2.4.4.

$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{\infty} exp\left(-\frac{t^2}{2}\right) dt \qquad erfc(x) = \frac{2}{\sqrt{\pi}} \int_x^{\infty} exp(-t^2) dt$

$Q(x) = \frac{1}{2} erfc\left(\frac{x}{\sqrt{2}}\right)$

Equation 2.4.4 Marcum Q function and complementary error function

2.4.2 OOK (On-Off Keying)

OOK is the simplest form of modulation suitable to transmit a VLC signal. Turning on and off the light is the simplest way to transmit information. OOK has carrier wave with two amplitude levels that represent the bits 0 and 1 of the modulating signal.
This modulation is simple to implement and is little affected by LED non linearity because has only two levels of amplitude. The flickering of OOK is in general low because the On-Off frequency of LED are very big (at least 200Khz). The only OOK drawback is low spectral efficiency $\frac{1bit}{s}/hz$ this because the OOK is only a one-dimensional modulation scheme. There are some different methods to support dimming in this modulation scheme, for example: redefining the "on" or "off" levels of the OOK to have a lower or higher luminous intensity

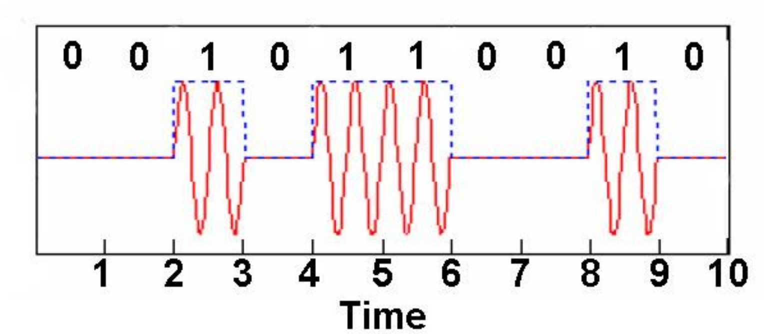


Figure 2.4.2: OOK modulation example from internet

OOK example: With 40 Mhz LED we transmit 40 millions of pulses in one second. Pulses are 0 or 1 therefore 1 bit every pulse, if we impose to use a band of 40Mhz we obtain example eq. 2.4.5:

$\Delta t = \frac{1}{f_{LED}} = \frac{1}{40\ Mhz} = 25ns \qquad f_{band-ook} = 40\ Mhz$

$$Sp_{eff} = \frac{1bit * 40Mhz}{40Mhz\ 1s} = \frac{1bit}{s * hz}$$

Equation 2.4.5 OOK spectral efficiency example

This tech is included in IEEE standard for VLC (802.15.7) described at the beginning of chapter 2.

The standard PHYI and standard PHYII include the OOK and PPM modulations.

Modulation	Optical clock rate	Data rate
OOK	200 kHz	11.67 kb/s
		24.44 kb/s
		48.89 kb/s
		73.3 kb/s
		100 kb/s

Table 2.4.1 OOK PHY I VLC standard IEEE 802.15.7

Modulation	Optical clock rate	Data rate
OOK	15 MHz	6 Mb/s
		9.6 Mb/s
	30 MHz	12 Mb/s
		19.2 Mb/s
	60 MHz	24 Mb/s
		38.4 Mb/s
	120 MHz	48 Mb/s
		76.8 Mb/s
		96 Mb/s

Table 2.4.2 OOK PHY II VLC standard IEEE 802.15.7

2.4.3 L-PPM (Pulse position modulation)

L-PPM (in short PPM) as OOK is one of the simplest and intuitive form of modulation to transmit a signal in VLC.
L-PPM, pulse position modulation is based on the position of the pulse inside the symbol. The symbol time T_s is divided in many time slots L. The position of the pulse is the value of the symbol.
The figure 2.4.3 is a good example to understand PPM.

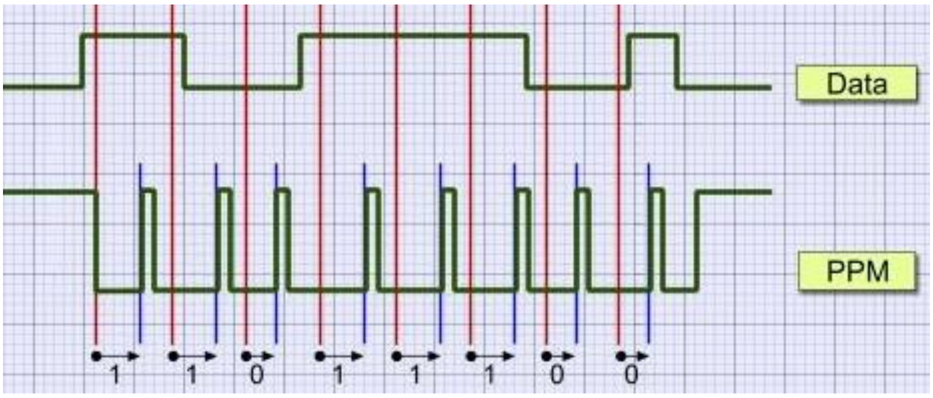


Figure 2.4.3: 2-PPM example

The L-PPM is very simple to implement. It has very little susceptibility to LED non linearity because it has only two amplitude levels. It has no flickering because every symbol is represented with the same power. In this way the average intensity of the modulation is the same for each symbol. Moreover the PPM dimming level can be adjusted simply by varying the duty cycle of the pulse, this variant is called variable pulse position (VPPM). Unfortunately this modulation scheme has two drawbacks: first, most important, is the low spectral efficiency due to the reason that it has only two amplitude levels to represents the symbols. Second it has big PAPR because it has only one time/slot pulse in symbol that lasts a minimum of two time slot or more. As one can see in chapter 2.1.1, the PAPR is important in VLC system because LED is not a linear device.
2.4.6 is the equation to calculate how many bits one can represent with L-PPM modulation.

$$\mathbf{M} = \log_2 L \Leftrightarrow \ L = 2^M$$

Equation 2.4.6 number or symbols to bit conversion equations. M is the significant bit number and L is the number of symbols.

To understand better the L-PPM the example 2.4.7 shows 4-PPM.

L= 4 $\quad$ $\mathbf{2 = \log_2 4}$

Equation: 2.4.7a 4-PPM example equation

T1	T2	T3	T4	Bit1	Bit2
1	0	0	0	0	0
0	1	0	0	0	1
0	0	1	0	1	0
0	0	0	1	1	1

Table 2.4.3 example of L= 4 PPM symbol.

Now, like in equation 2.4.5, we make an example of spectral efficiency calculation. We impose the LED blinking at 40Mhz, we consider that the occupied band for 4-PPM is two times the bandwidth of OOK. Therefore we can write the example 2.4.7b.

$$\Delta t = \frac{1}{f_{LED}} = \frac{1}{40\ Mhz} = 25ns$$

$$\log_2 4 = 2\ bits \qquad f_{LED} = 40Mhz$$

$$f_{4-PPM\ Band} = \frac{4}{2} f_{Band-ook}$$

$$Sp_{Eff\,4-PPM} = \frac{f_{LED}}{f_{4-PPM\ Band}} = \frac{2\ bits * 40Mhz}{4\ \ L - spaces * 40Mhz} = \frac{0.5\ bits}{s * hz}$$

Equations 2.4.7b 4-PPM spectral efficiency calculation

Equation 2.4.7a calculates directly how many significant bits one can represent with 4 symbols, Table 2.4.3 shows how the symbols are associated to bits and Equation 6.3.2.b shows the spectral efficiency for the 4-PPM.

Differently from the equation 2.4.5 this time we do not have one bit every symbol, instead we have 2 bits every 4 symbols consequently the spectral efficiency of 4-PPM is reduced by factor $\frac{1}{2}$. The example can be generalized with the formula 2.4.7c.

$$f_{Band\ L-PPM} = \frac{L}{M} f_{Band\ OOK}$$

$$Sp_{Eff\,L-PPM} = \frac{f_{LED}}{f_{Band}} = \frac{M * f_{LED}}{L * f_{Band-OOK}}$$

Equations 2.4.7c L-PPM spectral efficiency calculation

This is a big issue of PPM modulation. This seems to make the PPM worse than the OOK, but in chapter 3 section 2 we will see that PPM has a smaller BER than the OOK.

VPPM (Variable Pulse position modulation)

Variable pulse position (VPPM) is a dimming variant of PPM. The idea is to change the time duration of the pulse according to the incoming signal intensity in order to deal with dimming: changing the time duration of the pulse means changing the duty cycle of the signal and consequently its average power and brightness. In this way the user can change the illumination level of the room.

This tech presents some problem of flickering. Changing the energy of the symbol can bring to large variations of the signal's intensity.

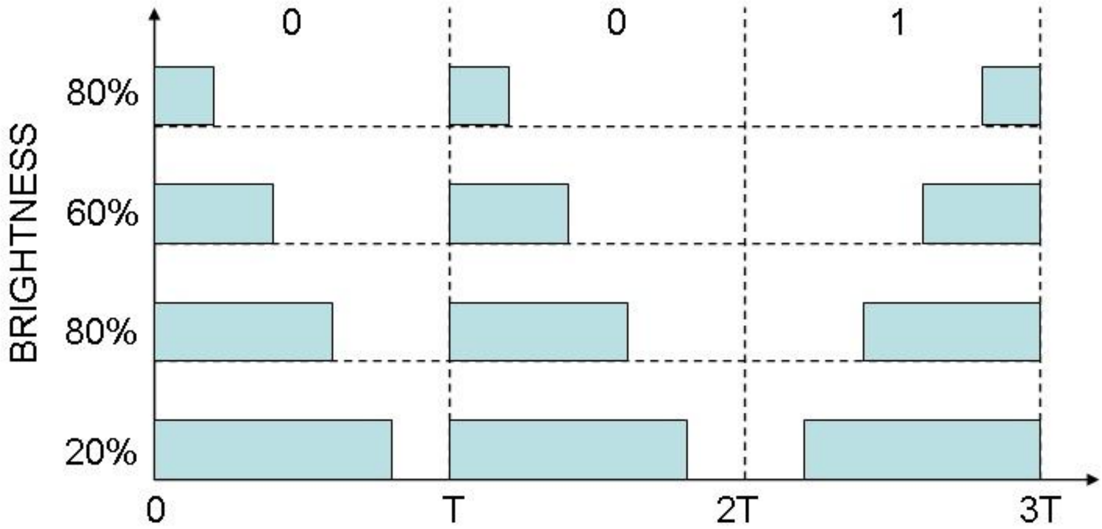


Figure 2.4.4 Dimming example with VPPM

This tech is included in IEEE standard for VLC (802.15.7).

The standard PHY I and standard PHY II include the OOK and VPPM modulations:

Modulation	Optical clock rate	Data rate
VPPM	400 kHz	35.56 kb/s
		71.11 kb/s
		124.4 kb/s
		266.6 kb/s

Table 2.4.4 VPPM PHY I VLC standard IEEE 802.15.7

Modulation	Optical clock rate	Data rate
VPPM	3.75 MHz	1.25 Mb/s
		2 Mb/s
	7.5 MHz	2.5 Mb/s
		4 Mb/s
		5 Mb/s

Table 2.4.5 VPPM PHY II VLC standard IEEE 802.15.7

2.4.4 CSK (Color shift Keying)

This modulation scheme is similar to FSK for radio transmission because changing frequencies means changing the wavelength. The base band signal is modulated with carrier waves of many wavelenghts. The color in the name of this modulation is referred to the wavelengths of the carrier: changing the wavelength means changing the color of the carriers.

To transmit through the colors there are two types of colored LED as one can see in chapter 3: slower blue LED with yellow phosphor, and faster three-color LEDs. These two LEDs type can be used to implement this modulation technique.

This modulation has good resistance to flickering because the average power transmitted is constant. For CSK the led non linearity is not a problem because the modulation has only two levels of intensity.
The issues are with dimming control techniques. There are two techniques for dimming control with CSK:

1) One can vary the input LED's current ->this method doesn't change the bit rate but can generate color shifting.
2) One can add a compensation signal -> this method doesn't shift the color but lowers the bit rate and also could generate synchronization problems.

This modulation scheme is included in the IEEE standard for VLC (802.15.7).
The standard PHY III include the CSK modulation.

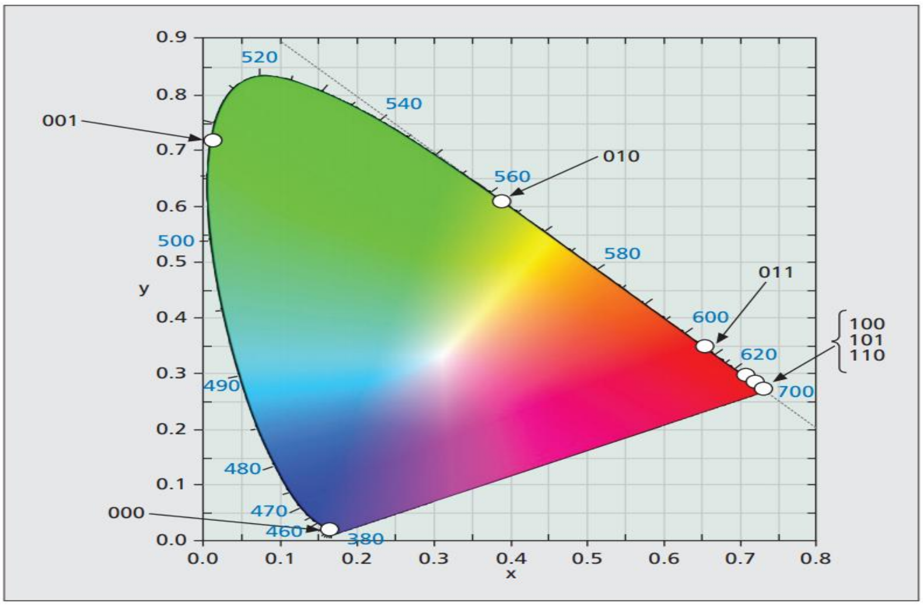


Figure 2.4.5: IEEE 802.15.7 CSK colors standard on CIE 1931 xy colors space.

Frequency band	Spectral width (nm)	Color	Proposed Code
380-450	70	pB	000
450-510	60	B,BG	001
510-560	50	G	010
560-600	40	yG,gY,Y,yO,O	011
600-650	50	rO	100
650-710	60	R	101
710-780	70	R	110
		Reserved	111

Table 2.4.6 VLC standard table. Provides support for 7 bands in the visible light spectrum.

Modulation	Optical clock rate	Data rate
4-CSK	12 MHz	12 Mb/s
8-CSK		18 Mb/s
4-CSK	24 MHz	24 Mb/s
8-CSK		36 Mb/s
16-CSK		48 Mb/s
8-CSK		72 Mb/s
16-CSK		96 Mb/s

Table 2.4.7: IEEE 802.15.7 CSK standard for VLC

2.4.5 N-PAM (Pulse amplitude modulation).

N-PAM is a variant of On-Off keying modulation but with more then two amplitude levels. With more than two amplitude levels the spectral efficiency increases significantly, as calculated in equation 2.4.8.

Ex.

Remembering section 2.4.2:

$$Sp_{eff_{OOK}} = 1\left(\frac{bit}{s}\right)/hz$$

While the N-PAM:

$$Symbols = N = 2^b$$

$$Sp_{eff_{N-PAM}} = (\log_2 N)\left(\frac{bit}{s}\right)/hz$$

Equation: 2.4.8 example where N is the number of symbol for PAM and b is the number of significant bit for N symbol.

In PAM one has a carrier signal that changes its amplitude (for VLC is the intensity) to represent the digital modulating signal. There are several types of PAM based on how many symbols one wants to represent: 2 PAM for 2 symbols, 4 PAM for 4 symbols, etc….

This modulation scheme has high spectral efficiency based on how many symbols one can represent: the bigger N the higher the spectral efficiency.
The drawbacks are flickering, dimming, shadowing and susceptibility to non linearity. All these problems increase when the N of PAM raise because if N grow the signal could exit from led linear zone and the PAPR.

3 Simulation

In order to study the performance of VLC systems, for which the theory was described in the previous chapters, we performed some simulations.
In general a simulation is a simplified model of reality in which the focus is on what are believed to be the dominant phenomena and second-order effects can be neglected.
This model usually is translated into a computer algorithm that is able with a finite number of steps to calculate the parameters that one wants to evaluate.
For our simulations we decided to use the Matlab computer software because it is a very popular software with a toolbox pack (like BER function) suited to communication systems. Furthermore many VLC research entities use Matlab algorithms.
In the following we present three simulations:

1. LOS channel simulation
2. OOK SNR(BER) simulation
3. PPM SNR(BER) simulation

The first simulation concerns the physical channel meanwhile the second and third deal with two of the modulation schemes that can be used in a VLC system.

3.1 LOS channel simulation

The theory of LOS channel was already explained in chapter 2.3 section 3.
The Matlab algorithm that calculates the LOS channel gain is in APPENDIX A.1.
The algorithm takes the LED luminous power and geometry and uses eq. 2.3.9 and 2.3.10 to propagate it to every point in the room using the LOS channel model and calculate the luminous power at the receiver.
The algorithm considers:

- Empty room with x, y, z dimension
- LED located on the ceiling at fixed position (lx,ly,lz), with fixed direction of emission perpendicular to the floor and fixed half-power angle (halfpower).
- Photo-receiver parallel to floor at rz distance from ceiling with responsivity “resp” and with receiving surface “ar”.

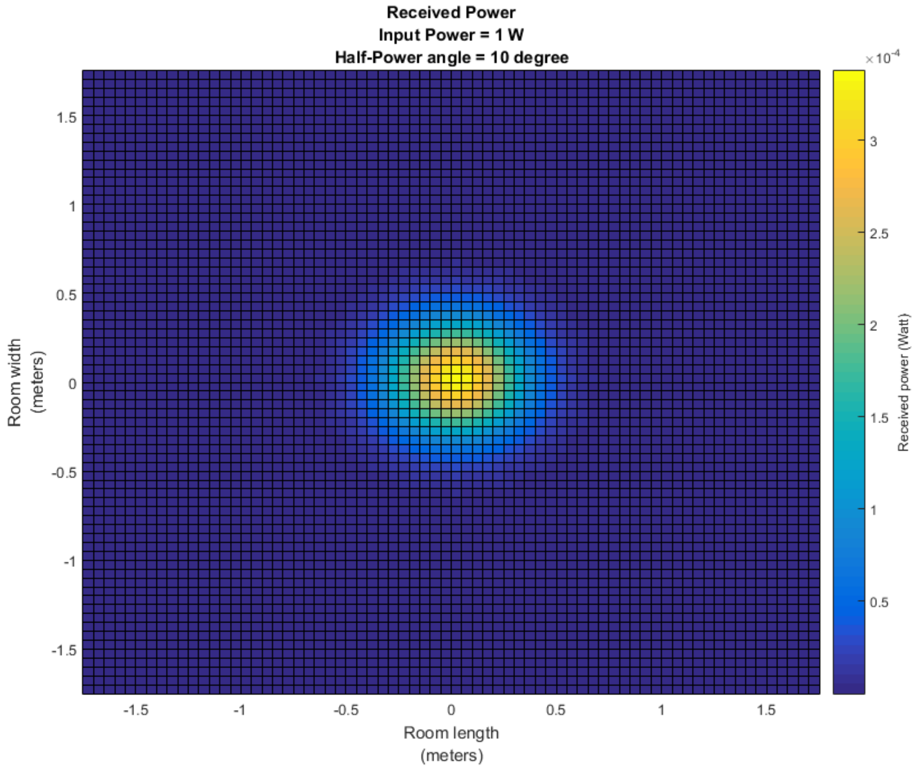
- Optical filter index “n” on the photoreceiver (if n is set to 1 the optical filter is negligible).

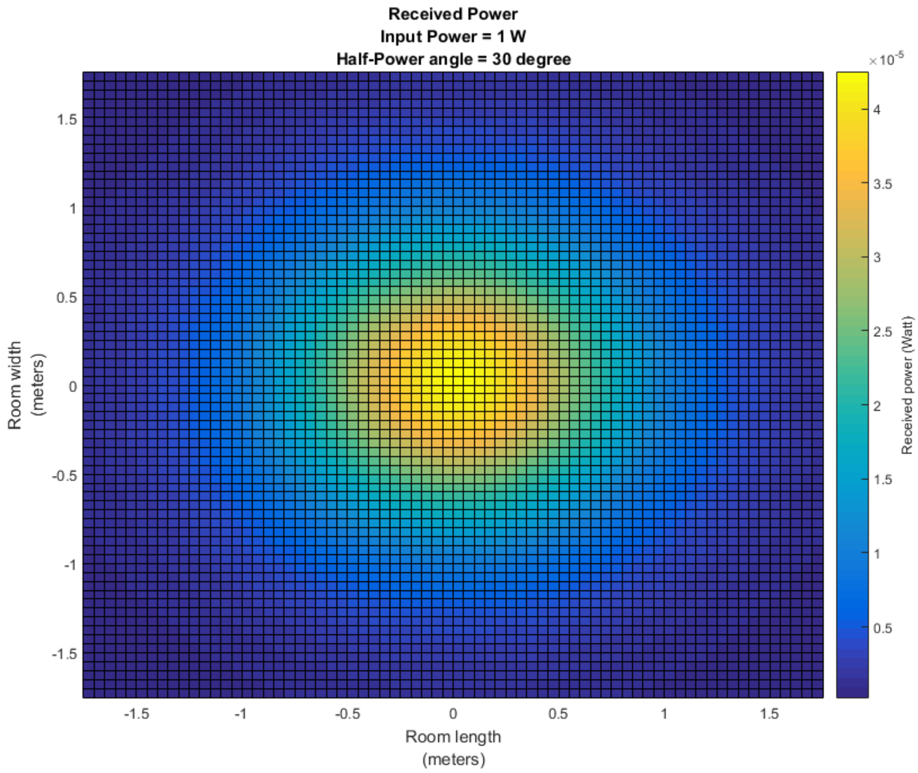
From this algorithm one calculates the gain for every position of the photodiode on x,y,fixed rz plane.
The code with simple modifications can be used to analyze various situations.
We simulate the resulting received power with the following condition:

- Empty room dimension (x=3.5m;y=3.5m;z=2.5m)
- 1 LED
  - fixed position in ceiling center
  - **1Watt** luminous power
  - Led half-power angle (30°)
- 1 Photoreceiver
  - Floor distance (rz = 1m)
  - with neglected responsivity (= 1 $\frac{A}{W}$)
  - receiver area (=1cm^2)

We vary the half-power angle: the three plots we obtained are listed after. In order from top to bottom:

- 10 degrees
- 30 degrees
- 75 degrees


Received Power
Input Power = 1 W
Half-Power angle = 10 degree
Room width
(meters)
Room length
(meters)
Received power (Watt)
×10^-4


Received Power
Input Power = 1 W
Half-Power angle = 30 degree
Room width
(meters)
Room length
(meters)
Received power (Watt)
×10^-5

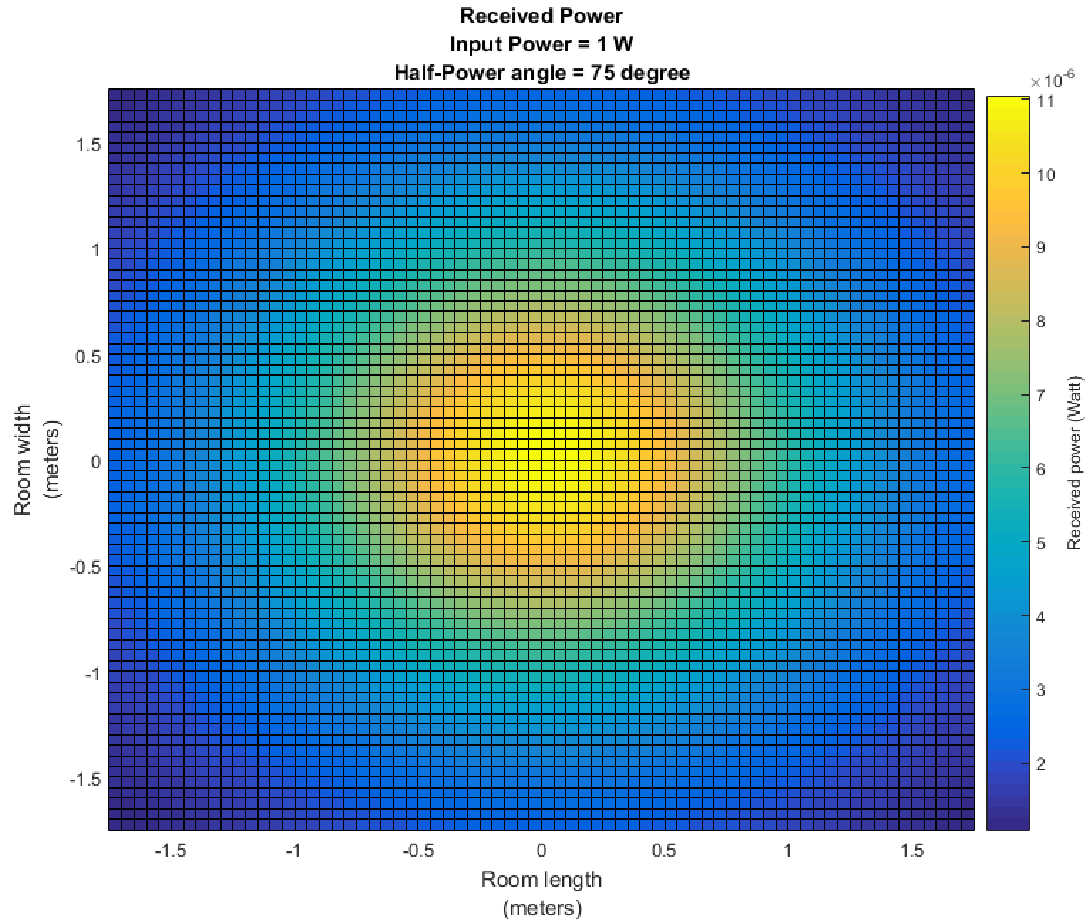


Figure 3.1.1: Received luminous power. From top to bottom: 10, 30, 75 degrees half-power angle . The color indicates the received power. In the center the full yellow color is the biggest power received, on sides the dark blue indicates the lowest power received. The numbers on three colors bars are different. . Figure made with Matlab code appendix A.1

In figure 3.1.1 the received power distribution in the various points of the room is shown. It depends strictly on LED half-power angle: with small half-power-angle the received power is very concentrate consequently the signal would be directional. If one uses a LED with big half-power angle instead will have all the transmitted power diffused in the room.

For basic experimental purposes, for example in our chapter 8, one should prefer to use a LED with small half-power angle because the power is very concentrated and in this way one can use only few LED in a small space.

As one can see in figure 3.1.1 the energy received is in general very low. This problem could be solved through the use of the LED array/Matrix. This issue should be considered in a future experimental design phase.

3.2 Modulation simulation

We want to simulate with MATLAB software the BER (Bit Error Ratio) curve as a function of the SNR for the OOK and PPM modulations in a VLC system.

The BER can be calculated by comparing the original signal upstream of the transmitter, with the received signal after the receiver. This will not be equal to transmitted signal because it will be immersed in noise. The BER is the function that tells how many errors there are as a function of the SNR used in the transmission.

In telecommunication, BER is very important because is a good parameter to estimate the quality of the communication system. Usually this parameter can be estimated with simulation or with theoretical analytical calculations. In general, the theoretical formula is more convenient to use but the calculations to obtain it are very complicated. The simulations are made with a stochastic method, usually they work generating a random signal (the transmission signal) to which is added a random noise due to receiver (the stochastic parameter) in this way the received signal has a random component that which can cause a reception error. The simulation method may require large computation resources.

The OOK and L-PPM are simple modulation therefore in our case we were able to estimate and calculate both the theoretical and simulated curves for both modulations. In the following section (3.2.1 and 3.2.2) we estimate the BER for OOK and L-PPM in both theoretical and simulation ways. Before one can simulate a system, one should define the model used. To do this we have to make some considerations about the system properties. In our case, the following assumptions are made:

1) The channel considered is LOS
2) Neglect path loss and multipath dispersion
3) Neglect interference due to artificial light
4) Neglect thermal noise
5) Dominant noise is background light
6) Dominant noise considered white gaussian
7) Neglect the band limitation imposed by the transmitter and receiver
8) Neglect the transmitter receiver synchronization

1, 2 and 3 are channel considerations, with these assumption one can neglect the channel that can be considered a simple wire.
4, 5, 6 are noise considerations, with these assumption one can only consider the background noise as white Gaussian. This is no loss of generality for indoor system, for example if the room considered has the background noise added by a window and that noise is much larger with respect to the thermal, dark current and quantum noise these last three noises can be neglected respect to the first.
7 and 8 can be considered later, when one needs a very well detailed real prototype.

3.2.1 OOK simulation

OOK (On-Off Keying) is one of the most used modulation in OWC and VLC. This modulation associates the bit 1 to on signal and the bit 0 to off signal (see chapter 2 section 4).
The modulation block consist in taking the bit value and multiply that by the led luminous power. On the reception block the photodiode generates a photocurrent proportional to the transmitted luminous power plus white Gaussian noise n(t), both proportional to R the responsivity of photodiode.

Pavg, Ipeak and Epeak for OOK consideration:

The OOK modulation is illustrated in Figure 3.2.1.1.

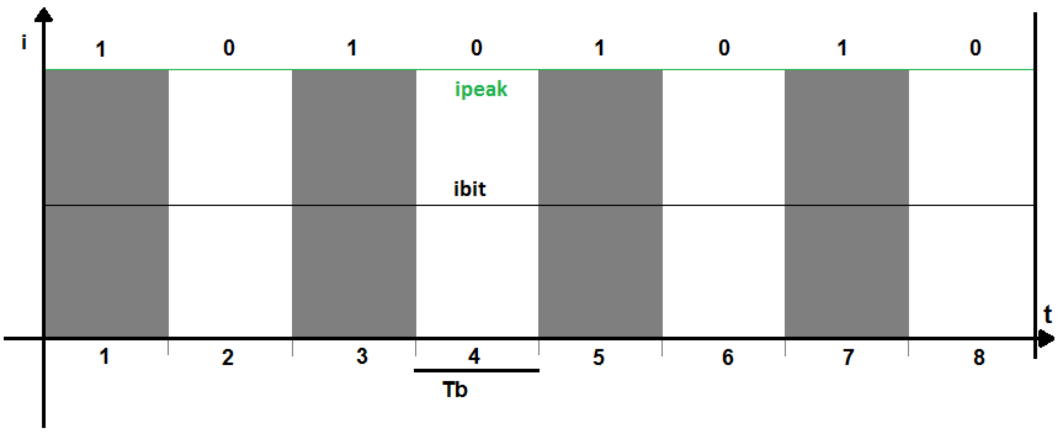


Figure 3.2.1.1: OOK received wave envelope. ipeak is the generated electrical current after the receiver (y axis), t is the time (x axis), ipeak is the peak wave current energy, ibit is the bit current.

The electrical power and the peak and bit energy for one period of 3.2.1 wave is calculated in equation 3.2.1.1:

$$\boldsymbol{E_{peak} = I_{peak}^2 T_b = 2E_b}$$

Equation 3.2.1.1 Peak power and energy peak and bit for one period.

The electric peak current is proportional (chapter 2 section 2) to the luminous power received and the luminous power received is proportional to the square root of the electric wave power (equation 3.2.1.2):

$$I_{peak} = 2RP_{avg-ook} = 2R\sqrt{\frac{R_b E_b}{2R^2}} = \sqrt{\frac{2E_b}{T_b}}$$

$$P_{avg-ook} = \sqrt{\frac{N_0 R_b SNR}{2R^2}} = \sqrt{\frac{R_b E_b}{2R^2}} = \sqrt{\frac{R_b E_b}{2R^2}}$$

Equation 3.2.1.2: Received luminous power and peak current generated from photoreceiver.

The block scheme for OOK is shown in figure 3.2.1.2:

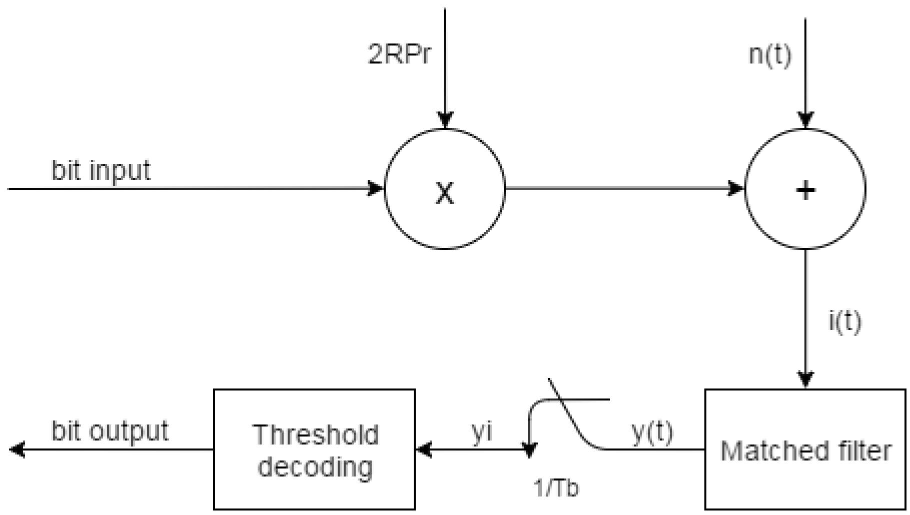


Figure: 3.2.1.2 OOK block scheme.

The ideal device used for reception of OOK signal with WGN noise is matched filter followed by threshold detector, the matched filter performs a convolution integration to maximize the SNR [13]. The signal before the matched filter should be a square wave like in equation 3.2.1.3.

$$i(t) = \begin{cases} I_{peak} + n(t) \\ n(t) \end{cases} \qquad I_{peak} = 2P_{avg}R \qquad P_{avg-ook} = \sqrt{\frac{R_b E_b}{2R^2}}$$

Equation 3.2.1.3 received signal (before matched filter). I_{peak} **is the peak wave current, n(t) is the noise, Pavg is the average received luminous power, R is photodiode responsivity.**

After a matched filter a triangular wave is generated:

$$r(t) = \begin{cases} E_{peak} + n(0,\sigma) \\ n(0,\sigma) \end{cases} \qquad E_{peak} = I_{peak}^2 T_b$$

Equation 3.2.1.4 Received signal after matched filter. Epeak is the energy peak of the triangular wave, n(0,σ) is the noise with average μ=0 and standard deviation σ = σ.

The WGN noise calculated after matched filter:

$$\sigma^2 = \frac{N_0}{2}\int_0^{T_b} r^2(t)\,dt = \frac{N_0}{2}E_p \rightarrow \sigma = \sqrt{\frac{E_p N_0}{2}}$$

Equation 3.2.1.5 Noise after matched filter. σ^2 **is the noise variance, Tb is the bit time, N0 is the spectral noise, Ep is the peak energy and σ is the noise standard deviation.**

Where factor 2 is given by N_0 spectrum: the spectral noise N_0 is calculated on all the frequency spectrum (from -∞ to +∞) while the noise used is only on positive time.
After that the simulation takes the output energy signal from matched filter and compares it with a threshold detector where the threshold taken is at half energy because symbols are equiprobable.

$$E_{threshold} = \frac{E_p}{2} \qquad \begin{cases} E_{received} \leq E_{threshold} \rightarrow 0 \\ E_{received} > E_{threshold} \rightarrow 1 \end{cases}$$

Equation 3.2.1.6 OOK threshold

After threshold decoding one should have the original random generated signal and the received decoded signal, to calculate the BER one can evaluate the different bits of two signals and divide the total by the bit number. Our simulation uses directly the Matlab biterr function that receives in input the two signals and returns the BER.
The BER is calculated for many SNR values making a cycle that every time increases the SNRdB by one and consequently the Energy and noise. If the SNR increases too much, the computational resources may become insufficient, because the error becomes really small and one should have in input a very large signal to detect few errors.

Theoretical OOK error probability:

To evaluate the simulated BER curve one can compare the simulated BER curve with a theoretical BER curve.

The OOK plus WGN noise bit error probability is given by 3.2.1.7:

$$P_{error} = p(0)\int_{i_{th}}^{+\infty} p(i/0)\, di + \ p(1)\int_{0}^{i_{th}} p(i/0)\, di$$

Equation 3.2.1.7 OOK bit error probability.

The p(0) and p(1) represent the probability of obtaining respectively 0 or 1, p(i/0) and p(i/1) are the tail density probability of obtaining 0 instead 1 and vice versa, ith represent the threshold level. The integral gives the distribution function. If one consider (like we say in previous chapter) white Gaussian noise the tail density probability for Gaussian noise are equation 3.2.1.8.

$$p(i/0) = \frac{1}{\sqrt{2\pi\sigma^2}} exp\left(\frac{-i^2}{2\sigma^2}\right) \qquad p(i/0) = \frac{1}{\sqrt{2\pi\sigma^2}} exp\left(\frac{-\left(i-I_p\right)^2}{2\sigma^2}\right) \qquad I_p = 2I_{th}$$

Equation 3.2.1.8 Gaussian tails density probability. i is the current variable, $\boldsymbol{\sigma^2}$ is the noise variance, Ip is current peak and Ith is the threshold current.

If symbols are equiprobable the probability of detecting one or zero are the same $p(0) = p(1) = \frac{1}{2}$ and the optimum threshold point is $I_{th} = 0.5I_p$.

With some mathematical passage one can obtain 3.2.1.9:

$$P_{error} = \frac{1}{2}\int_{i_{th}}^{+\infty} \frac{1}{\sqrt{2\pi\sigma^2}} exp\left(\frac{-i^2}{2\sigma^2}\right) di + \frac{1}{2}\int_{0}^{i_{th}} \frac{1}{\sqrt{2\pi\sigma^2}} exp\left(\frac{-\left(i-I_p\right)^2}{2\sigma^2}\right)\, di =$$

$$= \frac{1}{2\sqrt{2\pi\sigma^2}}\left(\int_{i_{th}}^{+\infty} exp\left(\frac{-i^2}{2\sigma^2}\right) di + \int_{0}^{i_{th}} \frac{1}{\sqrt{2\pi\sigma^2}} exp\left(\frac{-\left(i-I_p\right)^2}{2\sigma^2}\right)\, di\right) =$$

$$= \frac{1}{2\sqrt{2\pi\sigma^2}}\left(2 * \int_{i_{th}}^{+\infty} exp\left(\frac{-i^2}{2\sigma^2}\right) di\right) = \frac{1}{\sqrt{2\pi\sigma^2}}\left(\int_{i_{th}}^{+\infty} exp\left(\frac{-i^2}{2\sigma^2}\right) di\right)$$

replacing: $\begin{cases} \frac{i^2}{\sigma^2} = t^2 \\ i = t\sigma \\ di = \sigma dt \end{cases} \qquad \begin{cases} t(i_{th}) = \frac{i_{th}}{\sigma} \\ t(i_{th} = \infty) = \infty \end{cases}$

$$\frac{1}{\sqrt{2\pi\sigma^2}}\left(\int_{i_{th}}^{+\infty} exp\left(\frac{-i^2}{2\sigma^2}\right)di\right) = \frac{\sigma}{\sqrt{2\pi\sigma^2}}\left(\int_{\frac{i_{th}}{\sigma}}^{+\infty} exp\left(\frac{-t^2}{2}\right)dt\right) = \frac{1}{\sqrt{2\pi}}\left(\int_{\frac{i_{th}}{\sigma}}^{+\infty} exp\left(\frac{-t^2}{2}\right)dt\right) =$$

$$= Qfunc\left(\frac{i_{th}}{\sigma}\right)$$

$$\boldsymbol{P_{error} = Qfunc\left(\frac{i_{th}}{\sigma}\right) \qquad Qfunc(x) = \frac{1}{\sqrt{2\pi}}\int_{x}^{+\infty} e^{\frac{-\alpha^2}{2}}\,d\alpha}$$

Equation 3.2.1.9 Error probability. Qfunc is Marcum's Q-function on right.

The 3.2.1.9 probability is calculated before the matched filter, after matched filter $I_{peak} \rightarrow E_{peak}$ and with substation of 3.2.1.3 one obtain 3.2.1.10:

$$P_{error} = Qfunc\left(\frac{i_{th}}{\sigma}\right) = Qfunc\left(\frac{i_{peak}}{2\sqrt{\frac{E_{peak}N_0}{2}}}\right) = Qfunc\left(\frac{E_{peak}}{2\sqrt{\frac{E_{peak}N_0}{2}}}\right) = Qfunc\left(\sqrt{\frac{E_{peak}}{2N_0}}\right) =$$

$$= Qfunc\left(\frac{E_b}{\sqrt{2E_bN_0}}\right)$$

$$\boldsymbol{P_{error-OOK} = Qfunc\left(\sqrt{\frac{E_b}{N_0}}\right)}$$

Equation 3.2.1.10 Theoretical OOK error probability.

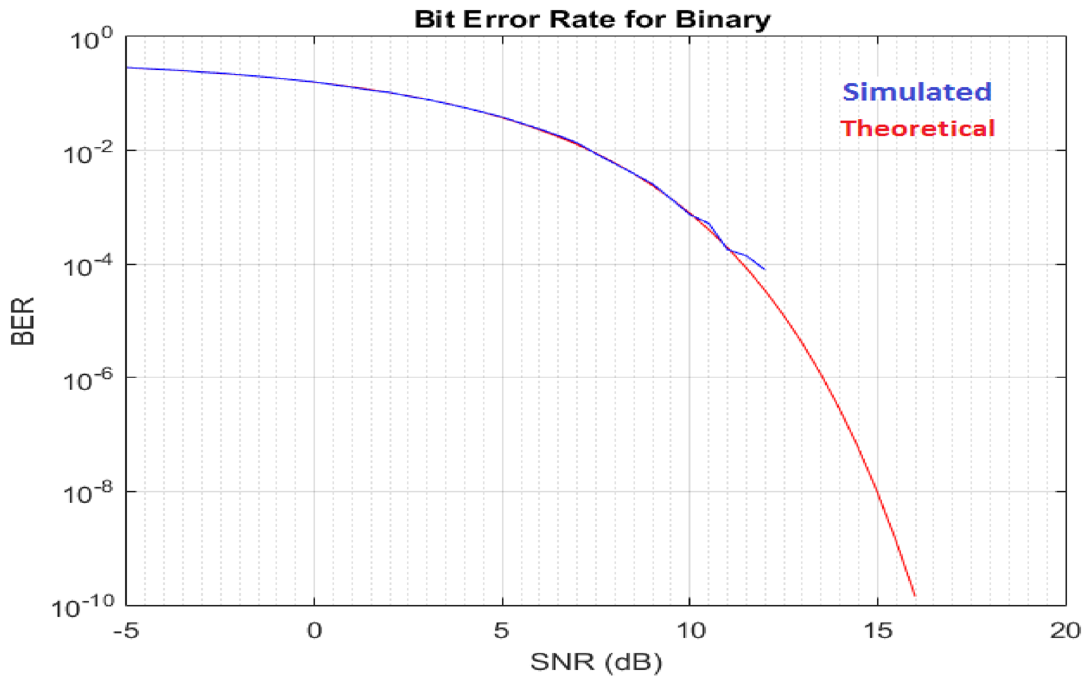


Figure 3.2.1.3: OOK BER simulation with random generated signal of 50000 bit. . Figure made with Matlab code appendix A.2

Figure 3.2.1.3 shows the simulated (from A.2 algorithm) and the theoretical (equation 3.2.1.10) BER. Looking the figure we can say that the theoretical BER can well approximate the simulated BER for an OOK. From this moment we approximate the OOK BER with theoretical function equation 3.2.1.10.

3.2.2 L-PPM simulation

In L-PPM (pulse position modulation) the bit is encoded on symbol that is dived in L time slots, for each symbol one pulse is transmitted that changes its time position according to the bit that it should represent. L-PPM has in general (except for the 2-PPM) better BER(SNR) performance respect OOK but at the cost of increased bandwidth occupation and greater system complexity.

The block scheme of an L-PPM modulation is illustrate in figure 3.2.2.1.

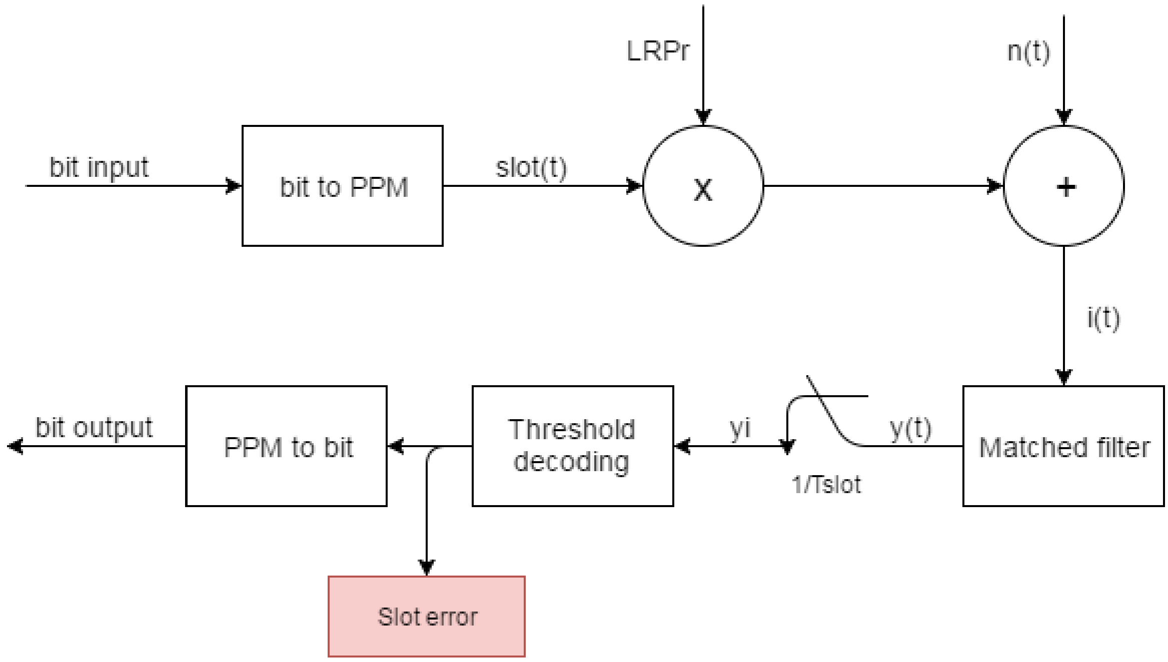


Figure 3.2.2.1 L-PPM block scheme

The red block only indicates where the error is measured.

After this intro, some examples of L-PPM encoding will be presented and finally the L-PPM BER will be theoretically derived and compared with simulation results.

A first example of 4-PPM was shown in chapter 2 section 4.3.

With a simple table 3.2.2.1 and figures 3.2.2.2 one can compare the OOK with 2-PPM.

$E_{peak} = = 1$	Ts1/Tb1	Ts2/Tb1	Ts3/Tb2	Ts4/Tb2	Ts5/Tb3	Ts6/Tb3	Ts7/Tb4	Ts8/Tb4	Eavg-Sym
OOK	1	1	0	0	1	1	0	0	0.5
2PPM	1	0	1	0	1	0	1	0	0.5

Table 3.2.2.1 representation of a segment of 2-PPM vs OOK.

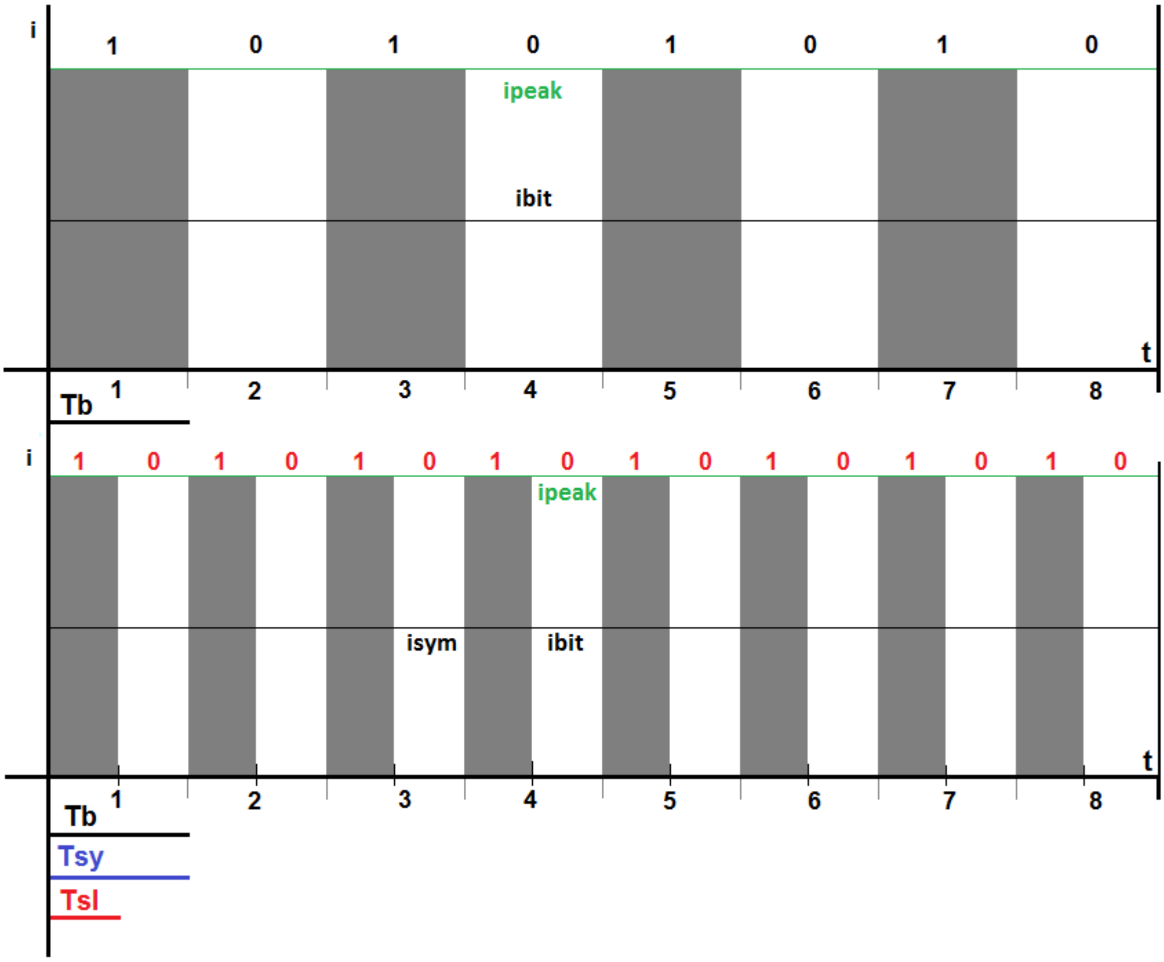


Figure 3.2.2.2: OOK vs 2-PPM comparison. ipeak is the peak current, isym is the symbol current and ibit is the bits current. Tsy is symbol time, Tsl is slot time, Tb is bit time. *The signals presented in picture don't represent the same encoded message, the picture is used only to current and energetic considerations.

As one can see in figure 3.2.2.2 these two modulations have the same Eb, Es, Ep, Tb and Tsy while the Tsl is Tb/2.

With other simple graphical example one can pass from 2-PPM energy dependence to 4-PPM energy dependence.

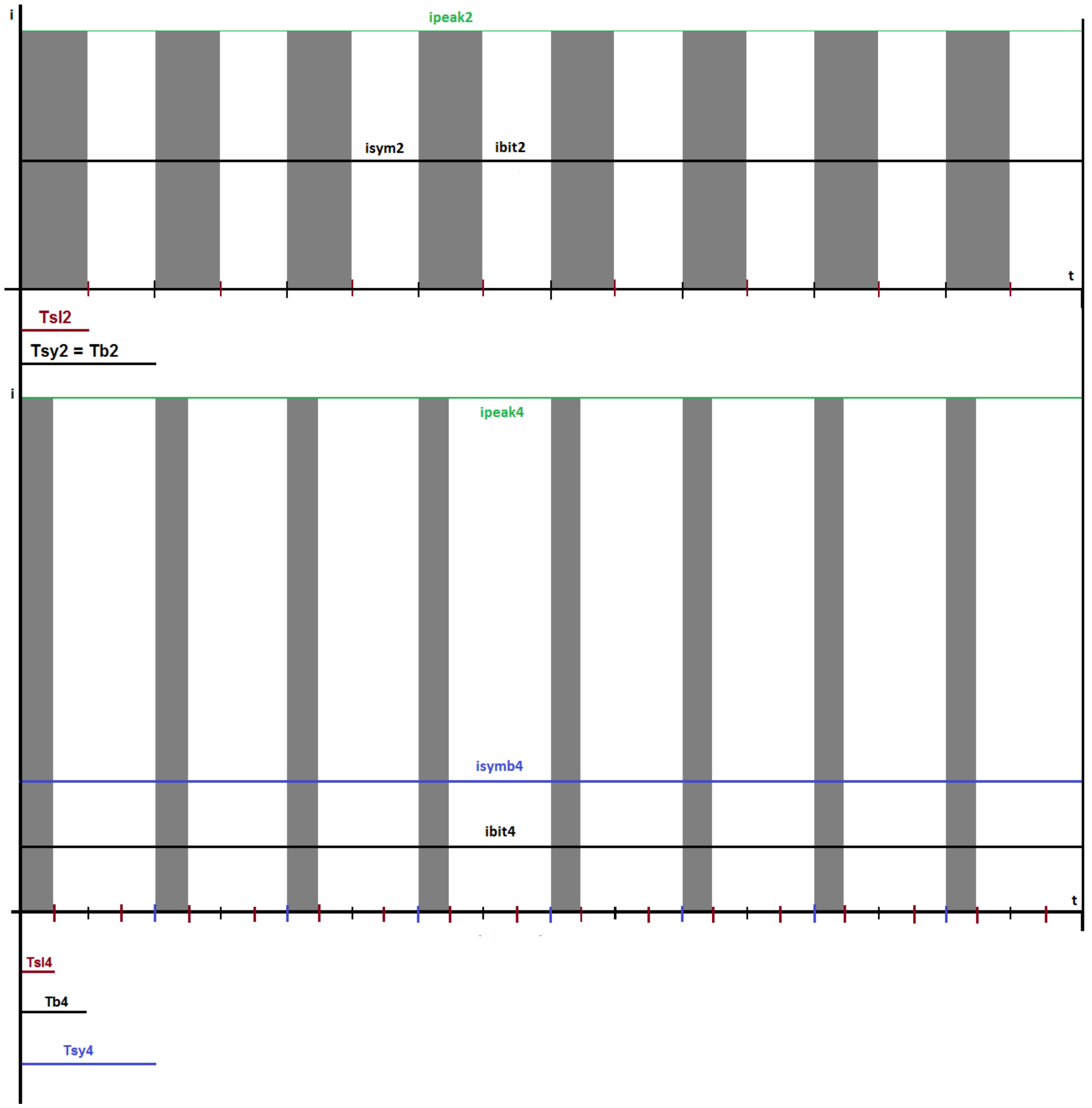


Figure 3.2.2.3 Example of 2-PPM (top) vs 4-PPM (bottom). ipeak is the peak current, isymb is the f symbol and Ibit is the bit current. Tsy is symbol time, Tsl is slot time, Tb is bit time. The number after the value name is the L of the PPM, for example 2-PPM has peak energy etc. *The signals presented in picture don't represent the same encoded message, the picture is used only to current and energetic considerations.

The figure 3.2.2.2 and 3.2.2.3 are comparison of OOK, 2-PPM and 4-PPM. All these figures are made in order to have the same energy per symbol for every modulation.

After these examples is easily to understand which are the times values and how they change in function of L. The time values are: the bit time Tb, the symbol time Tsy and the slot time Tslot. The first is the time duration for the modulation to transmit one significant bit, the second is the time to transmit one symbol of the modulation and the third is the time to transmit one pulse of the symbol.

With the increase of L there is another quantity that change: the energy. Three quantities related to energy are important: the peak energy, the bit energy and the symbol energy. Symbol and bit energy are referred to the average energy to transmit one symbol or one bit, while the peak energy is referred to the most energetic slot in one symbol. The important feature is that to represent one symbol of L-PPM one should use the peak energy proportional to L in order to have the same energy symbols for all the L-PPM and the previous chapter OOK (for OOK symbol energy is equal to bit energy). The equations set 3.2.2.1 define the values listed above.

$$\begin{cases} T_{slot} = T_{bit}\frac{M}{L} \\ T_{sym} = T_b M \\ \frac{T_{sym}}{L} = T_{slot} \end{cases} \quad \begin{cases} \frac{E_{peak}}{L} = E_{sym} \\ E_b \log_2 L = E_{sym} \end{cases} \quad \begin{cases} E_{peak} = LE_{sym} = LE_b \log_2 L = E_b LM \\ \log_2 L = M \end{cases}$$

Equation 3.2.2.1: Equation set that define the L-PPM modulation. Epeak, Es and Eb are respectively: peak energy, symbol energy and bit energy. Tslot, Tsym and Tb are respectively: one slot duration, one symbol duration and one bit duration. L is the number of slot used to represent one symbol and M is the significant bit for an L-PPM scheme.

The current signal before the matched filter is written in 3.2.2.2:

$$i(t) = \begin{cases} I_{peak} + n(t) \\ n(t) \end{cases} \quad \boldsymbol{I_{peak} = LP_{avg}R}$$

Equation 3.2.2.2 received signal (before matched filter). $\boldsymbol{I_{peak}}$ is the peak wave current, n(t) is the noise, Pavg is the average received luminous power, R is photodiode responsivity.

To obtain the previous Ipeak proportionality, one should define the Pavg for L-PPM:

$$P_{avg-LPPM} = \frac{\mathbf{1}}{\boldsymbol{L}}\sqrt{\mathbf{2M}\frac{\boldsymbol{R_{sym}N_0SNR}}{\boldsymbol{2R^2}}} = \frac{1}{L}\sqrt{\mathrm{M}\frac{R_{sym}E_b}{R^2}} = \frac{1}{L}\sqrt{\frac{R_\mathrm{b}E_b}{R^2}}$$

Equation 3.2.2.3: L-PPM Pavg definiton

With that I_{peak} and $P_{avg-LPPM}$ the E_{peak} can be obtained from some algebra:

$$r(t) = \begin{cases} E_{peak} + n(0,\sigma) \\ n(0,\sigma) \end{cases} \qquad E_{peak} = I_{peak}^2 T_{sym} = \boldsymbol{LE_{sym}} = \boldsymbol{MLE_b}$$

$$E_{peak} = I_{peak}^2 T_{slot} = \left(LRP_{avg}\right)^2 T_{slot} = \frac{R_\mathrm{b}E_b L^2 R^2}{R^2} T_{slot} = T_b \frac{L^2}{T_b}\frac{M}{L}E_b =$$

$$= \boldsymbol{LME_b = LE_s = E_{peak}}$$

Equation 3.2.2.4 Received signal after matched filter. Epeak is the energy peak of the triangular wave, n(0,σ) is the noise with average μ=0 and standard deviation σ = σ.

And the WGN noise calculated after matched filter in equation 3.2.2.5:

$$\sigma^2 = \frac{N_0}{2}\int_0^{T_b} r^2(t)\,dt = \frac{N_0}{2}E_p \rightarrow \sigma = \sqrt{\frac{E_{peak}N_0}{2}} = \sqrt{\frac{LE_{sym}N_0}{2}} = \sqrt{\frac{LME_{bit}N_0}{2}}$$

Equation 3.2.2.5 Noise after matched filter. $\boldsymbol{\sigma^2}$ is the noise variance, Tb is the bit time, N0 is the spectral noise, Ep is the peak energy and σ is the noise standard deviation.

Where the factor 2 is given by N_0: the spectral noise N_0 is calculated on all the frequency spectrum (from -∞ to +∞) while the noise used is only on positive time.
After that the simulation takes the output energy signal from matched filter and compares it with a threshold detector where the threshold taken is at half of peak energy.

$$E_{threshold} = \frac{E_p}{2} \qquad \begin{cases} E_{received} \leq E_{threshold} \rightarrow 0 \\ E_{received} > E_{threshold} \rightarrow 1 \end{cases}$$

Equation 3.2.2.6 L-PPM with threshold.

After threshold decoding one can compare the transmitted with the received signal to calculate the BER, to make that one can evaluate the different slot of two signals and divide the total for the number of slot but this is the **Slot error ratio (SLER)**.
As one can see in the following, at this point one can compare the SLER with theoretical SLER, and after this the way to obtain the BER is the same for both function.

Theoretical L-PPM error probability:

To obtain the theoretical L-PPM error probability in AWGN noise the procedure is similar to the one applied for OOK error probability.
One can writing the probability of the slot error equation 3.2.2.7.

$$\begin{cases} P_{slot} = P(0)Q\left(\frac{Threshold}{\sqrt{\frac{N_0}{2}}}\right) + P(1)Q\left(\frac{\sqrt{E_{peak}} - Threshold}{\sqrt{\frac{N_0}{2}}}\right) \\ P(0) = \frac{L-1}{L}; P(1) = \frac{1}{L}; Threshold = \frac{\sqrt{E_{peak}}}{2} \end{cases}$$

Equation 3.2.2.7 L-PPM error probability equations. Where P(0) and P(1) are the probability to obtain respectively 0 or 1, Q is the Qfunction explained before. Epeak is the peak energy of the most energy slot of the symbol, N0 is the spectral noise density, threshold is the level of 0 and 1 separation. L is the number of slot of one symbol.

The correct threshold should not stay perfectly in the middle of the level but for small error probability one can approximate [6] the threshold with $\frac{\sqrt{E_{peak}}}{2}$ (in the middle of the minimum and maximum energy). With this approximation, one can proceed with the substitution to obtain the Slot Error Ratio:

$$P_{slot} = \frac{L-1}{L}Q\left(\frac{\frac{\sqrt{E_{peak}}}{2}}{\sqrt{\frac{N_0}{2}}}\right) + \frac{1}{L}Q\left(\frac{\sqrt{E_{peak}} - \frac{\sqrt{E_{peak}}}{2}}{\sqrt{\frac{N_0}{2}}}\right) = \frac{L-1}{L}Q\left(\frac{\sqrt{E_{peak}}}{2\sqrt{\frac{N_0}{2}}}\right) + \frac{1}{L}Q\left(\frac{\sqrt{E_{peak}}}{2\sqrt{\frac{N_0}{2}}}\right) =$$

$$= \frac{L-1}{L}Q\left(\sqrt{\frac{E_{peak}}{2N_0}}\right) + \frac{1}{L}Q\left(\sqrt{\frac{E_{peak}}{2N_0}}\right) = \left(\frac{L-1}{L} + \frac{1}{L}\right)Q\left(\sqrt{\frac{E_{peak}}{2N_0}}\right) = \boldsymbol{Q}\left(\sqrt{\frac{\boldsymbol{E_{peak}}}{\boldsymbol{2N_0}}}\right)$$

$$\boldsymbol{Q}\left(\sqrt{\frac{\boldsymbol{E_{peak}}}{\boldsymbol{2N_0}}}\right) = \boldsymbol{Q}\left(\sqrt{\frac{\boldsymbol{LE_s}}{\boldsymbol{2N_0}}}\right) = \boldsymbol{Q}\left(\sqrt{\frac{\boldsymbol{M * L * E_b}}{\boldsymbol{2N_0}}}\right)$$

Equation 3.2.2.8. Demonstration and equation of theoretical Slot error probability (SLER) of L-PPM modulation.

One can compare the output of simulation and the output of the theoretical equation to obtain the PPM SLER.

Furthermore if one would obtain the Symbol error probability and the Bit error probability from the previous data one can make another two mathematical step:

$P_{sym} = 1 - (1 - P_{slot})^L$

Equation 3.2.2.9 equation to obtain SER from SLER.

The previous formula can be easily understand with the following consideration 3.2.2.10:

$$\begin{cases} 1 = P_{slot-error} + P_{slot-correct} \\ (P_{slot})^L = P_{symbol} \end{cases} \rightarrow$$

$$\rightarrow \begin{cases} P_{slot-correct} = 1 - P_{slot-error} \\ (1 - P_{slot-error})^L = P_{symbol-correct} \end{cases} \rightarrow P_{sym-err} = 1 - P_{sym-correct}$$

Equation 3.2.2.10 equation to explain how obtain SER from SLER

As a last step to obtain the BER function one can use the conversion formula on [14] to obtain equation 3.2.2.11.

$$\boldsymbol{BER = P_{bit-error} = \frac{\frac{L}{2}}{L-1} * P_{sym-error}}$$

Equation 3.2.2.11 L-PPM theoretical BER function [14].

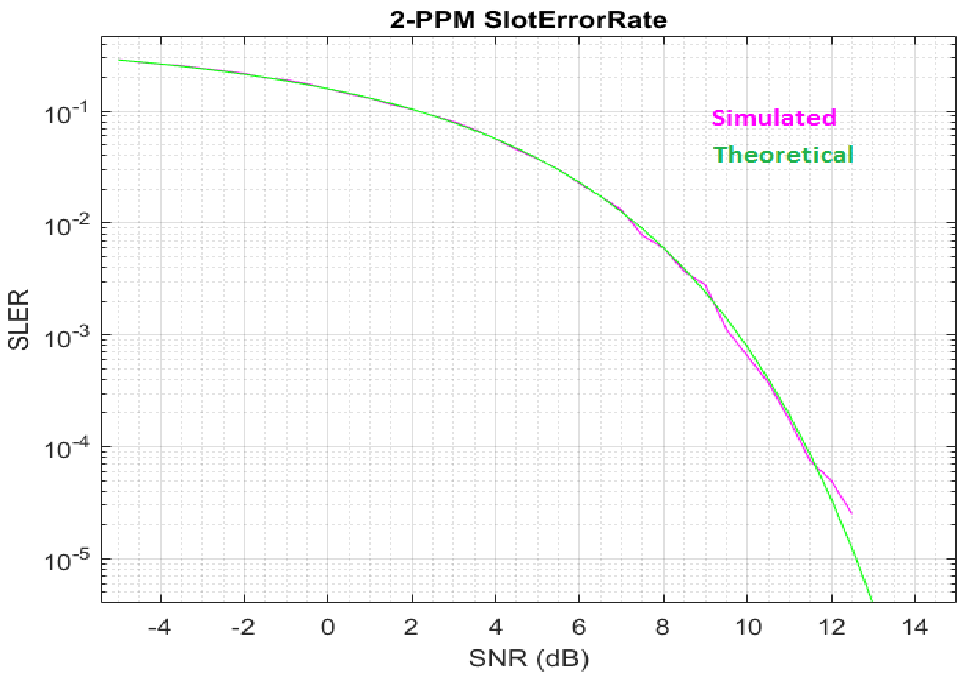


Figure 3.2.2.4: 2-PPM SLER simulation with random generate signal of 40E3 slots. Green is theoretical function and magenta is the simulated function. Figure made with Matlab code appendix A.3

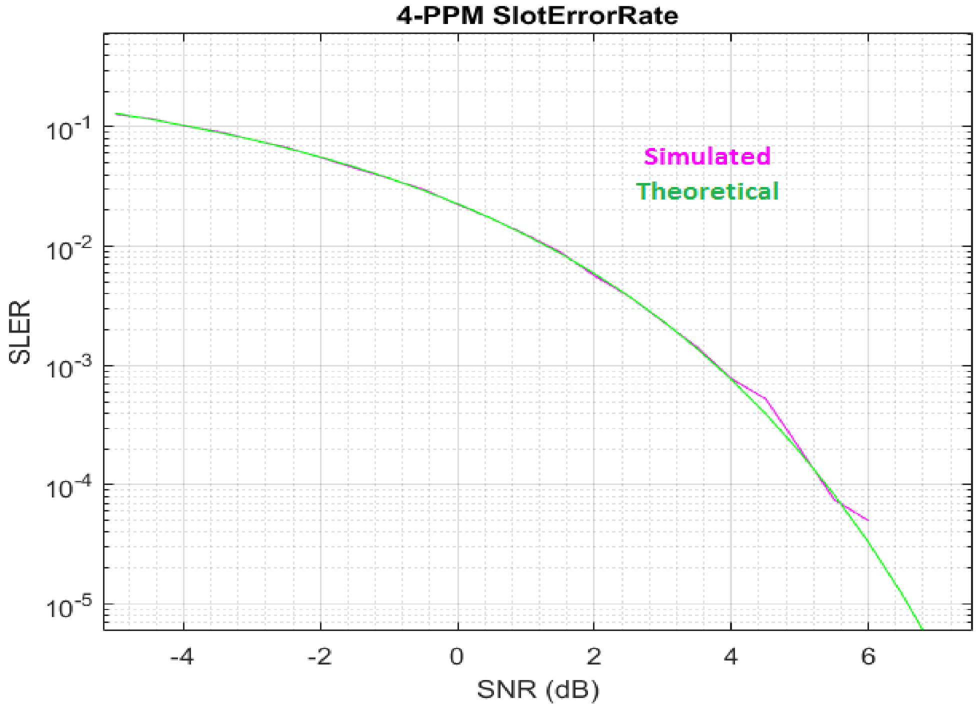


Figure 3.2.2.5: 4-PPM SLER simulation with random generate signal of 80E3 slots. Green is theoretical function and magenta is the simulated function. Figure made with Matlab code appendix A.3

Figure 3.2.2.4 and 3.2.2.5 compare the simulated 2-PPM and 4-PPM BER with respectively the 2-PPM and 4-PPM theoretical BER (in APPENDIX B.2 there are another two BER function). Examining the figure we can say that the theoretical BER calculated fits well the simulated BER, from now we approximate the BER (or SLER) of L-PPM with the theoretical function 3.2.2.11 (or 3.2.2.8) in this way, like for the OOK, the complex calculations for the L-PPM BER are reduced to the use of the formula 3.2.2.11 and 3.2.2.8.

3.2.3 Simulation results

The results graphics from section 3.2.1 and 3.2.2 seem to confirm the accuracy of the theoretical function for the OOK and L-PPM modulation.

For this reason in the remaining of 3.2.3 section the theoretical function have been used instead the simulation algorithm.

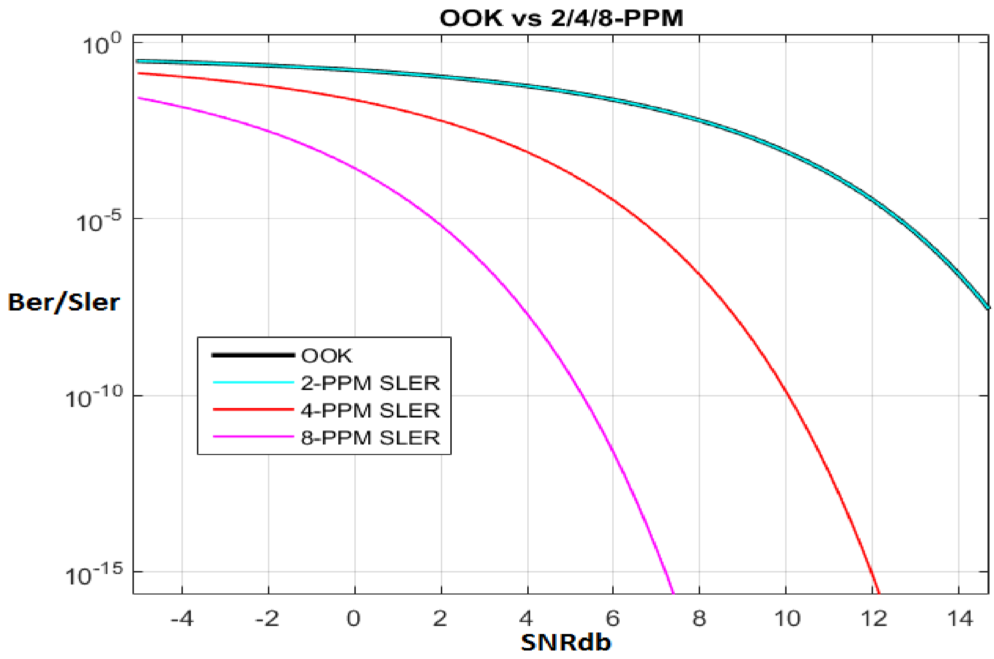


Figure 3.2.3.1: BER/SLER of OOK, 2-PPM, 4-PPM, 8-PPM comparison. X-axis is SNRdb, Y-axis is BER/SLER.

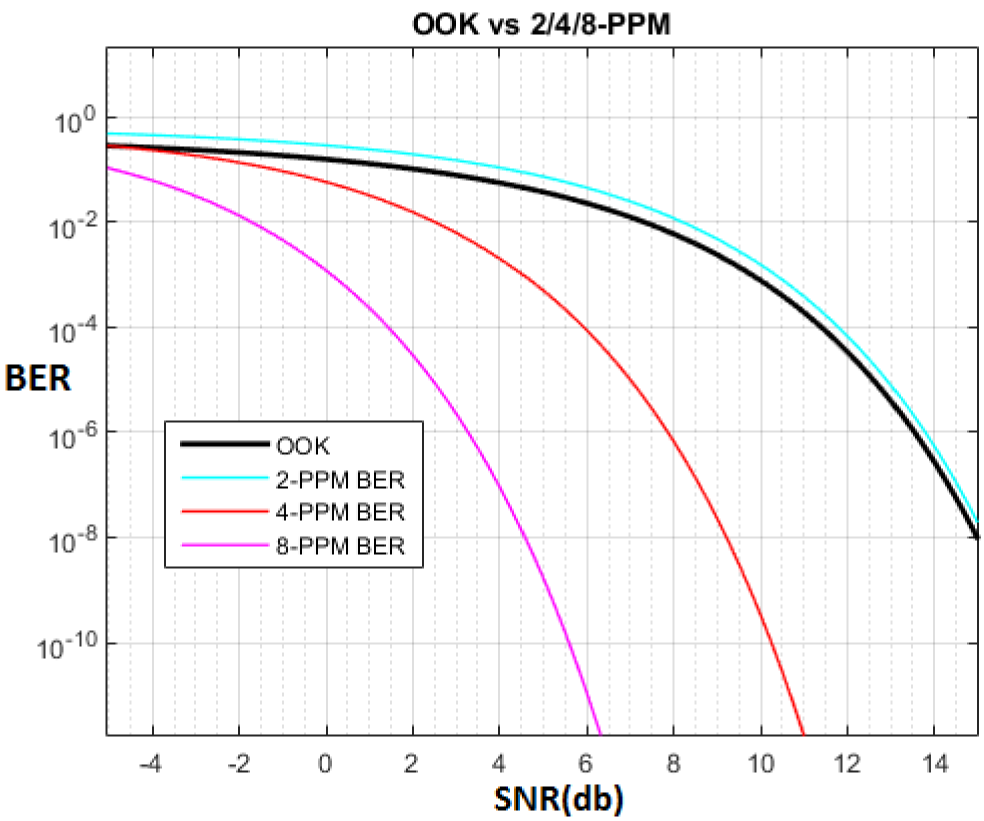


Figure 3.2.3.2: BER of OOK, 2-PPM, 4-PPM, 8-PPM comparison.

As one can see in the graphic 3.2.3.1 and 3.2.3.2 the OOK is very similar to 2-PPM BER. Furthermore the previous graphics seem to confirm that the BER of the L-PPM decreases

with increasing of the L number of pulses of the PPM modulation at fixed energy. This could be a great advantage for the L-PPM respect OOK because with less energy per bit the L-PPM can communicate with less bit error. Unfortunately the occupied bandwidth from L-PPM increases with the L increase. If one takes the example 2.4.7 notes that the band of L-PPM is $f_{L-PPM} = \frac{L}{M} * f_{OOK}$. This means that with equal band an OOK can transmit $\frac{L}{M}$ times faster compared to L-PPM (2 times respect 2-PPM, 8 times respect 8-PPM, etc…).
One needs to reach a compromise between the bandwidth and the BER but considering that we say chapter 1: the VLC bandwidth is almost unlimited and in chapter 3 section 1: the VLC received simulated power is very small the best solution seems to be the L-PPM modulations.

4 Arduino prototype

After the theoretical and simulation work we decided to implement a small prototype to confirm the previous results.

A good cheap and easy solution to implement a VLC prototype is presented in [15]. This solution consists in the use of two Arduino microcontrollers: one to implement the LED transmission and second to implement the photo-receiver reception. The modulation used to transmit/receive signal is the OOK (chapter 2.4.2).

To design our prototype we use:

1. Two Arduino Uno microcontrollers
2. Three LEDs LW514, the best LEDs model cited in [4]
3. Two Photo-Receiver SFH 213, the LEDs used in [15] and cited in [4]
4. Various resistors
5. Many breadboards and cables
6. One notebook PC with two usb connectors

To transmit/receive we use the **OOK** modulation because we study this in our previous work and we put the LED and Photodiode at fixed distance **14.5cm**.

The experiment we made can be divide in three steps that are described in sections 4.1, 4.2 and 4.3:

4.1 First bits: verify the possibility of transmitting bits between the two Arduino.

4.2 Synchronization and ASCII transmission: we transmit word with ASCII conversion and we address the synchronization problem

4.3 OOK BER test: we estimate the BER of our transmission.

For these experiments we calculated the threshold level using the two Arduinos and the code in appendix C.1. For 4.1 and 4.2 the threshold level is calculated one time before the experiment start, for 4.3 the process is similar but is repeated every times one takes a dataset (explain in 4.3). Figure 4.0 shows a setup for experiment 4.3.

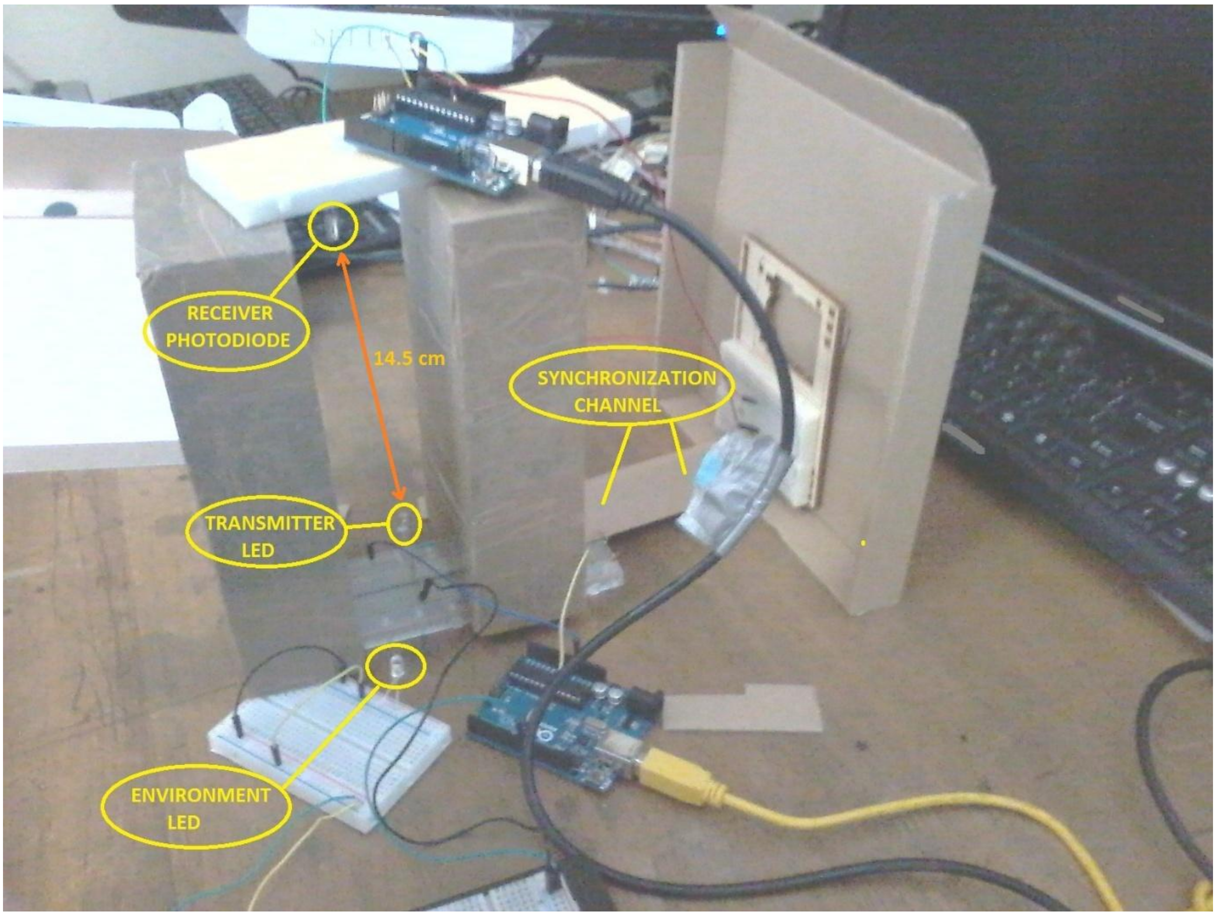


Figure 4.0 Experimental setup for section 4.3

4.1 First bits

The first thing we verify is the feasibility of a bits transmission between the two Arduinos. To do this we mount the two electrical circuits showed in figure 4.1a and 4.1b at fixed distance 14.5cm and we use the code in C.2 appendix.

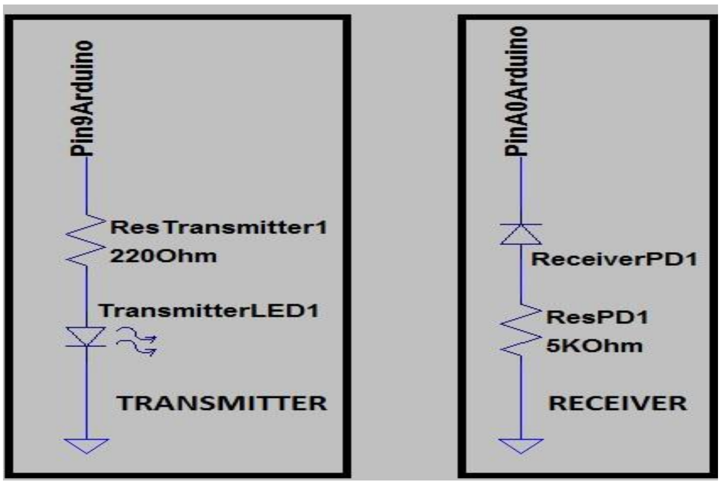


Figure 4.1a e b: On left a the transmission circuit on right b the receiving circuit.

The program C.2.1 (transmission) receive in input the bits (a bits array is present in program) to transmit and transform that in a sequence of luminous pulses (if bit is 1 set the LED at HIGH if the bit is 0 set the LED at LOW) every pulse is followed by time delay corresponding to T_{bit} of the OOK modulation. The program at receiver evaluates the luminous pulse every time delay time. If the received pulse value is higher than a threshold level (setting with C.1) the program prints a 1 at contrary the program prints 0.

First bits result:

We perform the experiment varying the delay time (descending from long time to short time). With high delay time the system works and transmits the signal without error, with the reduction of delay time the system starts to present systematic error: after many transmitted bits that depend inversely on the delay time the received message becomes totally incorrect. This error is caused from the loss of synchronization between the LED and Photo receiver. The previous considerations are only qualitative.

4.2 ASCII transmission Synchronization

In the second experiment we try to transmit a sequence of characters separated by a break time (different than delay time). The strings are inserted by the user who can write on the Arduino serial monitor. When the user presses the return key the string is sent. Afterwards the user can write other strings and press enter to send them. If the time between the two strings (or the length of string are too long) is too small the Arduino software misses some data.

One string is composed of many bits (for every character one needs 7bits) and from the experiment 4.1 we understood that for high numbers of bits we need a method for synchronize the transmission in order to avoid the synchronization error. In other word we need to communicate with the receiver when our transmission starts and ends. To do this we use another LED (SyncroLED) and photodiode (SyncroPD) system: when the transmission starts the SyncroLED turns ON while when the transmission is over the SyncroLED turns OFF.

To achieve this we write and use the Arduino sketch C.3.1 and C.3.2. and we mount the electrical circuits described in 4.2.

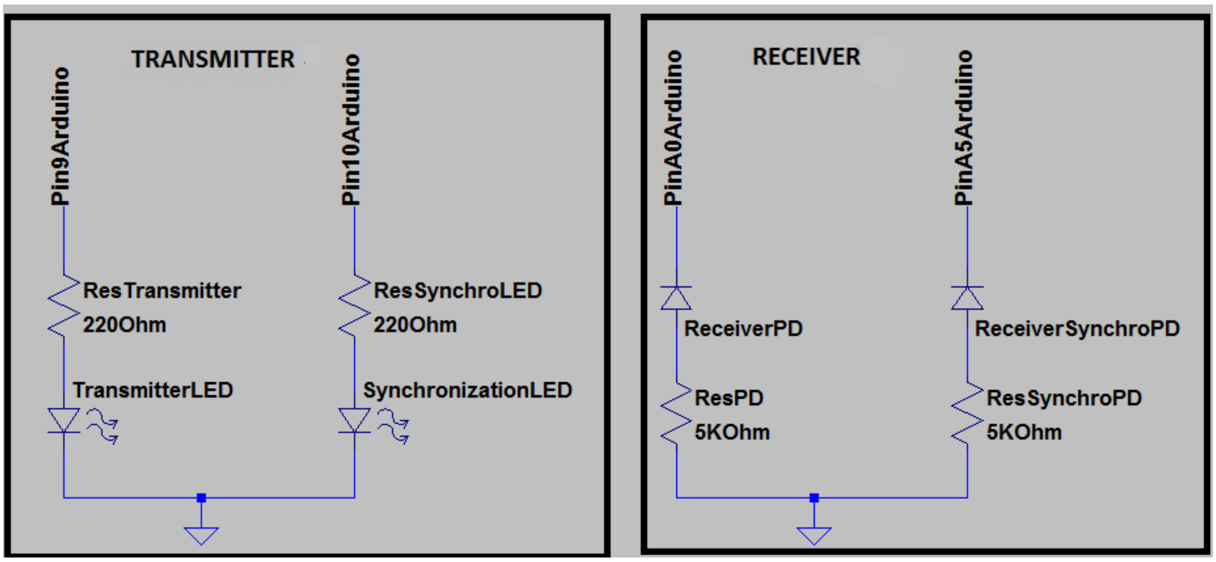


Figure 4.2: On left the transmission circuit on right the receiving circuit.

An example of the result from experiment is shown in figure 4.3.

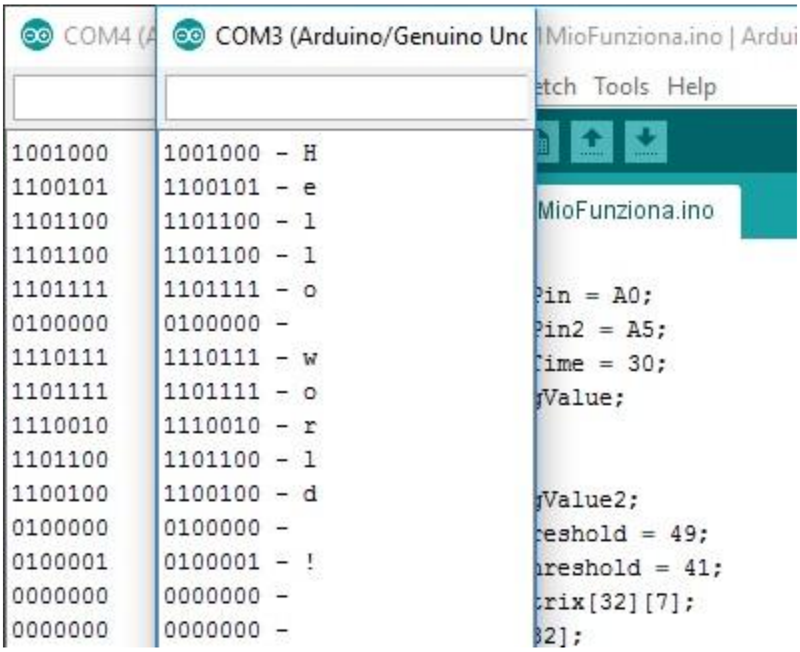


Figure 4.3 Transmission/receiving example. On left the transmitted signal in bits on center the received signal in bits and character, on right a piece of receiver Arduino code.

Results:

We acquired several sets of data. The results of the experiment are listed in table 4.1.

Delay time (ms)	Synchronization error bit (included)
1	First bit/Second bit
2	Third bit
3	Fourth/Five bit
5	Seventh
10	Fourteenth
20	Twenty-seventh
25	Thirtieth

Table 4.1 Synchronization error bit.

Table 4.1 shows after how much error the signal is affected from synchronization errors, the bit indicated in the second column is the first wrong bit, the following bits are wrong as well. As we say in section 4.1 the table shows that this error depends inversely on delay time.

4.3 Estimation of an OOK BER

To calculate the BER of an OOK system described before in the theoretical chapter one should implement an algorithm that can transmit tens of thousands of bits and can receive the same number. To do this we need to consider the section 4.2 to avoid the synchronization error and to maximize the speed of our system in order to collect the data. To avoid the synchronization problem and to have a good speed of transmission, after many tests we decide to use a delay time of 20ms with a pause between two strings of 50ms. We transmit every time an automatic default word ("abcdefgh") composed by eight characters (7*8 = 56 bits). In this way when one controls the received signal it is easy to check if the signal contains errors. An example of the received signal is shows in figure 4.4.

1100001;1100010;0100011;1000000;1000101;1100100;0100111;1001000;
1100001;1100010;1100011;1100000;0100101;0100110;1100111;1101000;
1100001;0100010;0100010;1100100;1100100;1000100;1100110;0101000;
1100001;0100010;1000000;1100100;0100001;1100100;1100101;0000000;
1100001;1100010;1100011;1100100;1100100;1100110;1100010;1000000;
1100000;0000010;0100011;1000100;1100101;1100010;0000101;0101000;
1100001;1000010;1100010;1100100;1100101;1100110;1000101;1101000;
1100000;1100010;1100011;0000100;0100100;1000100;0100111;1001000;
1100001;0100010;1100011;1100100;0100101;1100100;1100101;1101000;
1100001;1000000;1000011;1100000;0000001;1100110;1100111;1001000;
1000001;1000010;1100000;1100100;0100101;1000110;1000111;1101000;
1000001;0000010;1000011;1000000;1000001;1000010;1100111;1001000;
1100001;0100010;1100010;1100000;1000100;1000010;0100111;1001000;
1100001;1000010;1000011;1100100;1100101;0100010;1000111;1100000;
1100001;1100000;0000100;0100100;1100100;1100010;0000111;1101000;
1100001;0100010;0100010;0100000;1100101;1100010;1000111;1101000;
1100001;1100010;1100011;0000100;1000000;0000110;1100011;1001000;
1100000;0000000;0100011;0100100;1000100;0000110;1100010;0101000;

Figure 4.4 received signal serial print Arduino example.

In order to obtain these result we use the algorithms C.1 (described in 4) and C.4, the two algorithms in C.4 create a VLC communication with two LEDs and two photodiode similarly to C.3 but this time the transmission is repeated automatically.

To neglect the synchronization LED background light we mount the synchronization LED and photodiode in a small box (figure 4.0 synchronization channel on right). Moreover, we put all the system in another bigger closed box with a third LED inside (environment LED in figure 4.0 on bottom left) biased from an Arduino 5V pin and serial mounted with a 1KOhm (likes the other LEDs circuit). In this way, one can generate a constant controllable background light noise.

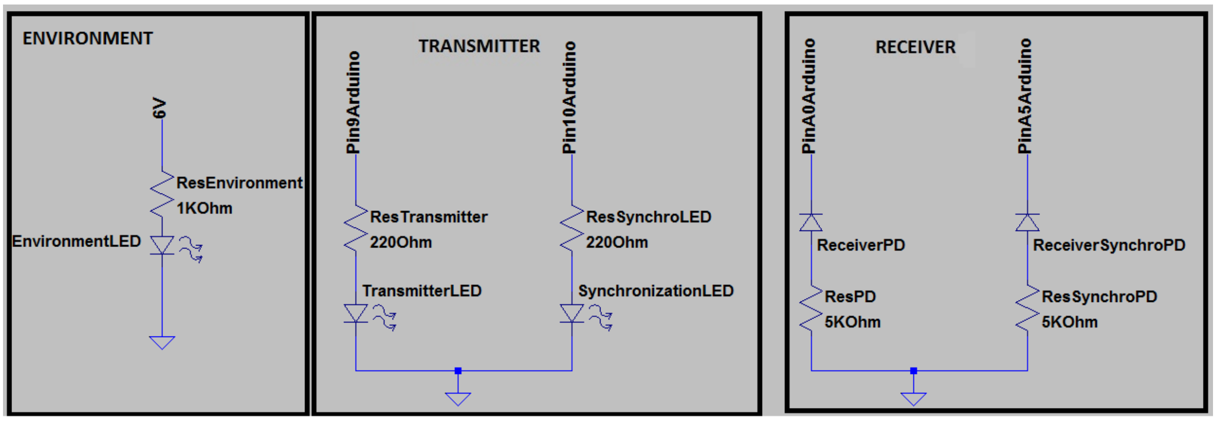


Figure 4.5 Transmission/Receiving and environment circuits. Left environment circuit, center the transmission circuit and right the receiving circuit.

The system shown in figure 4.5 can be used to do only one BER measure because the SNR is constant. To varying SNR we decide to increase every time the resistance of the transmitter LED. In this way the LED brightness decrease and consequently the photodiode generates every time a smaller current. The SNR ratio depends directly by the received current and inversely from the noise, the system is designed to have a constant noise then changing the resistance allows the SNR variation.
We use a breadboard placed outside the box and connected with the transmitter LED to change the transmitter LED resistance without touching the system.
The process for estimate the BER is the following:

1) Mount the resistance on the external breadboard
2) Start the code C.1 for transmitter and receiver Arduinos to calculate the threshold level
3) Set the threshold level on the C.4 receiver code
4) Start the code C.4
5) After a while the serial window of the receiver Arduino presents various received bit, we copy this in an excel file

One can reiterate this method for various resistance value.
After various test we decide to use a set of resistances between 30KOhm and 90KOhm, moreover to be sure that Arduino works properly for the background noise we decide to use an external generator with 6V voltage and a 1KOhm serial resistor.

Results:

We obtain various datasets for various resistances values; we decide to use Matlab to process the data. With a Matlab script, we import the data, we calculate the BER and we plot the obtained result.
The results were plot in figure 4.6 where on the left one can see the experimental plot and on right one can see the theoretical plot. The plot is logarithmic plot of the BER in function of the SNR [a.u.] where a.u. stands for arbitrary units. We use the a.u. because we were not able to calculate the absolute SNR of our system.
Comparing the two curves qualitatively one can see that they have the same trend. We were very satisfied with the results obtained.

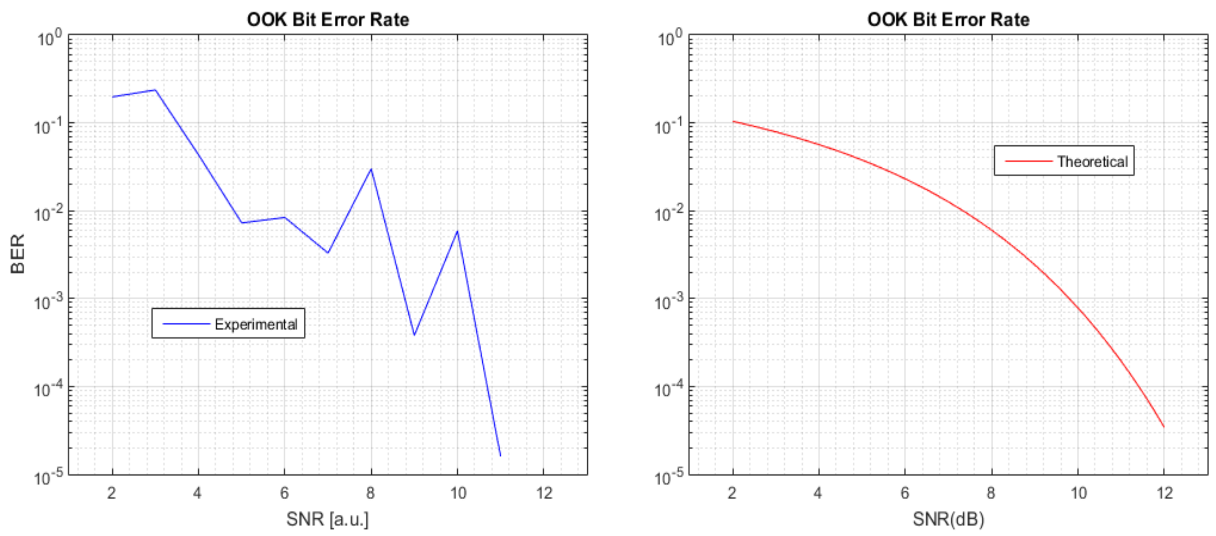


Figure: 8.2.2 Experimental versus theoretical BER. Left: experimental result, right theoretical result. Experimental and statistical uncertainties are too small to be appreciated on this scale.

5 Present and future applications

The possible applications of VLC are many, this chapter attempts an overview of these technologies.

VLC is a new technology that is emerging in the few last years for different reasons: the expensive price of RF equipment compared to LED light, the saturation of the radiofrequency spectrum and the fact that the VLC LED are not dangerous makes the VLC a good candidates to help the existing communication system.

Various uses for VLC systems have been suggested. In the following we give a description of various possible hypothesized application (section 5.1) and after we treats VLC existing technologies (section 5.2).

5.1 Future applications

5.1.1 Hybrid system (Radio + VLC)

The thought of replacing all radio communications with VLC is impracticable, just think the shadowing problem that affects light transmission. However VLC has many advantages with respect to radio transmission (chapter 1), in particular for indoor environments. The most popular solution is to support the radio Wi-Fi with VLC: if VLC signal isn't present the radio connection can replace the VLC and viceversa. The system must be able to select the best solution at the right moment.

5.1.2 Li-Fi

Harald Hass coined the Li-Fi term. Harald Hass is a professor of Edinburgh's University and he is the co-founder of pureLiFi.He is one of the first that project working VLC system. Hass demonstrated the VLC technology during a TED talk: he spoke about the VLC technology and how it could be applied to every lamp with huge energy savings respect to radio wireless. Furthermore in the show he used a LED to transmit an HD video to computer using a VLC system.

The difference between VLC and Li-Fi is very thin: VLC is a general concept, it can include all types of visible light communication also the unidirectional and low data.

LiFi instead contains only a restricted class of VLC. Li-Fi systems are capable of:

- Roaming: allows users to move between lights maintaining the connection

- Multiple access: one single light can enable different people to connect to internet at the same time

This definitions isn't accurate because it was coined recently and all the VLC technologies are new and mutable.
For many researcher the VLC and Li-Fi word are interchangeable.

5.1.3 Sites for VLC (Hospital, Airplane, underwater)

VLC can be used safely in hospital where the radio transmission can interfere with the hospital equipment. For the same reason VLC can be used on airplanes.
Underwater the radio waves do not propagate far this because the high attenuation coefficient. The light, instead, propagates very easily through the water, for that, VLC could be a great technology to transmit underwater signals.

5.1.4 High density wireless

High density wireless is a wireless with high energy density to transmit more fast and more amount of information. A good example is university classroom: if every students using the wireless at same time the network should probably collapse. With VLC this issue can be avoided.
High density wireless is difficult to realize with radio waves because if one increases the radio waves energy this can becomes dangerous for human safety, furthermore as we say before the cheapness and simplicity of VLC system make them more attractive for built an high density wireless.

5.1.5 Smart drive/shop/city.

The car, the semaphore, the public illumination and private shop signs are all sources of lights, even more in most case that lights are generated from LEDs sources.
A car, a semaphore, a public illumination system and private shop signs are all sources of lights: moreover in most case that lights are generated from LEDs sources.
With that premise one can consider to use that sources to illuminate and transmit information.
In that way many solutions are possible to imagine.

- Figure 5.1.5.1 show a three colors VLC music transmission system, each different color transport different sound.

- Figure 5.1.5.2 show a market positioning system. The visible light receiver on the shopping cart acquires the product information and the shopping cart IDs (Exact Position and Time) are stored in a memory card when the shopping cart goes through the passages.
- Figure 5.2.5.3 show traffic lights VLC system, the traffic lights communicate information with the small computer showed in figure.
- Figures 5.2.5.4 show “SMART” car VLC system, the car headlights communicate with other car headlights, with the traffic lights or with the road signs.
- Figure 5.2.5.5 show “SMART” braking system. The headlights car communicates the braking to other cars headlights

Figure 5.1.5.1: VLC music system. Prototype presented by SONY and Agilent Technologies [16].

Figure 5.1.5.2: SMART market VLC system [16].

Figure 5.1.5.3: The neighbor information distribution system from a traffic light [16].

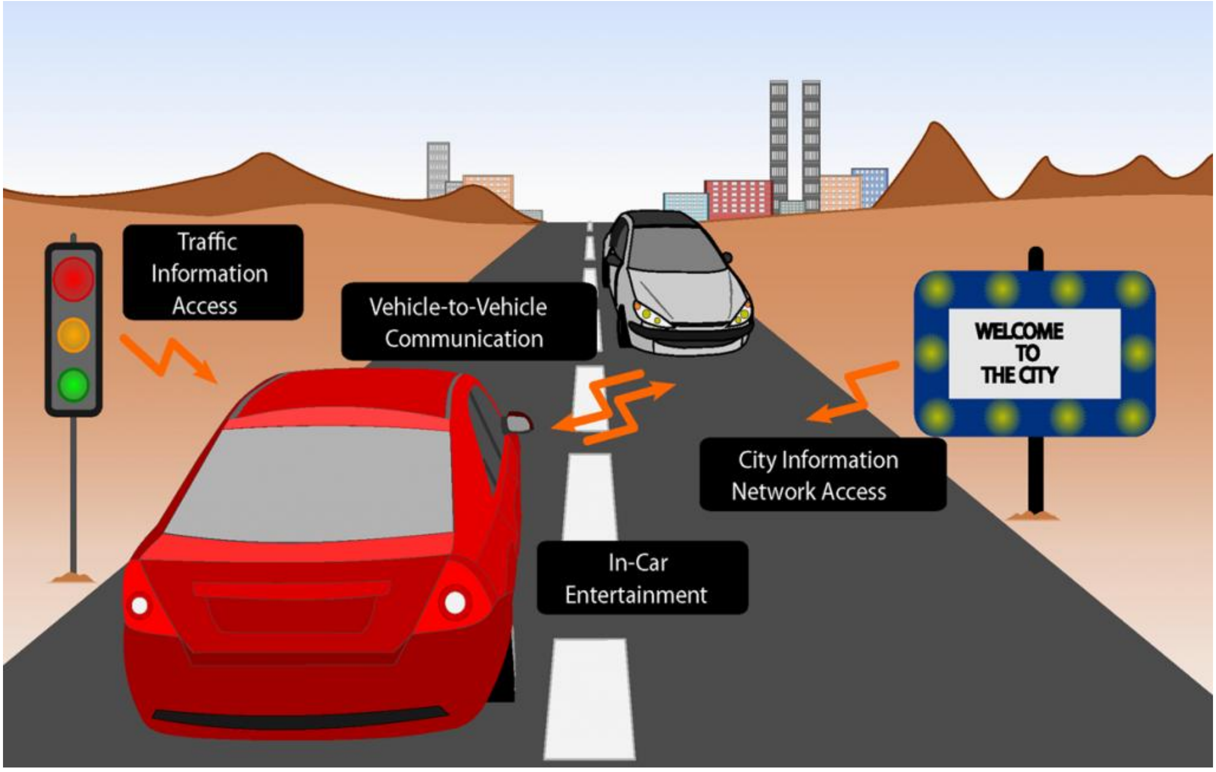


Figure 5.1.5.4 VLC for automotive applications [Web].

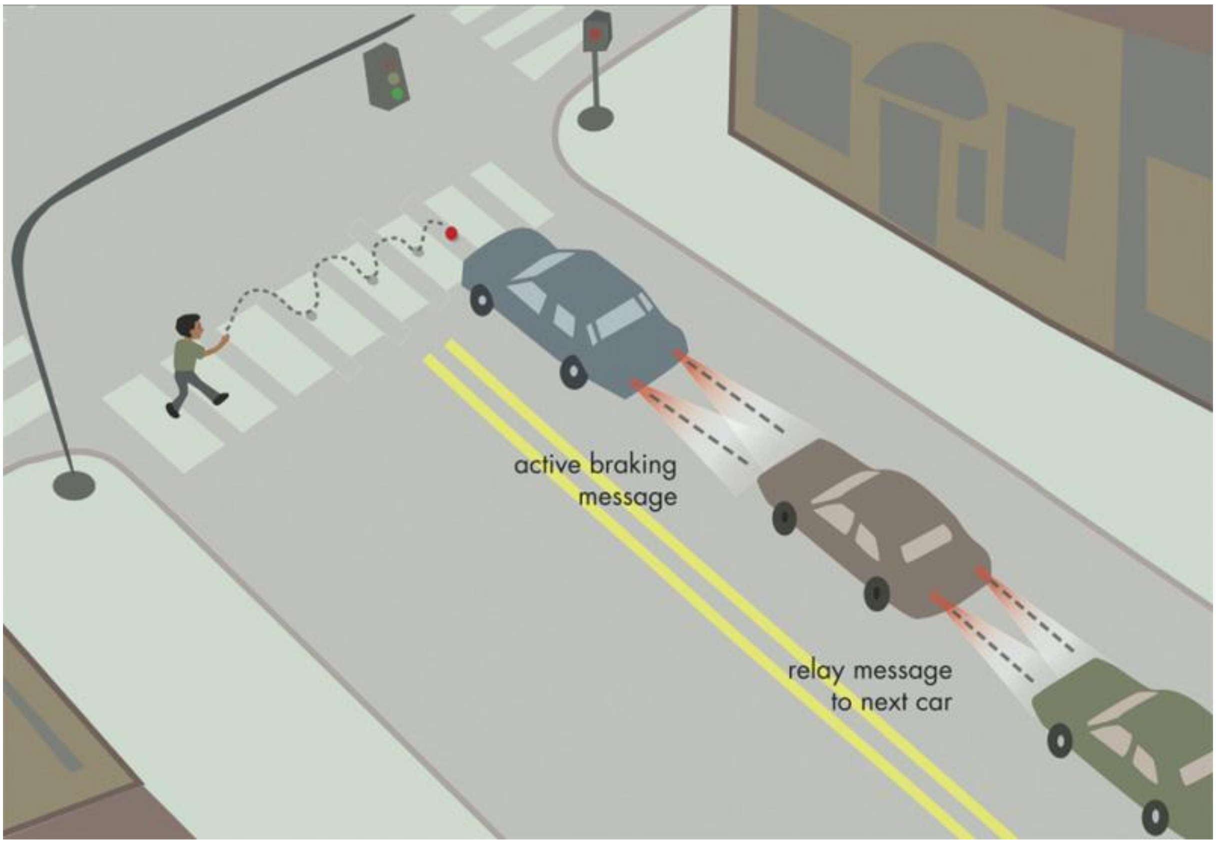


Figure 5.1.5.5 VLC braking system [Web].

5.2 Existing applications

We dedicate some time to find some existing applications and their features. We found all information on the web, unfortunately sometimes information are incomplete, often because this product are intended for sale.

5.2.1 MOMO by Axrtek

Axrtetk is company that designs smart light system.
Axrtek realizes it defining a set of rules, technologies and software that "transform" the indoor illumination system into a smart light system.
Smart light system can provide both the illumination and wireless communication.
The most important Axrtek device is the MOMO reported in the figure 5.2.1.1.

Figure 5.2.1.1 MOMO VLC LINK transmitter/receiver devices.

As one can see the device is composed of LED, photoreceiver, Ethernet and supply sockets.
MOMO is a transceiver that uses RGB LED light to send and receive data signals: with three color channels MOMO has potentially triple throughput of a single-color VLC system.
MOMO is a bidirectional device, with two MOMO one can realize a duplex system because every device has LED to transmit and photodiode to receive signals.
These date are taken from MOMO web site:

- Max throughput 300Mb/s data transfer rate (bidirectional)
- RGB color channel orientation
- Streaming coverage of up to 25 feet (7,62 meters)

- Capable of handling 256 concurrent connections
- Line of sight needed
- Max angle of 60 degrees
- Power line communication (PLC) compatible
- Session Initiation Protocol (SIP)
- Four Ethernet socket ports
- Security: signals don’t pass through the walls

For the future Axrtek version the company has plans to upgrade the MOMO technology to a second generation system: throughput should increment to 1Gb/s and the distance to 330 feet (around-100 meters).

5.2.2 Fraunhofer Institute for Telecommunications VLC systems

Fraunhofer Heinrich Hertz Institute (HHI), is an organization of the Fraunhofer Society based in Berlin that deals with telecommunications, It has a large photonics networks department , and within it a sub-department that studies optical indoor networks and VLC. In the HHI website one can find some devices that are based on VLC and IR shown in figure 5.2.2.1 and 5.2.2.2.

Figure 5.2.2.1 Fraunhofer VLC transmitter/receiver devices.

These devices have LED to transmit and photodiode to receiver. Furthermore a power line and Ethernet socket are present.

The Fraunhofer device is very similar to MOMO. Below some features reported on the Fraunhofer web site:

- Light Source: any current high-power LED
- Bidirectional data exchange
- Dynamic rate adaptation
- Adaptable optics depending on application
- Peak data rate 1 Gbit/s
- Low latency (< 2 ms)
- Universal RJ45 Ethernet Interface
- Footprint: 87 mm x 114 mm x 42 mm (without lenses)
- Security: signal don’t pass through the wall

Picture 5.2.2.2 Fraunhofer LED-Blackhaul link

This is another link device based on light communication. This is based only on IR spectrum (IR is not visible for human eye).

The features of this object are:

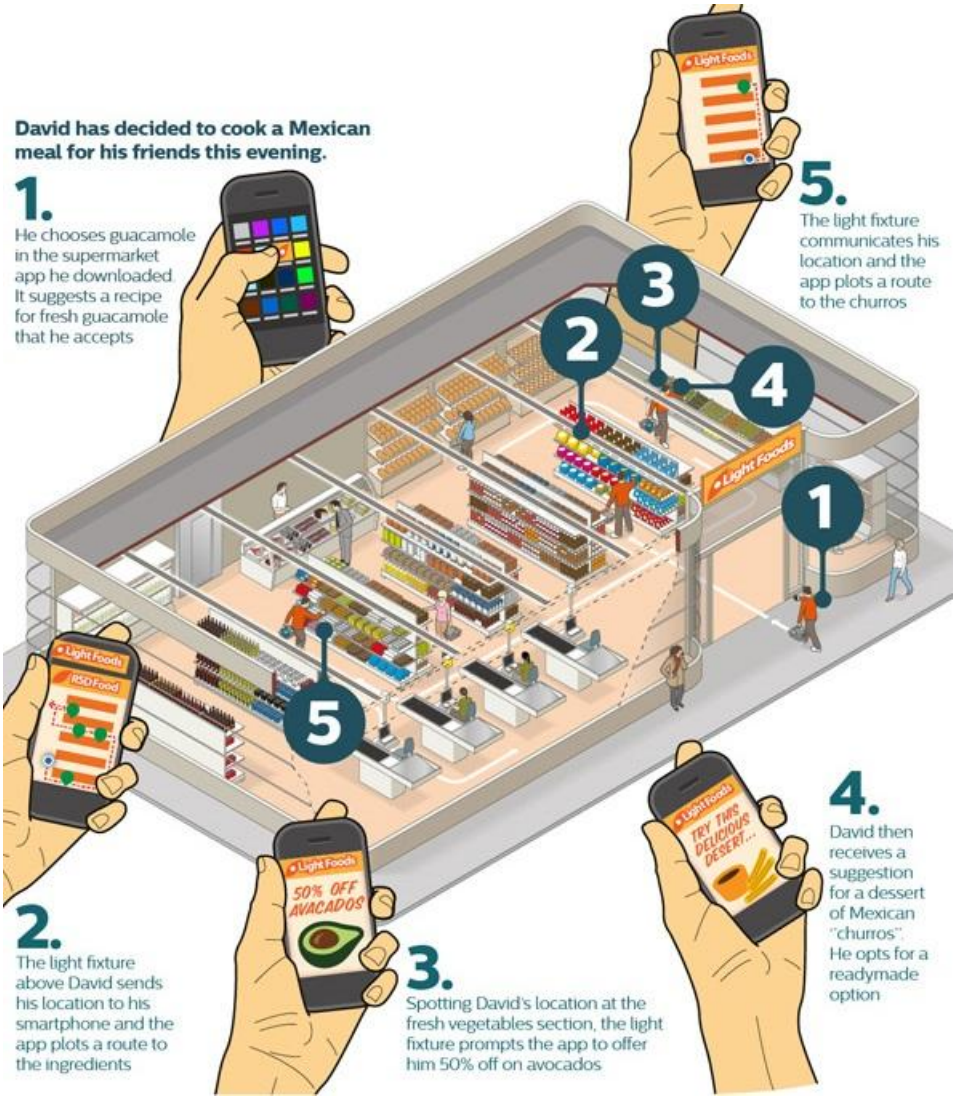
- Infrared LED based
- Easy alignment:
  500 Mbps over 100 m
  250 Mbps over 200 m
- Bidirectional data exchange
- Dynamic rate adaption
- Low latency (< 2 ms)
- 1 GbE chipset and interface
- Footprint and weight: 240 mm x 230 mm x 130 mm, 3 kg

- Security: signal don't pass through the wall

5.2.3 Philips smart lighting system

A shopping center in France has become the first in the world to install a smart light system created by Philips. The System employ LED-based technology for indoor illumination and customer communication.

With this technology the shopping center can save light energy and guarantee an internal wireless positioning system that connect customers and products.



Picture 5.2.3.1: example of how the Philips smart lighting system works [17].

Some working characteristic are listed in table 5.2.3.1.

Turning on app	When mobile device is in proximity of LED it automatically start interacting with network. The app need to be installed before.
Technology used	Visible light communications
Speed	From 10Kb/s to 500Mbit/s for LED
Range	Distance of 1 – 2 Km
Energy consumption	No separate infrastructure are required. In store energy efficient LEDs which are used for store lighting perform task
Hardware	In-store light fixtures
Internet connection	No connection to internet is required for propagations

Table 5.3.2.1 philips smart light system [17]

5.2.4 Velmenni Jugnu devices

Velmenni is a New Delhi startup working to launch a new device
that involves VLC: Jugnu are a new generation of smart LEDs that are project to illuminate and transfer data with visible light, in short jugnu are LI-FI LEDs.
The Velmenni has just realized a VLC project that use computer to transfer serial data from it to a photodiode that is connected to a micro-controller board realized with Arduino.
Now Velmenni is working on a project that allows to transfer data from LED to LED or to mobile phone and internet. Therefore the Velmenni is working to an Android app that can receive data from Jugnu LEDs.
The other project of Velmenni are not VLC based (the are focused also in home automation and traffic light systems) but, in future, if the jugnu project will be successful, the integration of VLC in other Velmenni’s project should be easy.

5.2.5 PureLiFi

PureLifi is a light communications technology company, that design Li-Fi system
The pureLifi has three projects based on VLC that in generational order are: Li-1st, Li-Flame, Lifi-X.

Li-1st product is VLC device that is composed of three pieces (in the figure from left to right): the ceiling unit, a LED lamp and last on the right the desktop unit. The ceiling unit is connected to internet and acts as modulator in downlink and demodulator in uplink. The LED transmits the informations using light and finally the desktop unit is a photoreceiver connected to a computer: it works as demodulator for downlink and as modulator for uplink connection.

Figure 5.2.5.1 Li-1st from left ceiling unit, LED lamp and desktop unit (photoreceiver).

Li-1st (figure 5.2.5.1) is full duplex communication device that could reach 5 Mbps in both downlink and uplink with the range of 5 meters. Furthermore the Li-1st can transmit with NLOS signal (can be seen in the demonstration video on the PureLIFI website).

Li-Flame (figure 5.2.5.2) is the second pureLifi VLC device. It was publicly presented at Mobile Word Congress in Barcelona on 2-5 March 2015 at the Scottish development international stand.

The general functions of this object is the same as the Li-1st but the specifications are differents.

The Li-Flame is composed of two units: Li-Flame ceiling unit and Li-Flame desktop unit.

The Li-Flame ceiling units has the following specs:

- Downlink speed up to 10 Mbps
- Range of 3 meters (max 5 meters)
- Data and power via Ethernet port (power over Ethernet POE)
- Multiple users access supported
- Handover control enables seamless switching through AP

The Li-Flame desktop unit has the following specs:

- Uplink speed of 10 Mbps through infrared connection
- Connects to computer via USB
- the desktop unit is more portable respect the Li-1st because it has a battery and the transceiver can be adjusted by user to optimize the connection

The Li-Flame is half duplex and not full duplex like Li-1st.

Figure 5.2.5.2 left: Li-Flame ceiling units, right: Li-Flame desktop unit

LiFi-X (figure 5.2.5.3) is the third and fastest product of pureLiFi company.
The LiFi-X was launched for first time in February 2016 at mobile wold congress.
The LiFi-X is the evolution of the Li-Flame system.
Following the main features of LiFi-X:

- full duplex communication
- uplink and downlink speed of 40 Mbps
- full mobility portable USB powered desktop station

The LiFi-X access point (AP):

- support power over Ethernet (POE) or power line communications (PLC)
- permit the multiple access
- handover control enable seamless switching through access point (AP)

The LiFi-X station:

- is powered through USB 2.0
- supports handover, permit user to maintain their wireless session if they move

Picture 5.2.5.3 left LiFi-X Access point, right LiFi-X station (desktop units)

5.2.6 RONJA

RONJA (Reasonable Optical Near Joint Access) is an optical communication wireless system. Ronja is a product of Twibright Labs from Czech Republic, that created a first experimental prototype in 1998 and launch an effective working device in 2005.

The Ronja devices are composed of two full duplex transceiver (two transmitters and two receivers) working with red light or with infrared. The devices are used to communicate within a middle range, about 1.4 km building to building.

Ronja has be defined from authors a "user controlled technology": building instructions, schematics and blueprints are published under free documentation license (GNU).

Ronja devices are three: Tetrapolis, Inferno and 10M Tetrapolis. The characteristics of these three devices are listed in table 5.2.6.1.

Device name	**Tetrapolis**	**Inferno**	**10MTetrapolis**
Gross data	10Mbps	10Mbps	10Mbps
Transmission mode	Full duplex (half duplex is supported)	Full duplex (half duplex is supported)	Full duplex only
Nominal range	1.4Km with 130mm lenses	1.25Km with 130mm RX loupe lenses and 90nm TX loupe lenses	1.4Km with and 130mm lenses
Minimum distance	¼ of nominal range	¼ of nominal range	¼ of nominal range
Power consumption	285mA at 12VDC (3.42W), 2W from external heating power supply (switchable off)	335mA at 12VDC (4.02W), 2W from external heating power supply (switchable off)	300mA at 12VDC (3.6W), 2W from external heating power supply (switchable off)
Operating wavelength	Visible, 625nm, 100nm spectral width (red-orange)	Infrared, 875nm wavelength, 37nm width	Visible, 625nm, 100nm spectral width (red-orange)
Optical ouput	17.2mW	30mW	17.2mW
Divergence cone half angle	1.9 mrad (130mm aperture transmitter lens)	3 mrad (90mm aperture transmitter lens)	1.9 mrad (130mm aperture transmitter lens)
Estimated optical EIRP	20 kW (130mm aperture lens)	Not present	20 kW (130mm aperture lens)
Operating temperature	-30→+70ºC (optical outdoor part) 0→+55 ºC (indoor part twister2)	-30→+70ºC (optical outdoor part) 0→+55 ºC (indoor part twister2)	-30→+70ºC (optical outdoor part) 0→+55 ºC (indoor part AUI interface)
Operating humidity	up to 100% (condensing) with lens heating on (outdoor part), up to 95%	up to 100% (condensing) with lens heating on (outdoor part), up to	up to 100% (condensing) with lens heating on (outdoor part), up

	with lens heating off (indoor part)	95% with lens heating off (indoor part)	to 95% with lens heating off (indoor part)
Weight	15.5Kg	15.5Kg	15.5Kg
Required	4Km at maximum range	4Km at maximum range	4Km for uninterrupted operation at full range
Optical modulation	BPSK plus 1MHz asynchronous 50% duty cycle square wave between packets	BPSK plus 1MHz asynchronous 50% duty cycle square wave between packets	BPSK plus 1MHz asynchronous 50% duty cycle square wave between packets
Indicators LEDs	Power, receive packet, transmit packet	Power, receive packet, transmit packet	Power, receive packet, transmit packet
Aiming system	Visual, reflector for transmitter and DC voltage signal monitor port for receiver	Infra-visual, retroreflector + infrared camera for Tx and DC voltage signal strength monitor port for receiver	Visual, reflector for transmitter and DC voltage signal monitor port for receiver

Table 5.2.6.1 Ronja characteristics from web [18]

6 Conclusion

The need to be always connected to the network and to have access to ever larger volumes of data has led to the improvement of the existing transmission technologies and the creation of new transmission techniques. One of these new techniques is the Visible Light Communication (VLC).

In this work we study the VLC from a physical point of view. After researching existing literature, we describe the VLC physical layer blocks: the transmitter (LED), the receiver (Photodiode), the channel and the modulation block. From this, we give a general idea of the current status of VLC applications. We found that the two most used modulation schemes are the ON-OFF Keying (OOK) and the Pulse Position Modulation (PPM), we covered the LEDs and photodiodes VLC most important characteristics and we understand how works a VLC channel.

After the theoretical work we start to investigate how we can implement a simulation of the physical layer. We started simulating the LOS channel and two modulations: OOK and PPM. From these simulations, we got a profile of the transmitted/received power of a LOS channel and the BER function of the OOK and PPM. We concluded that these BER curves match very well the theoretical analytical curves.

Afterwards we decided to test the feasibility of a VLC system by building a small prototype. With two Arduino microcontrollers, we have built a communication system able to transmit status of characters at small distance (about 15cm) using the OOK modulations. This prototype work properly at small speeds. Moreover, we estimated the BER of the prototype obtaining a curve that qualitatively follows the trend of the theoretical/simulated curve.

The VLC overview is wide and changes continuously, our work investigates only a small part of this technology. However, we realized that VLC would probably evolves very quickly and spread widely in the near future in order to satisfy the higher speed demand of wireless network and because the VLC can provide a solution to the problems of lighting and data transmission at the same time.

Appendix A Matlab simulation code

A.1 LOS channel

```
%   VLCLOSCHANNEL
%   LAMBERTIAN LOS channel function
%   input_power is watt input vector received from LED
%   output_power is power after channel vector
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   LOS-Channel Parameters
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Input Signal
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
meanInputPower =1; %input luminous power 1 Watt
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Optical concentrator and filter
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tsfilter = 1;   %optical filter 1 neglected
csilimitdegree = 75;    %concentrator filter angle limit
csilimit = degtorad(csilimitdegree);
n = 1;    %concentrator filter index
% n = 1 makes gcsi about 1 gcsi neglected
% n = 1.5 - 1.6 is the glass index
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Room detail
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
lx=0;   ly=0;   lz=0;  %led position (0,0,0) is ceiling centre
%   reference system starting from ceiling centre
x=3.5;    y=3.5; z=2.5; %absolute room dimension
rz=1.5; %receiver height(z) fixed position from ceiling (meters)
% to find distance from floor (z-rz)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Led/Receiver tech detail
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
ar= 0.0001;       %receiver area m^2
resp = 1; %photodiode responsivity 1 for neglect
halfpower = 30;   %LED halfpower angle
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Nx = x*10; %x grid dimension
Ny = y*10; %y grid dimension
X = -x/2:x/Nx:x/2;  %X matrix
Y = -y/2:y/Ny:y/2;  %Y matrix
[rx,ry]=meshgrid(X,Y); %matlab command to generating 2D grid
d= sqrt((rx-lx(1,1)).^2+(ry-ly(1,1)).^2+(rz-lz)^2);
%led photodiode distance matrix
costheta = sqrt((lz-rz)^2)./d;
%angle between led normal and d
coscsi = sqrt((lz-rz)^2)./d;
%angle beetween receiver normal and d
if ( csilimit >= acos(coscsi))             %concentrator factor
gcsi = n^2/(sin(csilimit))^2;             %optical concentrator function
else %if csilimit is excedeed the signal doesnt pass throught the channel
gcsi = 0;
end
m= -log(2)/log(cosd(halfpower));
%Lambertian mode number (cosd mean cosdegree)
```

```
aeff=(ar*coscsi);
%effective area for ray inclination respect receiver surface
H0= ((m+1)*gcsi*tsfilter*resp*aeff.*((costheta).^m))./(2*pi.*d.^2);
%equation for VLC channel LOS H0
H1 = meanInputPower*H0;
%Los*inputPower
surf(rx,ry,H1);
%Plotting surface
xlabel({'Room length','(meters)'});
ylabel ({'Room width','(meters)'});
zlabel ({'Luminous power','(Watt)'});
xlim([-x/2,x/2])
ylim([-y/2,y/2])
title({'Received Power';['Input Power = ',num2str(meanInputPower),' W']});
```

Equation A.1 LOS channel Matlab simulation.

A.2 OOK

```
%number of bits to generate
nSignal = 1000; %Signal lenght
ele = 1.6e-19; %Charge of Electron
Iback = 202e-6; %Background Noise Current
%N0 = 2*ele*Iback; %Noise Spectral Density, 2*electron*backgcurrent
N0 = 1;
Resp = 1; %Receiver responsivity
bitRate = 10^6 %BitRate
Tbit = 1/bitRate; %one bitTime
SNR_db = 1:16; %db SNR
SNR = 10.^(SNR_db./10); % linear SNR
randombinary = rand (1,nSignal)> 0.5;    % Random Binary Signal
randombinary = randombinary *1; %transform logical input in double
for i=1:length(SNR_db) %SNR_db cycle
    Pavg(i) = sqrt((N0*bitRate*SNR(i))/(2*Resp^2)); %Luminous power
    Ipeak(i) = 2*Resp*Pavg(i); %Photodiode Current
    Epeak(i) = Ipeak(i)^2 * Tbit; %Peak current energy
    sigma(i)=sqrt(N0*Epeak(i)/2); %standard deviation after receiver
    threshold=0.5*Epeak(i); %threshold level
    for j=1 : nSignal;
        % n = normrnd(0,sigma(i));
        receivedSignal(j) = randombinary(j)*Epeak(i)+ normrnd(0,sigma(i));
%matched filter output
        % bitsignal * Energy for one bit + normal distribuited noise
    end
   % receivedSignal = awgn(randombinary*Epeak(i),SNR_db(i)+3,'measured');
   % same of above cycle
    Rx = zeros(1,nSignal); %received signal inizialization
    Rx(find(receivedSignal>threshold)) = 1; %threshold detection
    [No_of_Error(i) simuBER(i)]=biterr(randombinary,Rx); %matlab function
end
theorBER = qfunc(sqrt(SNR)); %theorical formula of OOK BER
semilogy(SNR_db,theorBER,'red'); %theoretical BER graph
grid on
ylabel('BER'); xlabel('SNR (dB)');title('Bit Error Rate for Binary ');
%graph definition
hold on
semilogy(SNR_db,simuBER,'blue'); %simulation BER graph
```

Equation A.2 Matlab code to simulate BERook(SNR_dB) and compare with theoretical curve.

A.3 PPM

```
function PPM=generate_PPM(M,nsym)
% function to generate PPM
% 'M' bit order
% 'nsym': number of PPM symbol to generate
PPM=[]; %PPM array empty inizialization
for i= 1:nsym %cycle from 1 to number of symbol,every cycle generate one
symbol
    bitSig= rand (1,M)> 0.5; % random binary number
    dec_value=bi2de(bitSig,'left-msb');    %converting bit to decimal value
    tempPPM=zeros(1,2^M); %zero sequence of length 2^M
    tempPPM(dec_value+1)=1; %placing a pulse accoring to decimal value,
    %matlab index start from 1 and not from 0, so need to add 1;
    PPM=[PPM tempPPM]; %put tempPPM in array queue
    %PPM symbol
end %close for cycle
end
```

Equation A.3.1: Matlab code example, transform random binary signal to PPM signal

```
%%%%%%%%%%%%%%%%%%%%%%
%%% this program calculate SLER of PPM
%%%%%%%%%%%%%%%%%%%%%%
clear all;

M=3;     %bit order
N0=1;    %noise
L=2^M;   %symbol length
nsym=200; %number of PPM symbols
Lsig=nsym*L; %total length of PPM slots
R=1;         %photoreceiver responsivity
Rsymb=1e6; %slot rate symbol rat
Rb = (Rsymb*M); %bitRate
Tb=1/Rb;
SNR_db= -5:0.5:16; % Energy per bit db
EsN0=SNR_db+10*log10(M); % Energy per symbol db
SNR=10.^(SNR_db./10); %Energy per bit Eb/N0
PPM= generate_PPM(M,nsym); %function to generate PPM signal
PPM = PPM*1; %Matlab logic signal in double
for i=1:length(SNR_db)
    Pavg(i) =(1/L)*sqrt((((2*M)*N0*Rsymb*SNR(i))/(2*R^2)));
     %Luminous power factor (2M/L)
    Ipeak(i) = L*R*Pavg(i); %Photodiode Current
    Epeak(i) = L*M*Ipeak(i)^2 * Tb; %Peak current energy
    sigma(i)=sqrt(N0*Epeak(i)/(2)); %standard deviation after receiver
    threshold=0.5*Epeak(i); %threshold level
    for j=1 : Lsig;
        % n = normrnd(0,sigma(i));
         MF_out(j) = PPM(j)*Epeak(i)+ normrnd(0,sigma(i)); %matched filter
output
         % bitsignal * Energy for one bit + normal distribuited noise
    end
    received_PPM=zeros(1,Lsig); %generating empty PPM vector
    received_PPM(find(MF_out> threshold))=1;   %generating the received
signal
[No_of_Error(i) ser_hdd(i)]= biterr(received_PPM,PPM);
%Matlab function to caluclate the SER
end
semilogy(SNR_db,ser_hdd,'magenta'); %simulation BER graph
ylabel('SLER'); xlabel('SNR (dB)');
```

```
title([num2str(L),'-PPM SlotErrorRate']);
grid on
grid minor
hold on;
% theoretical calculation
Pse_ppm_theor=qfunc(sqrt(M*2^M*0.5*SNR)); %transform SLER to SER
semilogy(SNR_db,Pse_ppm_theor,'green','linewidth',0.1); %theoretical BER
graph
```

Equation A.3.2 Matlab code to simulate the L-PPM modulation

Appendix B BER graphics

B.1 Theoretical vs Simulated PPM

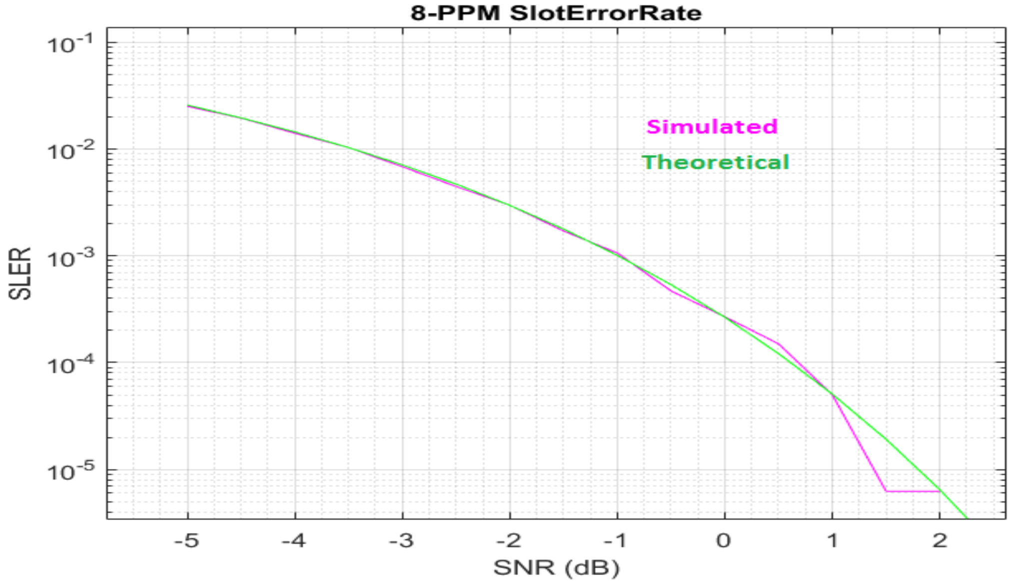


Figure B.1.1: 8-PPM BER simulation with random generate signal of 160E3 slots. Green is theoretical function and magenta is the simulated function.

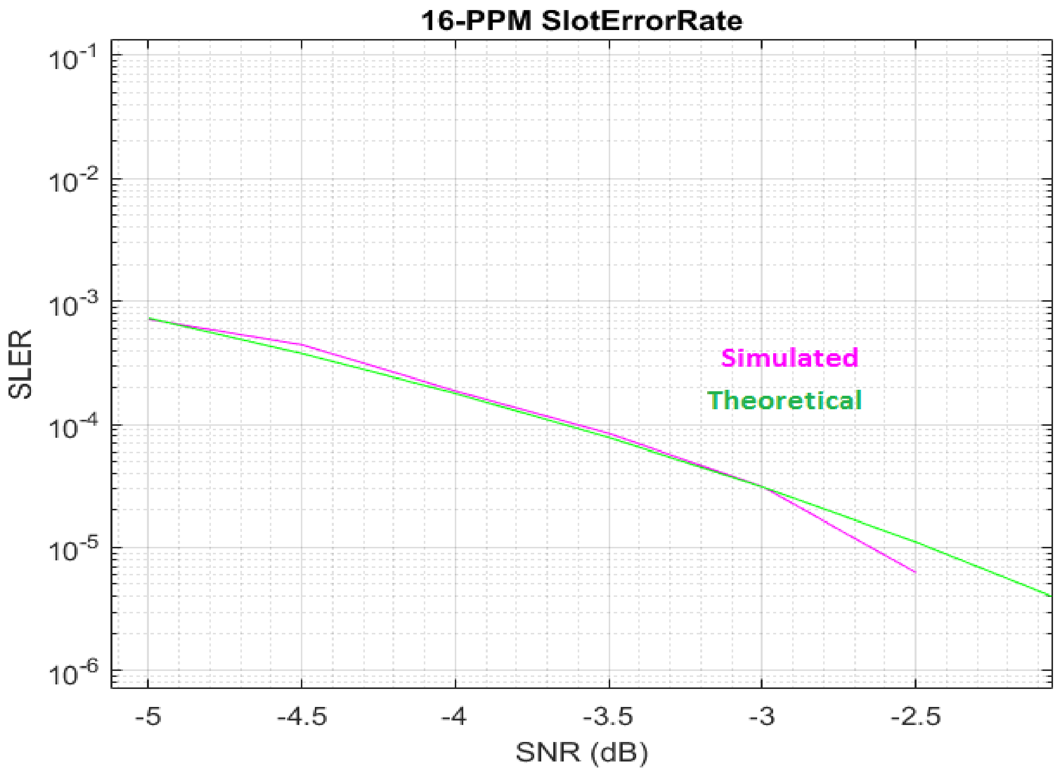


Figure B.1.2: 16-PPM BER simulation with random generate signal of 320E3 slots. Green is theoretical function and magenta is the simulated function.

B.2 Theoretical PPM vs OOK

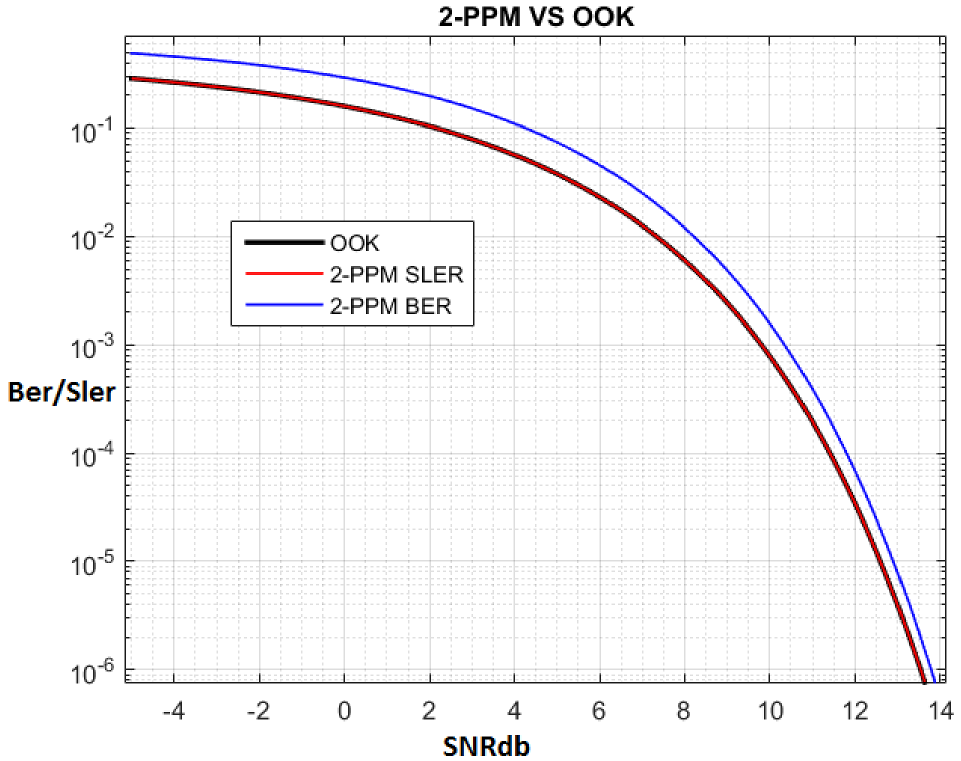


Figure B.2.1.: BER/SLER of OOK, 2-PPM comparison.

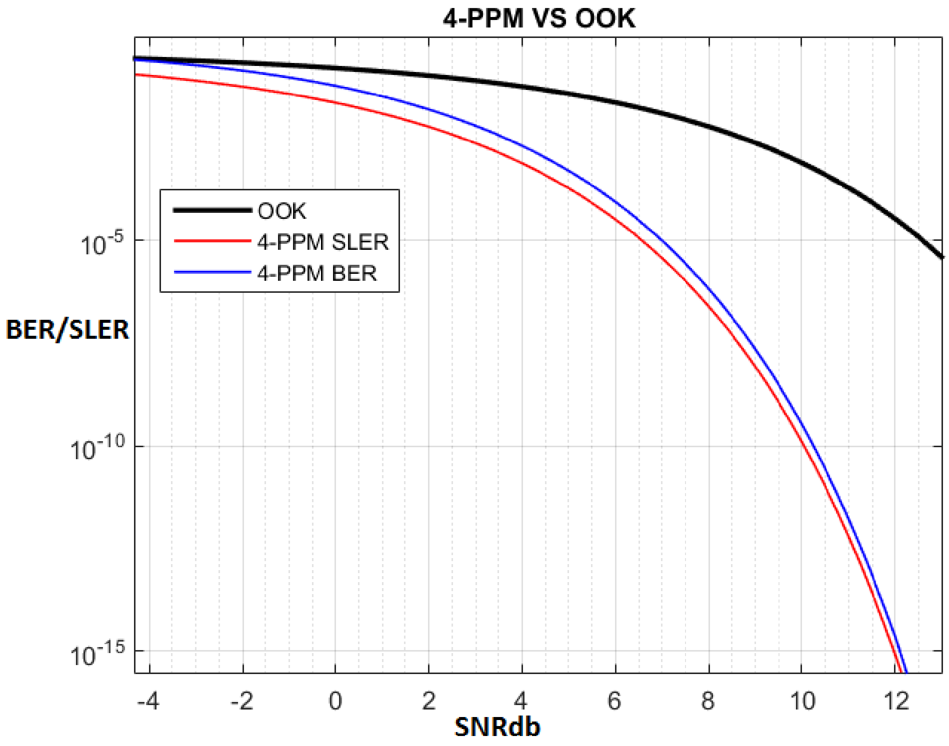


Figure B.2.2: BER/SLER of OOK, 4-PPM comparison. X-axis is SNRdb, Y-axis is BER/SLER.

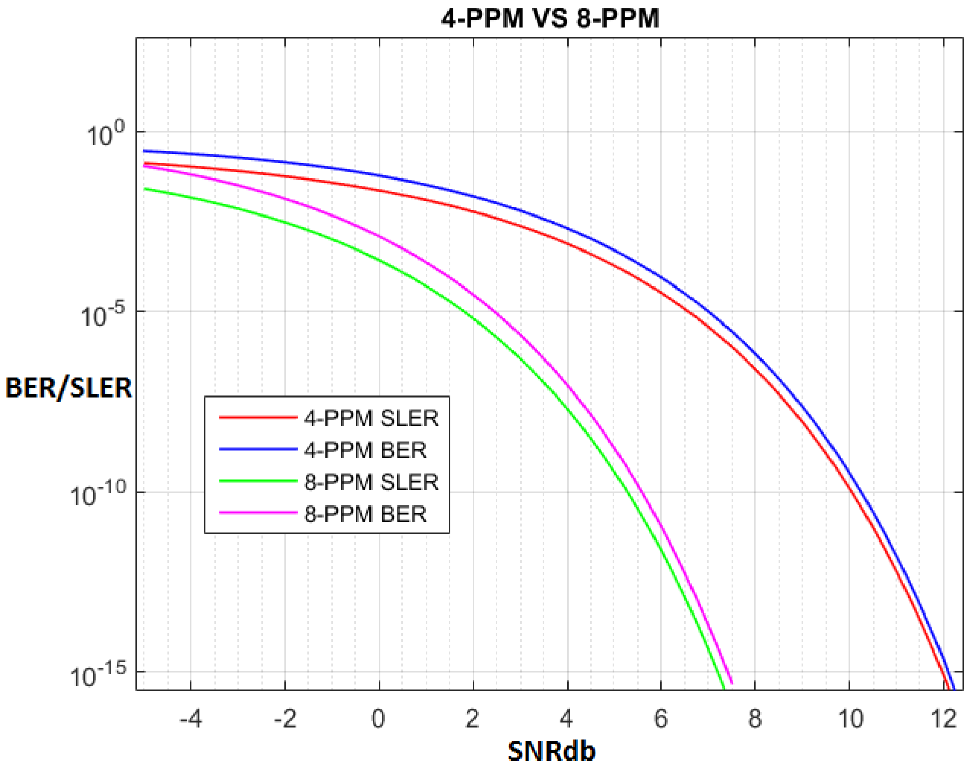


Figure B.2.3: BER/SLER of 4-PPM, 8-PPM comparison. X-axis is SNRdb, Y-axis is BER/SLER.

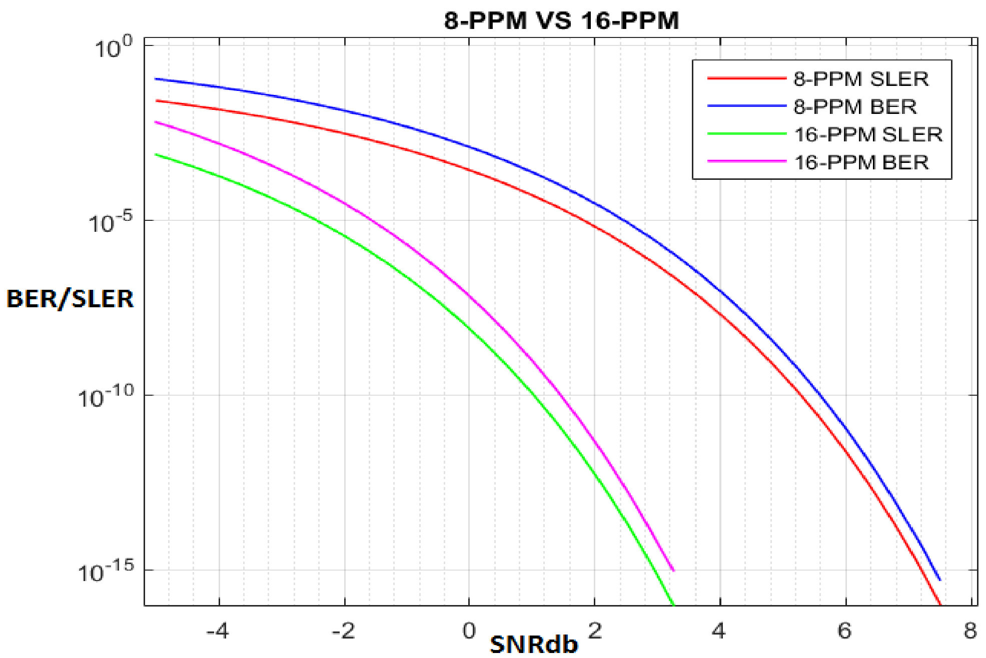


Figure B.2.4: BER/SLER of 8-PPM, 16-PPM comparison. X-axis is SNRdb, Y-axis is BER/SLER.

Appendix C Arduino code

C.1 VLC calculating threshold transmission/receiver

C.1.1: Transmitter code:

```
int ledPin = 9;               //definition of transmission LED pin
int ledPin2 =10;              //definition of synchronization LED pin

void setup() {
Serial.begin(9600);
pinMode(ledPin,OUTPUT);       //LED transmission pin as output
pinMode(ledPin2,OUTPUT);      //LED transmission pin as output
  digitalWrite(ledPin,HIGH);  //LED transmission on (this value can be changed to calculates other threshold conditions)
  digitalWrite(ledPin2,HIGH); //LED transmission on
}
void loop()
{
 }
```

C.1.2: Receiver code:

```
int analogValue[100];   //vector to calculate photoreceiver reception mean
int analogValue2[100]; //vector to calculate photoreceiver synchronization mean
int photoPin = A0;            //defining pin of reception pd
int photoPin2 = A1;     //defining pin of synchronization pd
int delayTime = 10;     //delaytime between reception
int meanValue2 = 0;     //mean of synchronization pd
int meanValue = 0;            //mean of reception pd

void setup() {
  Serial.begin(9600);
  pinMode(photoPin,INPUT);          //pin of pd reception defining input
  pinMode(photoPin2,INPUT);   //pin of pd synchronization defining input
}

void loop()
{
  for(int i = 0;i<100;i++)
  {
     analogValue[i] = analogRead(photoPin);       //cycle to read 100 values of pd reception
     delay (delayTime);
  }
  for(int i = 0;i<100;i++)
  {
     analogValue2[i] = analogRead(photoPin2);     //cycle to read 100 values of pd synchronization
     delay (delayTime);
  }
  for ( int i = 0 ; i<100 ; i++ )
    {
     meanValue = meanValue+analogValue[i];        //cycle to calculates sum of reception
     meanValue2 = meanValue2+analogValue2[i]; // and synchronization array
    }
    meanValue =  meanValue/100;          //mean
    meanValue2= meanValue2/100;          //mean

    Serial.print("LED1: ");
    Serial.print(meanValue);      //print meanValue
    Serial.print(" - LED2: ");
    Serial.print(meanValue2);     //print meanValue2
    Serial.print("\n");

   meanValue = 0;                 //resetting mean
   meanValue2 = 0;                //resetting mean
}
```

C.2: First bits transmitter/receiver

C.2.1 Transmitter code:

```
int ledPin = 9;                    //Led pin
int delayTime = 100;               //delay time
char value;                        //handy value to insert one bit
String values;                     //bit string to transmit
int length;                        //handy value to memorize the length of the transmission string
boolean shouldSend;                //if this is true the transmission start

void setup() {
pinMode(ledPin, OUTPUT);                //define the LED pin as output
Serial.begin(9600);
}

void loop()
{
 while (Serial.available())             //if the string is available start the while cycle
    {                                   //here the user should insert a binary string
     value = Serial.read();             //put in value one char from serial monitor
     // Build string
     values = values + String(value);   //create a string composed by each char values

     // Loop
     delay(1);                          //wait 1ms
    }
  if (values != "")                     //if the values string isn't empty
    {
     Serial.println(values);            //print the bits string
     length = values.length();          //acquire the length of the string
     shouldSend = true;                 //Set the sent value true
     }

  if (shouldSend)                       //if shouldSend true
    {                                   //start LED ON-OFF cycle
     for (int i = 0; i < length; i++)   //cycle the String character
       {
        if (values.charAt(i) == '0')    //if the character is 0
          {
           digitalWrite(ledPin, LOW);   //turn OFF led
           delay(delayTime);            //wait delaytime
           }
       else if (values.charAt(i) == '1')  //if character is 1
          {
          digitalWrite(ledPin, HIGH);   //turn ON led
          delay(delayTime);             //wait delaytime
           }
         }
    }
    values = "";                   //after the string transmission reset the string
    shouldSend = false;            //set the shouldSend false, in that way the program wait until other bits string is ready
   }
 }
```

C.2.2 Receiver code:

```
int photoPin = A0;       // photodiode receiver pin
int delayTime = 100;     // delay time in ms
int analogValue;         //handy variable to memorize the receiver value
int lowThreshold = 60; //threshold level calculated in C.1
char value;              //handy value to memorize on bit
String values;           //handy value to memorize bits

void setup() {
 Serial.begin(9600);
 pinMode(photoPin, INPUT);    //define photopin as input
}

void loop()
{
   analogValue = analogRead(photoPin);             //acquire analog photoPin value
if (analogValue < lowThreshold)                    // if photoPin value < lowthreshold set 0
   {      value = '0';     }
```

```
else                                            //else set 1
    {
    value = '1';
   }
  values = String(value);                       //save binary value in values string
  if (values != "")                             //if the string values is not empty
  {                                             // if there is transmission data
 Serial.print(values);                          // print on serial
 Serial.print("\n");
  }
  delay(delayTime);                             //attend delay Time after every acquisition
}
```

C.3: VLC ASCII transmission/receiver

C.3.1: Transmitter code:

```
int ledPin = 9; //Transmitter LED pin
int ledPin2 =10;          //synchronization LED pin
int valuecicle; // handy variable to convert decimal to bit
int bin[8];               // handy vector to convert decimal to bit
int delayTime = 10;       //delay time between bit (symbol)
char value;     // input value from console
char valuesArray[13];            //char array from serial console
int asciiArray[13];              //asciiArray for conversion
int binMatrix[13][8];            // where the binary value is store
String values;            //where put the input from serial console
boolean shouldSend;              // to check the state of led2 for coordinate transmission

void setup() {
Serial.begin(9600);
pinMode(ledPin,OUTPUT);     //pin defining
pinMode(ledPin2,OUTPUT);    //pin defining
}
void loop()
{
        while (Serial.available())  //if serial console is available enter the cycle
        {
         value = Serial.read();               //read serial console
           values = values + char(value);     //create string from console value
            delay(1);
        }
        if (values != "")                     //if string values is not empty enter here
         {
          values.toCharArray(valuesArray,12+1);    //convert values to char array and put in values array +1 is to count the \0 at the end
             for (int i = 0 ;i <12; i++)                //star cycle to convert from char to ascii and to number
                  {    asciiArray[i] = (int)valuesArray[i];        //transform the char values in int ascii decimal number
                                  }
           for (int j= 0; j<12;j++)  //cycle to convert the decimal ascii number in binary
         {
                              valuecicle=asciiArray[j];
                               for(int i =0; i<7; i++)
                  {
                              if(valuecicle%2==1)        //decimal to binary conversion
                    bin[i]=1;
                  else
                        bin[i] =0;
                         valuecicle=valuecicle/2;
                  }
                        for(int i =0; i<7; i++)  //the cycle before convert the binary number but is write from right to left
                                      {
                                       binMatrix[j][i]= bin[7-1-i];           //conversion from left to right
                               }
                         }
        for (int j= 0; j<12;j++) {   //cycle to print the results on monitor
         for (int i= 0; i<7;i++)      //these two nested cycles are only to debug
                              {
                               Serial.print(binMatrix[j][i]);
                              }
                   Serial.print("\n");
                  }
                shouldSend = true;
                  Serial.print("\n");
```

```
        }          //end of the serial print
      if (shouldSend)      //if shouldSend is true transmit the binmatrix
          {
        digitalWrite(ledPin2, HIGH);         //set led2 to high value to communicate the start of the transmission

        for (int j= 0; j<12;j++)                    //start the transmission of led1
            {
                              for (int i= 0; i<7;i++)
                                        {
             if(binMatrix[j][i]==0)
                                        {
                                             digitalWrite(ledPin, LOW);     //transmit low with led 1
                                            delay(delayTime);                     // attend delay time
                                        }
             else{
                                        digitalWrite(ledPin, HIGH);          //transmit high from led 1
                                       delay(delayTime);                        // attend delay time
                                         digitalWrite(ledPin, LOW);          // set low after delay time (from this to following blink the time is
neglected)
                                    }
                                         }
          }
           for (int i =0;i<12;i++)    //resetting value
   {
       valuesArray[i]=0;
       asciiArray[i]=0;
   }
      for (int j= 0; j<12;j++)
       {
           for (int i= 0; i<7;i++)
            {
             binMatrix[j][i]=0;       //resetting the value
            }
   }
  }
         values="";                       //resetting message
  digitalWrite(ledPin2, LOW);       //after the message the synchronization led turned off
shouldSend = false;                   //resetting the message status
 }
```

C.3.2: Receiver code:

```
int photoPin = A0;        //reception photo-receiver variable
int photoPin2 = A5;      //synchronization photo-receiver2 variable
int delayTime = 10;      /delay time between reading of photo-receiver reception warning this must be equal to LED delaytime
int analogValue;           //handle variable to read photoPin
int analogValue2;         //handle variable to read photoPin2
int lowThreshold = 56; //photoPin transmission threshold
int highThreshold = 57;//photoPin reception threshold
int binMatrix[13][8];                  // received binary matrix
int data[13];                 // where put the decimal converted matrix
char charData[13];                     // to print on serial monitor the decimal value
boolean shouldConvert;             //handle variable to enable the conversion cycle

void setup() {
  Serial.begin(9600);     //console begin
  pinMode(photoPin,INPUT);     //defining photo-pin as input
  pinMode(photoPin2,INPUT);   //defining photo-pin as input
}

void loop()
{   analogValue2 = analogRead(photoPin2); // read synchronization photo-pin2
    while (analogValue2 > highThreshold) {           // if the led2 is on start the reception
      for (int j=0;j<12;j++)             //cycle to reset data and charData vector
      {
        data[j]=0;
        charData[j]="";
      }               //cycle end
        for (int j= 0; j<12;j++) //cycle to read the LED1 on PhotoPin reception
         {            for(int i =0; i<7; i++)
           {   analogValue = analogRead(photoPin); //read reception pd
               if (analogValue < lowThreshold)                      //generating matrix with 0 or 1
               {    binMatrix[j][i]=0;   }                               // if the threshold is under lowThreshold 0
```

```
            else
                {    binMatrix[j][i]=1;   }                    //else reception is 1
            }
                delay(delayTime);                  //after one reception one should attend delaytime
        }
      shouldConvert = true;          // to start the conversion in cycle after
       analogValue2 = analogRead(photoPin2);  //overwrite the old synchronization value to recheck that
  }
  if(shouldConvert)       //if shouldConvert == true
   {
     for (int j= 0; j<12;j++)
        {                                          //cycles to convert the binary received matrix in char
         for(int i =0; i<7; i++)
           {
                Serial.print(binMatrix[j][i]);   //print of binary matrix      0.1 value after is write only for rounding Arduino problem
                    data[j]= (binMatrix[j][i])*((int)(0.1+pow(2,7-1-i)))+data[j];    //every cycles convert one significant bits
            }
          charData[j] = char(data[j]);           //print the decimal number on serial monitor
           Serial.print(" - ");
           Serial.print(charData[j]);
            Serial.print("\n");
         }
           Serial.print("\n");
   }

  shouldConvert =false;              //resetting the conversion value
}
```

C.4 VLC Automatic transmitter/receiver to test the BER

C.4.1: Transmitter code:

```
int ledPin = 9;              //Transmitter LED pin
int ledPin2 =10;             //synchronization LED pin
int valuecicle;              // handy variable to convert decimal to bit
int bin[8];                  // handy vector to convert decimal to bit
int delayTime = 20;          //delay time between bit (symbol)
byte valuea=97;              //default value:"a"
byte valueb=98;              // default value:"b"
byte valuec=99;              // default value:"c"
byte valued=100;             // default value:"d"
byte  valuee=101;            // default value:"e"
byte valuef=102;             // default value:"f"
byte  valueg=103;            // default value:"g"
byte valueh=104;             // default value:"h"
String values;               //default string
char valuesArray[9];         //handy value to convert string to char array
int asciiArray[9];           //handy value to convert string to ascii number
bool binMatrix[9][8];        //final transmission bits matrix

void setup() {
        Serial.begin(9600);
        pinMode(ledPin,OUTPUT);           //set ledPin as output
        pinMode(ledPin2,OUTPUT);          //set ledPin2 as output
        delay(5000);                      //delay time before start the auto LED switching
        }

void loop()
{    delay(50);                            //delay between two strings
       values="";            //reset the string value
       values =values+char(valuea)+char(valueb)+char(valuec)+char(valued)+char(valuee)+char(valuef)+char(valueg)+char(valueh);
        //creation of the defaults string
 if (values != "")           //if the values string is not empty start the cycle
     {
       values.toCharArray(valuesArray,8+1);          //convert string to char array
       for (int i = 0 ;i <8; i++)                    //convert the char array in the
                              {                      //ASCII corresponding decimal value
                  asciiArray[i] = (int)valuesArray[i];
                       }
       for (int j= 0; j<8;j++)
        {                                          //convert the decimal ASCII
```

```
        valuecicle=asciiArray[j];                //in a binary vector
          for(int i =0; i<7; i++)                //the binary vector is inverted
        {
         if(valuecicle%2==1)
           bin[i]=1;
         else
         bin[i] =0;

         valuecicle=valuecicle/2;
        }

         for(int i =0; i<7; i++)
           {
              binMatrix[j][i]= bin[7-1-i];       //turn the vector in the right way and put that in a matrix
              }
         }
        for (int j= 0; j<8;j++)
         {
                for (int i= 0; i<7;i++)
                      {
                      Serial.print(binMatrix[j][i]);          //print the converted matrix for debugging
                      }
                      Serial.print(";");
                      }
          Serial.print("\n");
    }
    digitalWrite(ledPin2, HIGH);             //turn ON the synchronization LED
  for (int j= 0; j<8;j++)
       {
          for (int i= 0; i<7;i++)
           {
            if(binMatrix[j][i]==0)                          //Cycle the matrix
              {
                digitalWrite(ledPin, LOW);
                delay(delayTime);
              }                                             //If the matrix value is 0 turn OFF the transmission LED and wait delaytime
            else
              {
                digitalWrite(ledPin, HIGH);
                delay(delayTime);                           //if the matrix value is 1 turn ON the transmission LED and wait delaytime
                digitalWrite(ledPin, LOW);                  //after delay time turn off the LED (this is necessary particularly for the last cycle)
              }                                             // otherwise the LED remain ON
            }
         }
          for (int i =0;i<8;i++)
   {
        valuesArray[i]=0;                        //RESETTING valuesArray, asciiArray and binMatrix
        asciiArray[i]=0;
   }
        for (int j= 0; j<8;j++)
        {
          for (int i= 0; i<7;i++)
           {
            binMatrix[j][i]=0;
           }
    }
  digitalWrite(ledPin2, LOW);                //Turn off the synchronization LED
}
```

C.4.2: Receiver code:

```
int photoPin = A0;                    //define transmission receiver pin
int photoPin2 = A5;                   //define synchronization receiver pin
int delayTime = 20;                   //delay time between symbol
float analogValue;                    //handy variable to put transmitter received value
float analogValue2;                   //handy variable to put synchronization received value
float lowThreshold =35.90;            //led trasmettitor threshold
float highThreshold = 27;              //led sincronization threshold
bool binMatrix[9][8];                 //acquired signal
void setup() {
  Serial.begin(9600);
  pinMode(photoPin,INPUT);    //set pin of the photodiode receiver as input
  pinMode(photoPin2,INPUT);  //set pin of the photodiode synchronizer input
}
```

```
void loop()
{
  analogValue2 = analogRead(photoPin2);            //read the photoPin2
  while (analogValue2 > highThreshold) {    //if photoPin2 (synchronization) is ON (pinvalue>highThreshold) start the receiving cycle
for (int i = 0 ;i<8 ; i++)                         //the transmitte signal is an 8*7 matrix
    {                                              //nested cycle the matrix
for (int j = 0 ; j < 7 ; j++)
    {
        binMatrix[i][j]=0;                         //reset matrix
    }
}
        for (int j= 0; j<8;j++)                    //received matrix creation
         {
             for(int i =0; i<7; i++)                       //cycle matrix with nested cycle
            {
                analogValue = analogRead(photoPin);        //read the transmission photodiode
                if (analogValue < lowThreshold)            //if the received value is < then lowthreshold
                {
              binMatrix[j][i]=0;                           //acquire 0
                }
             else              {
                 binMatrix[j][i]=1;                        //else 1
                  }
               delay(delayTime);                   //wait delaytime
            }
           }
      for (int j= 0; j<8;j++)
         {
           for(int i =0; i<7; i++)
            {
                Serial.print(binMatrix[j][i]);     //print the matrix on serial
            }
        Serial.print(";");
             }
           Serial.print("\n");
        analogValue2 = analogRead(photoPin2);      //read the synchronization photodiode for checking if the transmission is on
  }                                                //this is done in the next cycle
}
```

References

[1] Cui, Kaiyun. Physical Layer Characteristics and Techniques for Visible Light
[2] Wikipedia. “Flicker fusion” https://en.wikipedia.org/wiki/Flicker_fusion_threshold
Communications. UC riverside electronic theses and dissertations. 2012.
[3] Wikipedia. “OSI model”. https://en.wikipedia.org/wiki/OSI_model
[4] Shridhar Ambady, Megan Bredes, Calvin Nguyen. Visible Light Communication, A Major Qualifying Project Report, completed in partial fulfillment of the requirements for the degree of Bachelor of Science at Worcester Polytechnic Institute
[5] Wikipedia. Poisson distribution. https://en.wikipedia.org/wiki/Poisson_distribution
[6] Z. Ghassemlooy, W. Popoola, S. Rajbhandari. Optical Wireless Communications System and Channel Modelling with MATLAB.
[7] Wikipedia “Lambert’s cosine law” https://en.wikipedia.org/wiki/Lambert's_cosine_law
[8] Wikipedia “Specular reflection” https://en.wikipedia.org/wiki/Specular_reflection
[9] Farshad Miramirkhani, Murat Uysal. Channel modelling and characterization for visible light communication. Department of Electrical and Electronics Engineering, Ozyegin University, Istanbul, Turkey.
[10] John R. Barry, Joseph M. Kahn, William J. Krause, Edward A. Lee, and David G. Messerschmitt. Simulation of Multipath Impulse Response for Indoor Wireless Optical Channels
[11] Kwonhyung Lee, Hyuncheol Park and John R. Barry. Indoor Channel Characteristics for Visible Light Communications. IEEE COMMUNICATIONS LETTERS, ACCEPTED FOR PUBLICATION.
[12] Joseph M. Kahn, John R. Barry. Wireless infrared communications. IEEE 1997.
[13] Wikipedia, "Matched filter", https://en.wikipedia.org/wiki/Matched_filter
[14] John R. Barry, Edward A. Lee, David G. Messerschmitt. Digital communication third edition, volume 1. Springer science + Business Media, LLC.
[15] Pol Mesalles Ripoll supervised by Dr. Juan Carlos Escalera. Short-Range Optical Wireless Communications.
[16] Visible light communication: Tutorial. Project: IEEE P802.15 Working Group for Wireless Personal Area Networks (WPANs). University of Oxford.
[17] Nupur Tyagi, Amit Jain. Shifting Paradigms in IT: An Overview on VLC & BLE. Delhi University
[18] Web “RONJA” http://ronja.twibright.com/

Ringraziamenti:

Desidero dedicare un piccolo spazio a tutte le persone che mi hanno permesso di realizzare questo lavoro di tesi, sia per i consigli e suggerimenti tecnici sia solo per il supporto morale che mi hanno dato in questo anno di lavoro.

Ringrazio cortesemente e vivamente il professor Stefano Argirò, relatore, che mi ha aiutato moltissimo nella stesura della tesi ed ha permesso la realizzazione del capitolo riguardante Arduino.
Ringrazio il Co-relatore e tutor aziendale Antonio Orlando per la disponibilità e l’aiuto durante lo svolgimento della tesi.
Ringrazio la mia controrelatrice la professoressa Michela Greco per la sua disponibilità e cortesia.
Ringrazio il Telecom Italia Lab (TILAB) e tutte le persone al suo interno che mi hanno sostenuto ed affiancato tra cui in particolare Bruno Melis.

Ringrazio moltissimo i miei genitori Cristina e Gianni che hanno sempre creduto in me e mi hanno permesso di continuare il percorso di studi per lungo tempo.

Ringrazio Neda per avermi sostenuto, incoraggiato e sopportato nella stesura della metatesi e nell’ ultima fase della scrittura della tesi, Lorenzo Bellizzi per la compagnia e le decine di caffè offerti, Stefano Bocchio e Dario Cottafava per avermi prestato i due Arduino utilizzati nello svolgimento del capitolo 4, i miei coinquilini e tutti gli amici che mi hanno sempre motivato e supportato in questo anno di lavoro.

Senza tutte queste persone questo lavoro non sarebbe stato possibile.

Un grazie di cuore a tutti