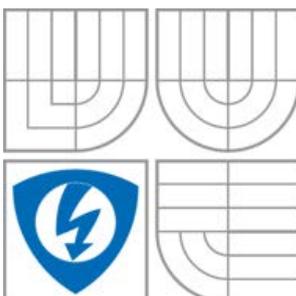


BRNO UNIVERSITY OF TECHNOLOGY



FACULTY OF ELECTRICAL ENGINEERING AND
COMMUNICATIONS
DEPARTMENT OF RADIO ELECTRONICS

MODEL FOR INDOOR WIRELESS OPTICAL LINK

MASTER'S THESIS

AUTHOR

Bc. RUBÉN BELLO CAMPOS

SUPERVISOR

Prof. OTAKAR WILFERT

BRNO 2012

**ESCUELA TÉCNICA SUPERIOR DE INGENIEROS
INDUSTRIALES Y DE TELECOMUNICACIÓN**

UNIVERSIDAD DE CANTABRIA



Proyecto Fin de Carrera

**MODEL FOR INDOOR WIRELESS OPTICAL
LINK**

Para acceder al Título de

INGENIERO DE TELECOMUNICACIÓN

Autor: Rubén Bello Campos

BRNO 2012

Brno, the 6th of June 2012

Prof. Otakar Wilfert
Supervisor



**FACULTY OF ELECTRICAL ENGINEERING AND
COMMUNICATIONS**

DEPARTMENT OF RADIO ELECTRONICS

MASTHER'S THESIS

MODEL FOR INDOOR WIRELESS OPTICAL LINK

Author: Rubén Bello Campos

Supervisor: Prof. Otakar Wilfert, CSc. <wilfert@feec.vutbr.cz>

An indoor wireless optical link works in IR spectral region and, therefore, any interference with other links are not presented. An interesting application of the indoor wireless optical links is for communications between PC and printer, communications between sensors within the electronic security alarm systems, etc. The transmitted power passes through the atmosphere and is reflected on the wall (reflecting surface). The incident optical power on the walls or barriers is subject to the laws of radiation and matter interaction. Reflection on the surface can have dispersive (diffused) or specular (reflective) character. The aim of this project is to create a model for the link design where power balance equation and the power level diagram are used.

[1] SANTAMARIA, A., LÓPEZ-HERNÁNDEZ, F. J. Wireless LAN Systems. London: Artech House, 1994.

[2] FRANZ, J. H., JAIN, V. K. Optical Communications Components and Systems. London: CRC Press, 2000.

Abstract

In the last few years, there has been a growing interest in optical wireless communications for indoor and outdoor applications. Infrared technology is the best option for indoor environments due to it is immune to radio interferences, the spectrum is freely available and infrared components are inexpensive, small and consume little power.

This thesis deals with the structure of indoor optical wireless communications and with their possible applications. Optical transmitters, optical receivers and lenses are the basic elements of optical links and their characteristics and parameters are described.

The beam in an indoor optical wireless link reflects on walls and various objects. This thesis presents a power budget of the link and includes the reflectivity of the surfaces where the beam is reflected.

Keywords

Indoor wireless optical link, surface reflectivity, infrared beam, optical transmitter, optical receiver, optical lenses, atmospheric phenomena, power link budget.

Declaration

I hereby confirm that I worked out this thesis, entitled “Model of Indoor Wireless Optical Communications”, myself with aid of my project leaders, bibliography, documentation and other sources of information that are mentioned in the end of thesis.

Brno, the 6th of June 2012

Rubén Bello Campos

Acknowledgements

This thesis would not have been possible without the help and support of the following people.

Firstly, I would like to thank and express my appreciation to my supervisor Prof. Otakar Wilfert from Czech Republic, for his help and attention in all related with my project. Without his comments, ideas and proposals, this thesis would not have been possible.

Secondly, I want to thank my parents because they trusted me from the beginning and they gave me everything I needed. Also to my brother, who has always been there helping and motivating me to finish this career. I would like to thank my grandmothers, my uncles and my cousins to make me happy, but especially my grandfather because despite his advanced age, his dream was to see me finish the career, and we have achieved.

I want to express my gratitude to all my classmates, without all of you this career would be much more difficult and bored, thank you for your help and company all these years. Also to all the teachers I have had and have helped me to be better.

Finally, I would like to thank to all my friends who have been there all these years and especially to my Erasmus friends because they have made this year the greatest experience I ever had.

Thank you to all of you.

Content

1. Introduction	12
2. FSO transceiver – structure and function.....	13
2.1. Transmitter	13
2.2. Receiver.....	17
2.3. Optics	20
2.3.1. Types of lenses	21
2.3.2. Lensmaker’s equation	25
2.3.3. Aberrations	27
2.4. The indoor FSO link – definition and operation	30
2.4.1. Radio Frequency	30
2.4.2. Infrared	31
2.5. Indoor FSO links – Classification and requirements	34
2.5.1. Classification	34
2.5.2. Requirements	35
2.6. Indoor FSO links – Limits.....	38
2.6.1. Bit Rate	38
2.6.2. Bit Error Rate (BER).....	38
2.6.3. Range	39
2.6.4. Power of the LED	39
2.6.5. Noise Equivalent Power (NEP).....	39
2.6.6. Receiver sensitivity	39
3. Atmospheric phenomena.....	40
3.1. Attenuation	40
3.1.1. Gas absorption.....	41
3.1.2. Molecular scattering.....	42
3.1.3. Aerosol extinction.....	42
3.2. Turbulence	43
3.3. Ambient light	43

4. The surface affects on the link function.....	45
4.1. Dispersive (diffused)	45
4.2. Specular (reflective).....	46
4.3. Relative directional reflectivity	47
4.4. Applications	48
5. Steady model of the indoor wireless optical link	50
5.1. Power budget.....	50
5.2. Link margin.....	51
5.3. Matlab design	53
5.4. Examples	55
5.4.1. Black and white ceiling	55
5.4.2. Modifications of a black and white ceiling.....	56
6. Conclusion.....	59
7. References	60
8. Appendix A.....	62
9. Appendix B.....	72

List of Figures

Figure 1 - Scheme of the FSO link.....	13
Figure 2 - Laser Diode	14
Figure 3 - Scheme of the connexion between LED and driver	15
Figure 4 - A) LED driver, B) LD driver	16
Figure 5 - Response Time.....	16
Figure 6 – FWHM.....	17
Figure 7 - Silicon Photodiode.....	17
Figure 8 - Silicon PIN Photodiode	18
Figure 9 - Scheme of the connexion between the photodetector and pre-amplifier....	18
Figure 10 - Typical Responsivity	19
Figure 11 - Response time	20
Figure 12 - Types of lenses	21
Figure 13 - Positive lens.....	22
Figure 14 - Negative lens	23
Figure 15 - Negative meniscus lens	23
Figure 16 - Aspherically Contoured Fresnel Lenses.....	24
Figure 17 – Distance bewteen object and real image.....	26
Figure 18 - Distances in optics.....	27
Figure 19 - Spherical aberration	28
Figure 20 - Coma Aberration	29
Figure 21 - Chromatic Aberration.....	29
Figure 22 - Achromatic Doublet	30
Figure 23 - Diffuse link.....	32
Figure 24 - Passive quasidiffuse link.....	33
Figure 25 - Active quasidiffuse link.....	33
Figure 26 - Classification of infrared links	34
Figure 27 - Maximum permissible exposure value vs. wavelength for various intrabeam viewing exposure times.....	37
Figure 28 - Composition of Earth's atmosphere.....	41
Figure 29 - Optical power spectrum of common ambient infrared sources.....	44
Figure 30 - Diffuse vs. specular Reflection	45
Figure 31 - Diffuse surface.....	45
Figure 32 - Example of Specular surface	47
Figure 33 - Application of Specular surface.....	48
Figure 34 - Application on night driving	49
Figure 35 - Example of Specular surface	49
Figure 36 - Diffuse reflection vs. specular reflection.....	51
Figure 37 - Power level diagram of the typical indoor link including the RDR function	52
Figure 38 - Specific parts of the link	52

Figure 39 - Main Program	53
Figure 40 - Reflectivity calculator	54
Figure 41 - Power budget with black ceiling	55
Figure 42 - White ceiling.....	56
Figure 43 - Black ceiling with modified parameters.....	57
Figure 44 - Black ceiling with modified parameters.....	57
Figure 45 - White ceiling modified	58

List of Tables

Table 1 - Safety classification	38
---------------------------------------	----

1. Introduction

Usually, the computer terminals are clustered within office environments and labs. The major problem in all of these environments is the cost of maintaining and reconfiguring wired systems. Such constraints have provided the motivation to look at other means of achieving high-speed wireless connectivity for indoor LAN applications.

Focusing on indoor WLANs, cellular architectures are preferred. The basic idea is to divide the building into cells and establish a wireless link in each cell. The size of each cell depends on several factors: the technology used to implement the system, the environment, the data rate, and so forth.

Infrared is one such alternative, which was first proposed for indoor optical wireless communications in 1979. If we compare infrared with radio frequency, IR systems are not subject to spectral regulations as RF systems are [1]. Another advantage is its inherent channel diversity, which makes multipath propagation fading much less of a problem.

Therefore, it offers a potentially huge bandwidth with is unregulated world-wide, and is capable of supporting the high data rates demanded by multi-media applications. The unregulated spectrum allows manufacturers to design a truly global product without the worries of facing regulations which differ from country to country.

The behaviour of infrared light is similar to visible light. It is absorbed by dark objects, diffusely reflected by light-colour objects and directionally reflected from shiny surfaces. It can penetrate through glass but no thought walls. Therefore, infrared communications are confined to the room in which they are integrated, making them inherently secure against casual attacks.

To make a model of an indoor wireless optical link is needed to know the directional reflective-diffusive characteristics of various surfaces. These characteristics can be measured by means of relative direction reflectivity (RDR) or optical cross section (OCS).

The power budget of the link includes not only the transmitted and received power, but also reflection and propagation losses. The surface chosen includes the diffuse and specular characteristics which make it more or less suitable for these kinds of communications.

2. FSO transceiver – structure and function

In many applications, the use of wiring communications is impossible because of the character of the environment. An economic alternative that works the same as any other means of communications are wireless infrared communications [2]. It works as follow: a baseband analog or digital signal is modulated and codified. After an electronic circuit drives the light source. Using lenses, this light is collected and transmitted to the wireless medium. On the other side of the FSO link, the light is collected by a lens in the receiver terminal and it is focused onto the optical detector, called photodiode. The photodiode converts the received optical power into an electrical current.

The next step is the transformation of this current into an amplified signal voltage. The signal is fed to the demodulator and there, the baseband signal information is retrieved from the modulated carrier.

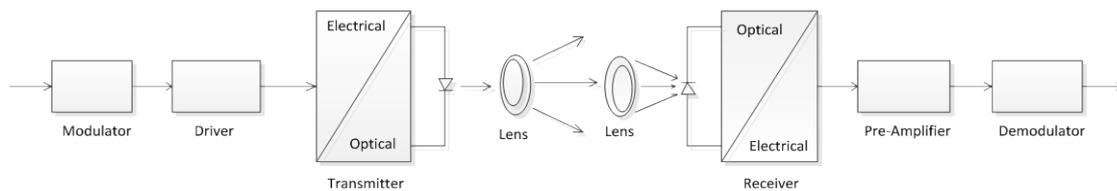


Figure 1 - Scheme of the FSO link

2.1. Transmitter

Two kinds of light sources have certain unique characteristics which make them attractive as lightwave communication sources: light-emitting diodes (LEDs) and laser diodes (LDs) [2]. They are the two semiconductor devices suitable for use in wireless optical communications.

A lower optical power, relatively small modulation bandwidth and harmonic distortion are some of the drawbacks of LEDs in comparison to semiconductor lasers. However, light-emitting diodes have a number of advantages that make them more useful for these kinds of communications. These differences must be considered in terms of the application and transmitter design.



Figure 2 - Laser Diode

One of the most important differences is the spatial and temporal coherence of laser light. Most LED sources are Lambertian and have a moderately large spontaneous spectral width. These factors govern, respectively, the amount of optical power which can be coupled into a fiber and the influence of material (or chromatic) dispersion on the dispersive properties of the fiber medium.

A second difference is in speed. The stimulated emission from lasers results in intrinsically faster optical rise and fall times in response to changes in drive current than rise and fall times obtainable with LEDs. The relative larger linewidth of LEDs is not a problem, because the dispersion does not matter in short distance transmission. A good LED can be operated at a bit rate of 100 Mbps which is adequate for most applications [3].

Thirdly, LEDs generate light almost linearly proportional to the current passing through the device. In contrast, lasers are threshold devices, and the lasing output is proportional to drive current only above threshold. Complicating the picture is the fact that the laser's threshold current is not constant but a function of device temperature and age. This has a significant impact on drive circuitry for lasers. The effect of temperature changes on LED output can be handled far more simply and in many cases is not even a problem. Lastly, differences exist in device reliability.

At present, LEDs have substantially longer operating lifetimes than lasers. The simpler construction of the LED leads to a much reduced cost, which is likely to always be maintained. The band gap of the semiconductor material is related to the wavelength of the LED. So it is very important to choose correctly the material to get the best features in the optical system.

Commercially available devices are optimized for the three fiber windows. Two of them are commonly used, one at around 850 nm and the other at about 1300 nm. In wireless link, the medium is the atmosphere, so the attenuation and dispersion are conditioned for the environment. In conclusion, absorption bands in the near infrared are produced by water vapour and carbon dioxide.

The most used option is the first window's GaAl/GaAlAs LED devices because they can be manufactured with a simple process and little costs than second windows InGaAsP devices. This choice is because there are also available more detectors in the first window. Although GaAs and GaAlAs diodes work in the same windows, they do not emit at the same wavelength. Infrared commercially available LEDs have a typical wavelength of 850 nm. For indoor applications, better results are expected with GaAlAs diodes.

Always before the transmitter is necessary the use of a LED driver. A current in the range of 100 to 200 mA must be switched ON and OFF at high speed through the light source in response to a low-level data input signal. A driver is a series of transistor switch which provides current gain and a small voltage drop. The maximum current flow through the optical source is limited by space charge and diffusion capacitance. The speed of the common-emitter driver can also be limited by the time required to remove minority-carried charge stored at the collector-base junction during saturation.

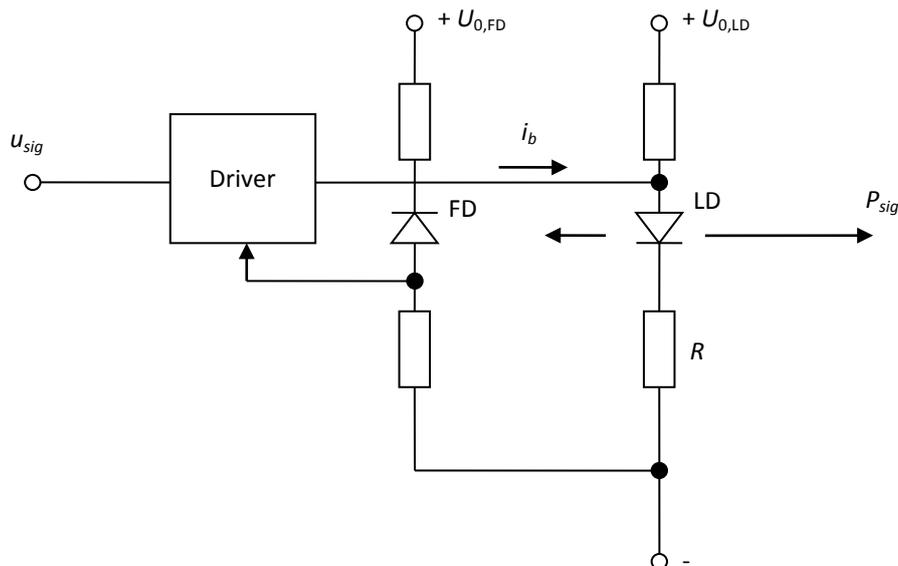


Figure 3 - Scheme of the connexion between LED and driver

In the figure we can see the use of two LEDs. One of them (FD) is used to measure the light emitted (the emitted power) and the other one (LD) is the main LED that provide the communication between the transmitter and receiver. The driver circuit is an electronic circuit that converts the voltage signal into an electronic current to modulate the light source. This is necessary because the sources are current injection.

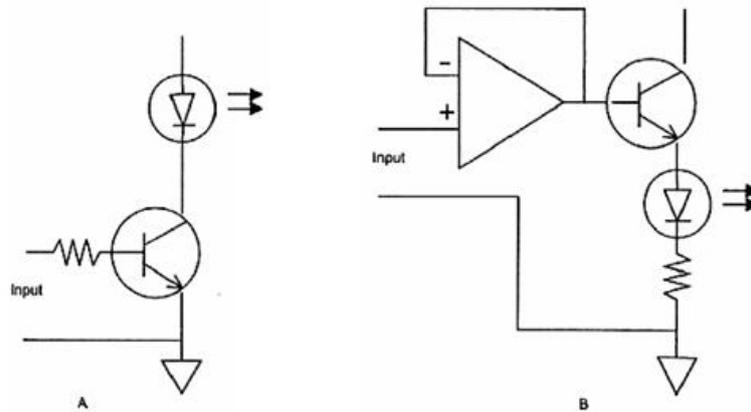


Figure 4 - A) LED driver, B) LD driver [5]

The main parameters which describe a typical transmitter are:

- Optical power $P[W]$: is the amount of optical power generated.
- Extinction coefficient $[K_e]$: Optical transmitters, no matter if directly or externally modulated, do not shut off completely when a zero is transmitted. This undesired effect is quantified by the extinction coefficient, which is defined as follows:

$$K_e = \frac{P_{max} - P_{min}}{P_{max} + P_{min}}$$

- Response time $t_r[s]$: is the time that the system takes to react to a given input.

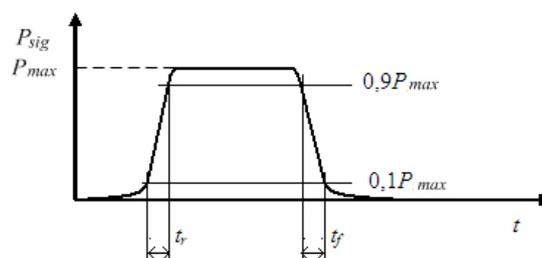


Figure 5 - Response Time

- FWHM: Full width at half maximum is an expression of the extent of a function, given by the difference between the two extreme values of the independent variable at which the dependent variable is equal to half of its maximum value.

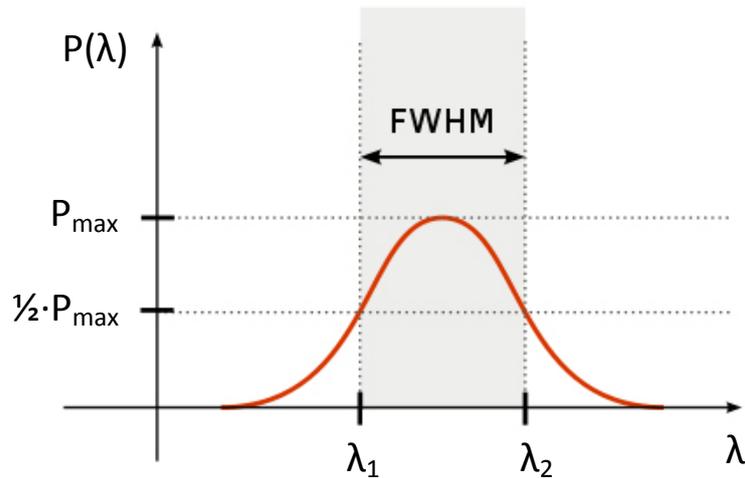


Figure 6 – FWHM

2.2.Receiver

Silicon photodiodes are semiconductor light sensors that generate a current or voltage when the P-N junction in the semiconductor is illuminated by light. These devices feature excellent linearity with respect to incident light, low internal noise, and a wide spectral response. They have a long life, and they are mechanically rugged, compact, and lightweight. However, some wavelengths are detected better (efficiency, response speed, etc.) than others.



Figure 7 - Silicon Photodiode

One of the important factors which determines the quantum efficiency is the absorption coefficient with a strong dependence of the wavelength. Only in a range of wavelength there will be an appreciable photocurrent generated. The long wavelength cutoff (λ_c) depends on the energy gap of the semiconductor and has a value of $\lambda_c = 1100 \text{ nm}$ for Si. The short wavelength cutoff comes because of the values of the absorption coefficient for short wavelengths are very large, and the radiation is absorbed very close the surface. To sum up, these two wavelengths limit the range of the photodetector's operation, and this is conditioned by the depletion region thickness.

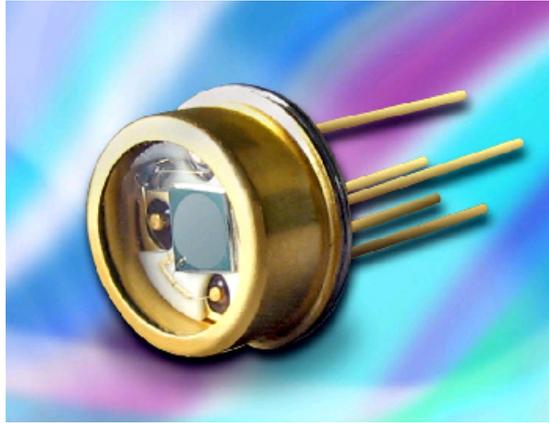


Figure 8 - Silicon PIN Photodiode

Photodiode performance is more readily optimized by the use of PIN structure. Silicon PIN photodiodes feature low capacitance, which enables them to deliver a wide bandwidth with only a low bias voltage. When connected to a high-speed preamplifier, their low terminal capacitance ensures a wide response speed and lower noise. This characteristic makes them ideal for high-speed photometry as well as optical communication.

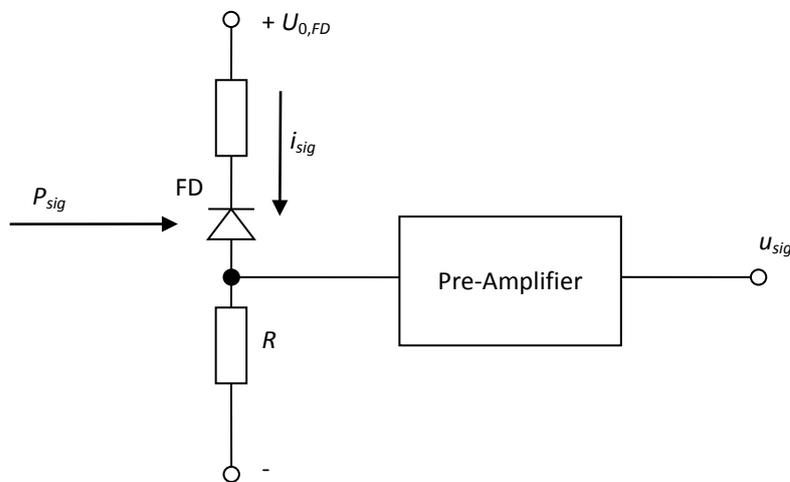


Figure 9 - Scheme of the connexion between the photodetector and pre-amplifier

Receiver sensitivity can be greatly enhanced in cases where an optical amplifier is employed in front of the photodiode. This method, also known as optical preamplification, is the most efficient if it used in combination with PIN photodiodes, since combination with APD could introduce relatively high shot noise and diminish the benefit of preamplification.

In conclusion, the choice of the photodetector material depends on the wavelength at which the system works. The best results in optical communications are given by silicon PIN photodiodes. These devices are characterized by these parameters:

- Responsivity $\mathfrak{R}[A/W]$: the ratio of generated photocurrent and incident optical power (neglecting noise influences), determined in the linear region of response.

$$\mathfrak{R} = \frac{I_0}{P_0}$$

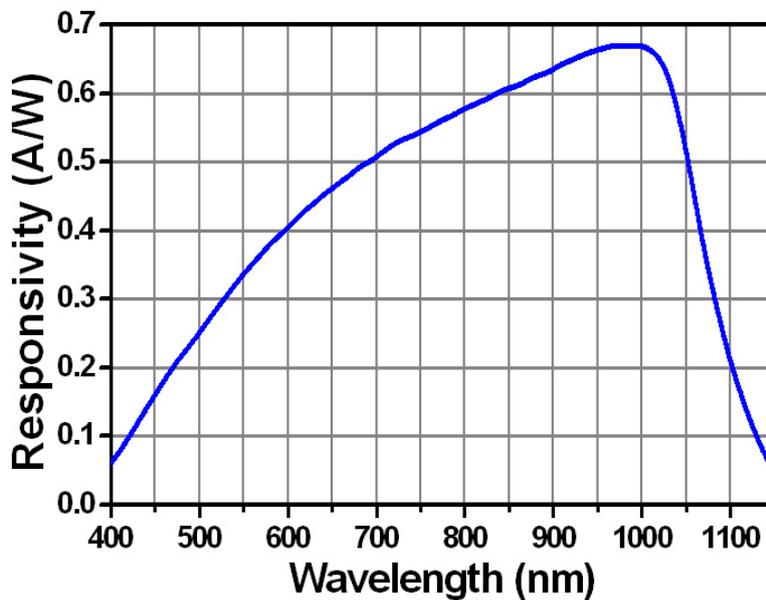


Figure 10 - Typical Responsivity

Responsivity is wavelength-dependent, and related to quantum efficiency (η) (the number of electrons released per incident photon) by

$$\mathfrak{R} = \frac{q}{hf} \cdot \eta$$

Where h is Planck's constant, f is the frequency of the incident radiation and q is the electron's charge.

- Sensitivity [dB]: the lowest detectable light level, which is typically determined by detection noise and significantly influenced by the required detection bandwidth.

- Dark current $I_D [nA]$: the reverse leakage current of a photodetector device in the absence of optical power entering the photo detector device. Dark current is an unwanted current in photo detectors. It occurs due to recombination of charge carriers within the depletion region and surface leakage current.
- Noise Equivalent Power $NEP [W\sqrt{Hz}]$: is a measure of the sensitivity of a photodetector or detector system. It is defined as the signal power that gives a signal-to-noise ratio of one in a one hertz output bandwidth.
- Signal to Noise Ratio $SNR [dB]$: the ratio of signal power to noise power at the input and output of an electronic device.
- Capacitance $C_S [pF]$: Total equivalent capacitance in parallel with the detector.
- Response time $\tau [ns]$: the time required by the generated carriers within the absorption region to travel that region under reverse bias conditions. Response time mainly depends on the thickness of the absorption region.

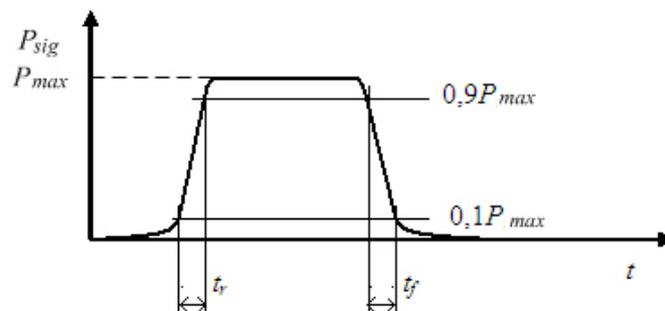


Figure 11 - Response time

2.3. Optics

A lens is an essential optical device with perfect or approximate axial symmetry, depending on the application. These lenses can transmit and refract light, converging or diverging the beam. While a basic lens consists of a single optical element, a compound lens is an array of simple lenses with a common axis allowing more optical aberrations to be corrected than a single element. The lenses are mostly made of glass or transparent plastic to refract electromagnetic radiation outside the casual spectrum.

The most common optical lenses are spherical. This means that their two surfaces are parts of the surfaces of spheres, with the lens axis ideally perpendicular to both surfaces. Each surface can be convex (bulging outwards from the lens), concave (depressed into the lens), or planar (flat). The imaginary line which joins the centres of the lenses is called the axis of the lens and typically passes through the physical centre of the lens, because of the way they are manufactured.

Toric or sphero-cylindrical lenses have surfaces with two different radii of curvature in two orthogonal planes. They have a different focal power in different meridians. This is a form of deliberate astigmatism.

More complex are aspheric lenses. These are lenses where one or both surfaces have a shape that is neither spherical nor cylindrical. Such lenses can produce images with much less aberration than standard simple lenses.

2.3.1. Types of lenses

Lens can be classified by the curvature of the two optical surfaces. In this way, there are many types of lenses:

- Biconvex: if both surfaces are convex.
- Equiconvex: if both surfaces have the same radius of curvature.
- Biconcave: if both surfaces are concave.
- Plano-convex or plano-concave: if one of the surfaces is flat and the other is convex or concave.
- Convex-concave or meniscus: if one of the surfaces is convex and the other concave.

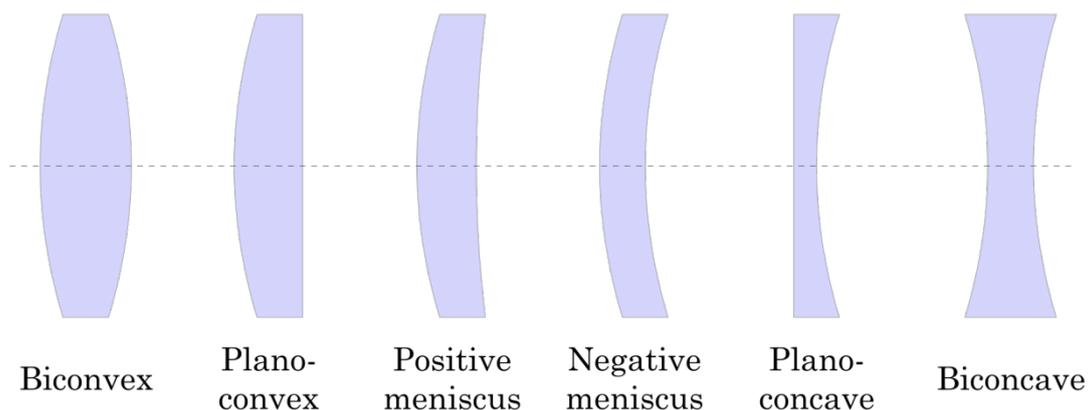


Figure 12 - Types of lenses [5]

When the lens is biconvex or plano-convex, a collimated beam of light travelling parallel to the lens axis and passing through the lens will be focused to a spot on the axis. The focal length will be the distance behind the lens and the focal point will be the point where the beam is focused. These kinds of lenses are called positive or converging lenses.

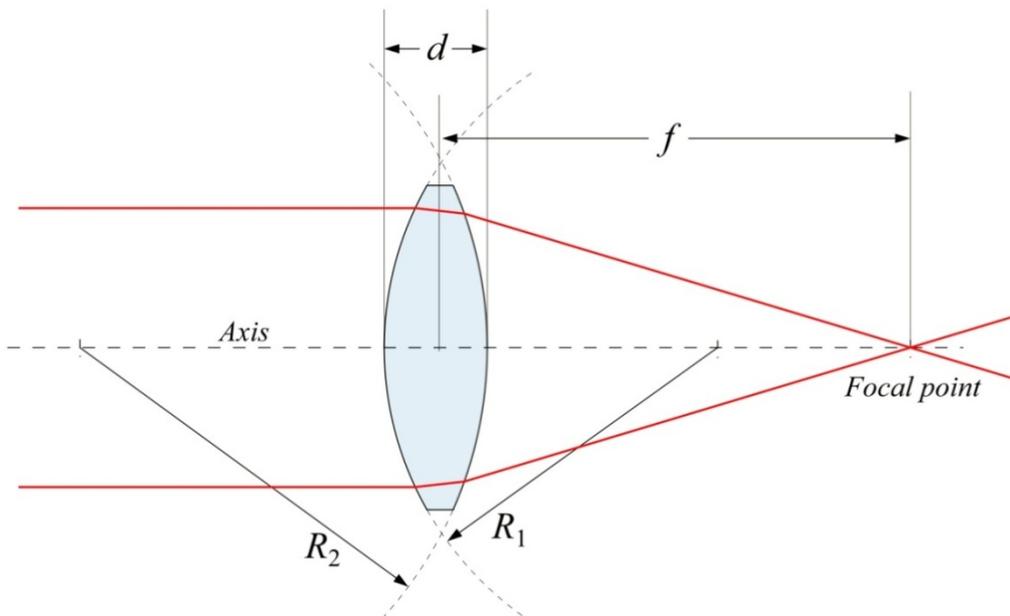


Figure 13 - Positive lens [5]

On the other hand, when the lens is biconcave or plano-concave, a collimated beam of light passing through the lens is diverged (spread). This kind of lens is called negative or diverging lens. The beam after passing through the lens appears to be emanating from a particular point on the axis in front of the lens. The distance from this point to the lens is also known as the focal length. In this case, the focal length is negative instead of the positive focal length of a converging lens.

Convex-concave (meniscus) lenses can be either positive or negative, depending on the relative curvatures of the two surfaces. A negative meniscus lens has a steeper concave surface and will be thinner at the centre than at the periphery. Conversely, a positive meniscus lens has a steeper convex surface and will be thicker at the centre than at the periphery.

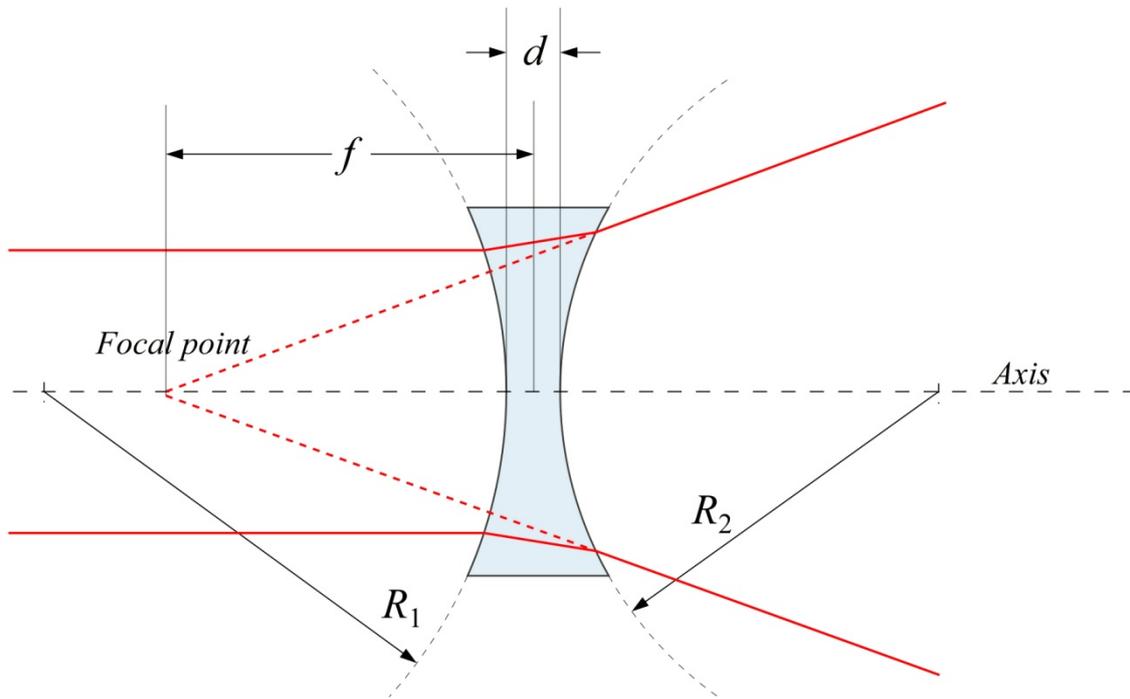


Figure 14 - Negative lens [5]

An ideal thin lens with two surfaces of equal curvature would have zero optical power, meaning that it would neither converge nor diverge light. All real lenses have a nonzero thickness, however, which affects the optical power. To obtain exactly zero optical power, a meniscus lens must have slightly unequal curvatures to account for the effect of the lens' thickness.

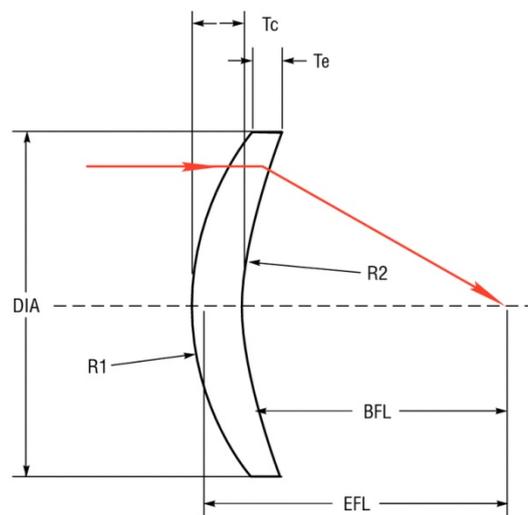


Figure 15 - Negative meniscus lens [6]

There are other kind of lenses called 'Fresnel Lens' that replace the curved surface of a conventional lens with a series of concentric grooves, molded into the surface of a thin, lightweight plastic sheet. The design allows the construction of lenses of large aperture and short focal length without the mass and volume of material that would be required by a lens of conventional design.

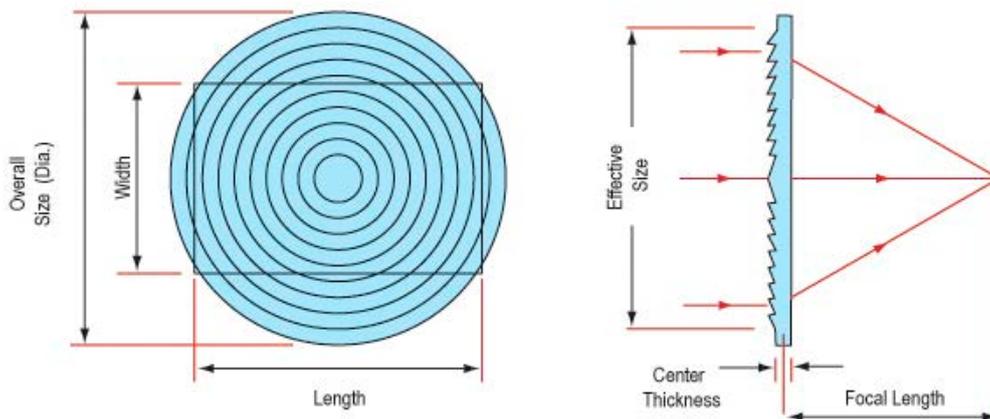


Figure 16 - Aspherically Contoured Fresnel Lenses [7]

The grooves act as individual refracting surfaces, like tiny prisms when viewed in cross section, bending parallel rays in a very close approximation to a common focal length. Very little light is lost by absorption because the lens is thin. The quality images depends on density of the groove: High groove density allows higher quality images, while low groove density yields better efficiency (as needed in light gathering applications). In infinite conjugate systems, the grooved side of the lens should face the longer conjugate.

Fresnel lenses are most often used in light gathering applications, such as condenser systems or emitter/detector setups. In indoor optical communications, these kinds of lenses are used in the receivers to capture as much light as possible. Fresnel lenses can also be used as magnifiers or projection lenses; however, due to the high level of distortion, this is not recommended.

The principal focal length of a lens is determined by the index of refraction of the glass, the radii of curvature of the surfaces, and the medium in which the lens resides. It can be calculated from the lens-maker's formula for thin lenses.

2.3.2. Lensmaker's equation

The Lensmaker's equation relates the focal length of a simple lens with the spherical curvature of its two faces [8]:

$$\frac{1}{f} = (n - 1) \left[\frac{1}{R_1} - \frac{1}{R_2} + \frac{(n - 1)d}{nR_1R_2} \right]$$

Where f is the focal length of the lens, n is the refractive index of the lens material, R_1 is the radius of curvature of the lens surface closest to the light source, R_2 is the radius of curvature of the lens surface farthest from the light source, and d is the thickness of the lens (the distance along the lens axis between the two surface vertices).

The sign of R_i is determined by the location of the centre of curvature along the optic axis, with the origin at the centre of the lens. If the first surface is convex, R_1 is positive, and if it is concave, R_1 is negative. On the other hand, if R_2 is positive, the surface is concave, and if R_2 is negative, it is convex. When one of the radiuses is infinite, the surface is flat.

Other form to understand it is explained as follow: The radius of curvature is taken to be positive if light, after passing through the surface of the lens in question (either the front or back surface), can pass through the centre of curvature of that surface. Otherwise, the radius of curvature is negative. This is very similar to the determination of the sign of the image position (d), in which the position is taken to be positive if light can pass through the image point after it passes through the lens, and negative otherwise.

With this convention the signs are determined by the shapes of the lens surfaces, and are independent of the direction in which light travels through the lens.

In the case that thickness is negligible compared to the focal length of the lens (d is smaller than R_1 and R_2), the lens can be approximated as a thin lens. The expression will change in this situation as follow:

$$\frac{1}{f} \approx (n - 1) \left[\frac{1}{R_1} - \frac{1}{R_2} \right]$$

Where $1/f$ is the optical power of the lens, measured in dioptries (m^{-1}). Lenses have the same focal length when light travels from the back to the front as when light goes from the front to the back, although other properties of the lens, such as the aberrations are not necessarily the same in both directions.

Converging lens will focus a collimated beam to a spot (known as the focal point) at a distance. Conversely, a point source of light placed at the focal point will be converted into a collimated beam by lens.

These two cases are examples of image formation in lenses. In the first situation, an object situated at an infinite distance (represented by a collimated beam of waves) is focused to an image at the focal point of lens. In the second situation, an object at the focal length distance from the lens is imaged at infinity. The plane perpendicular to the lens axis situated at a distance f from the lens is called the focal plane.

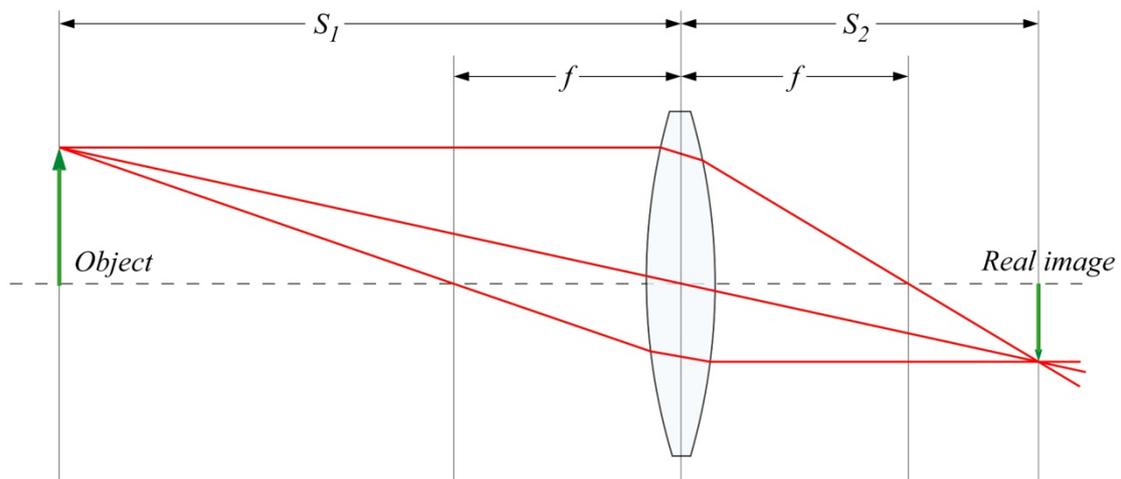


Figure 17 – Distance between object and real image

This figure shows the distance between the object and the lens (S_1) and between the lens and the image (S_2). For a lens of negligible thickness, in air, the distances are related by the thin lens formula:

$$\frac{1}{f} = \frac{1}{S_1} + \frac{1}{S_2}$$

This means that if an object at a distance S_1 is placed in front of a positive lens of focal length f , as long as $S_1 > f$, a sharp projection of the object will take place at distance S_2 . This is the most important concept in photography and the human eye.

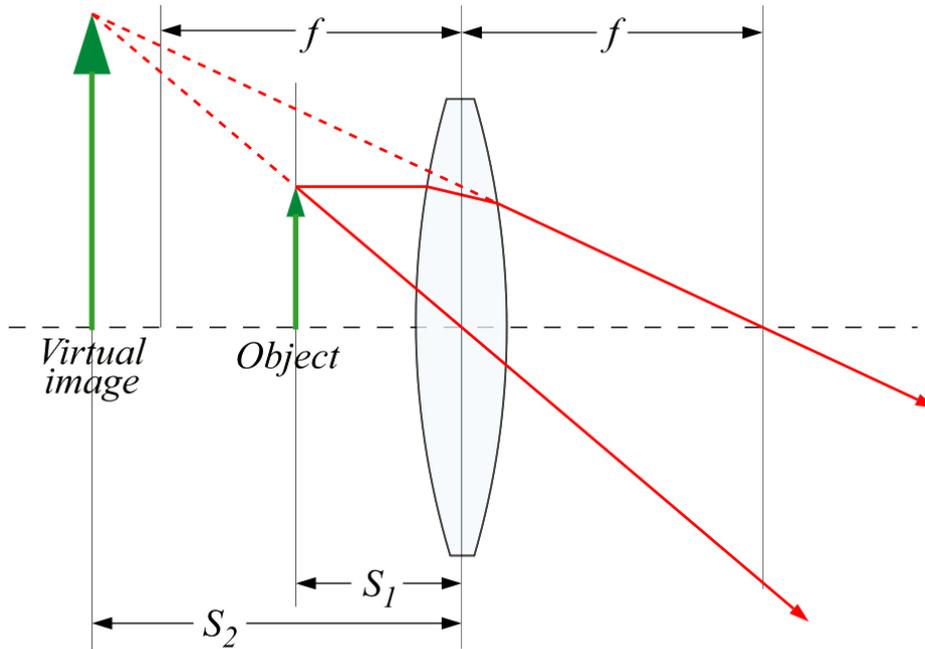


Figure 18 - Distances in optics

2.3.3. Aberrations

The lenses are not perfect, this means that do not form perfect images. Always there are some degree of distortion (aberration) introduced. The occurrence of these aberrations can be minimised by a careful design of the system. The different types of aberration are explained as follow.

2.3.3.1. Spherical aberration

Although the best way to make a lens is not by a spherical surface, it is the simplest shape to do it. That's the reason that produces this kind of aberration. The main effect it is that it causes beams parallel to the lens axis to be focused in a slightly different place than beams close to the axis. This effect produces a blurring of the image.

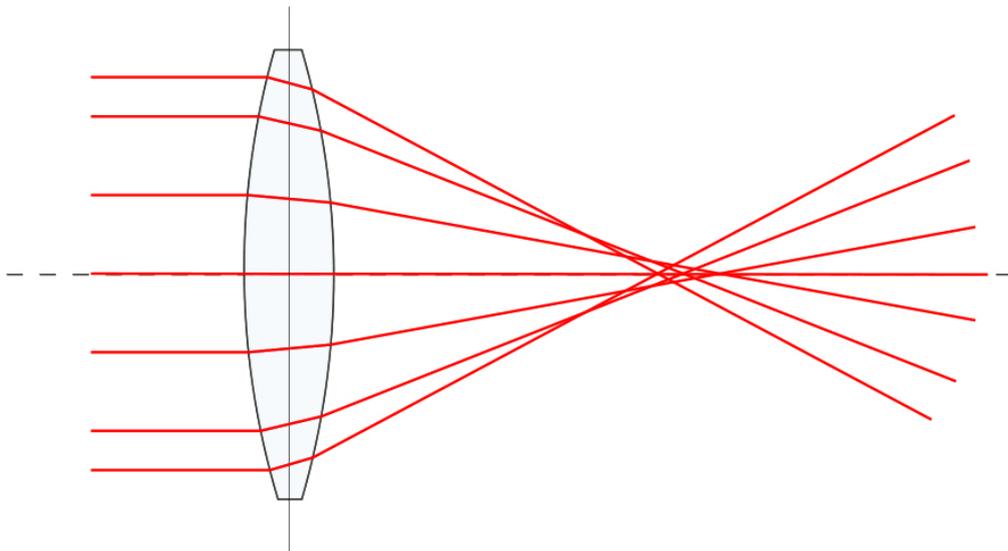


Figure 19 - Spherical aberration

To minimize this effect it can be used aspheric lenses. These non-spherical lenses were difficult to make and extremely expensive, but the last technological advances have reduced the costs substantially. Another way to minimize this effect is by a careful choice of the curvature of the surfaces. For example, you can use a plano-convex lens which is used to focus a collimated beam with the convex side toward the beam source to produce a sharper focal spot.

2.3.3.2. Coma

Coma aberration is produced when an object out of the optical axis is imaged, where rays pass through the lens at an angle to the axis θ . Rays which pass through the centre of the lens of focal length f are focused at a point with distance $f \cdot \tan \theta$ from the axis. The rays passing through the outer margins of the lens are focused at different points. When these points are focused further from the axis are called positive coma and when they are focused closer to the axis is called negative coma. In general, a bundle of parallel rays passing through the lens at a fixed distance from the centre of the lens are focused to a ring-shaped image in the focal plane, known as a *comatic circle*. The sum of all these circles results in a comet-like flare.

To minimize the coma aberration, is important to choose the curvature of the two lens surfaces according the application. Lenses in which both spherical aberration and coma are minimised are called *bestform lenses*.

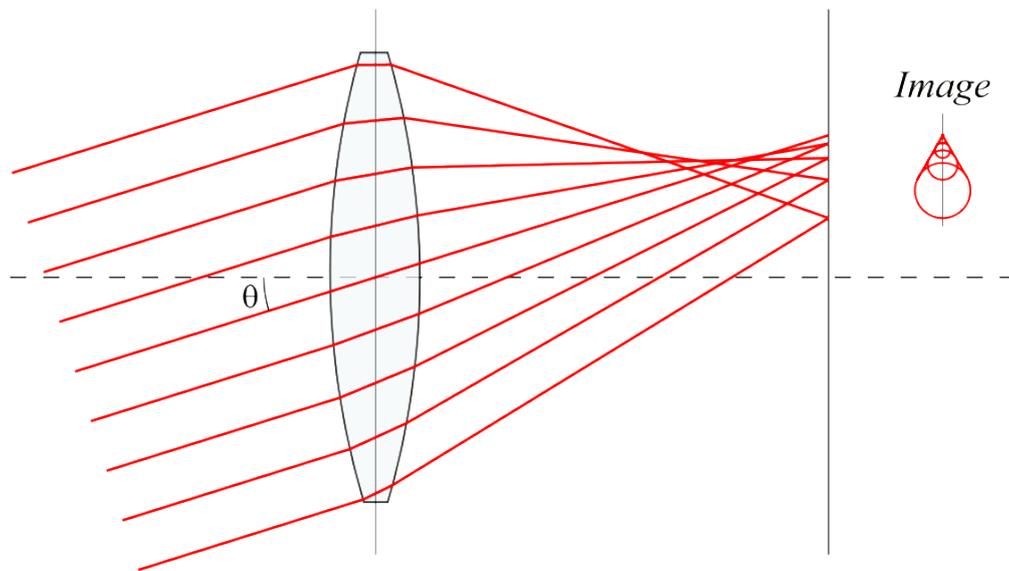


Figure 20 - Coma Aberration

2.3.3.3. Chromatic aberration

The last aberration is derived of the variations of the refractive index (n) of the lens material with the wavelength of light. So it is caused by the dispersion of the lens material. As mentioned above, f is dependent upon n , so different wavelengths of light (different colours) will be focused to different positions and produce on the image fringes of colour.

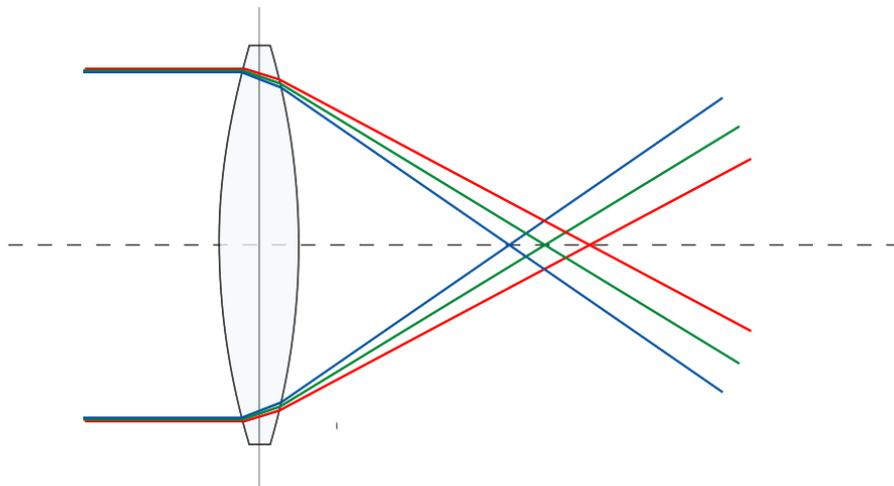


Figure 21 - Chromatic Aberration

This aberration can be minimised by using an achromatic doublet (achromat) in which two materials with differing dispersion are bonded together to form a single lens. In this way, it is possible to reduce the amount of chromatic aberration over a certain range of wavelengths. Another choice is the use of an apochromat, which is a lens (or lens system) which has even better correction of chromatic aberration, combined with improved correction of spherical aberration. The only drawback is that apochromats are much more expensive than achromats.

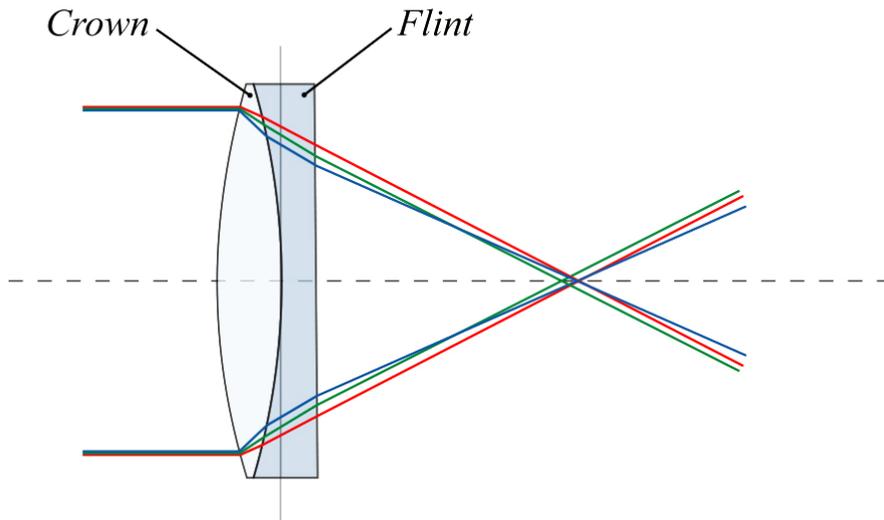


Figure 22 - Achromatic Doublet

The use of different lens materials may also minimise chromatic aberration, such as specialised coatings or lenses made from the crystal fluorite. This last material has the highest Abbe number which indicates that the material has low dispersion.

2.4. The indoor FSO link – definition and operation

There are only two technologies of establishing a wireless link which support the high speed data transmission necessary in indoor wireless LANs. These are the radio-frequency waves and the unguided optical signals.

2.4.1. Radio Frequency

Radio frequency (RF) is a rate of oscillation in the range of about 3 kHz to 300 GHz, which corresponds to the frequency of radio waves, and the alternating currents which carry radio signals. To implement a RF link are used two techniques: narrowband and spread-spectrum. Narrowband modulation has problems with multipath transmission and these schemes are very sensitive to interference. For this reason, spread-spectrum technology is more suitable.

Spread-spectrum techniques are methods by which a signal generated in a particular bandwidth is deliberately spread in the frequency domain, resulting in a signal with a wider bandwidth.

But Wireless RF systems present three important problems to be solved. Firstly, there are not too many free frequencies for develop RF WANs. The industrial, scientific and medical (ISM) radio bands are portions of the radio spectrum reserved internationally for the use of radio frequency (RF) energy for these purposes. The most common Wireless LANs are Bluetooth (2450 MHz) and IEEE 802.11/Wi-Fi (2450 MHz and 5800 MHz bands). These bands are limited and require the developer to make a frequency plan. Secondly, if a lot of WLANs are working in the same building, the RF signal can cross a wall and create interferences. Finally, is important a good encryption to prevent data theft because of it is impossible restrict the RF signals to a room.

2.4.2. Infrared

Although lasers were used to communicate information through the free air, they were substituted by optical-fiber technology. Nowadays, optoelectronic devices in the 840 to 950 nm range are used in television, wireless headphones and remote controls.

There are three important problems in a wireless IR link that must be solved: optical power emitted, multipath intersymbol interference (ISI) and environment noise. Is known that LDs are fast and powerful than LEDs, but they have the disadvantage of spatial coherence. The light in this case is confined in a small area and high power densities are obtained. From the point of view of eye safety, lasers emitters are not adequate for wireless indoor applications. The main drawback of LEDs is the power-speed product. There are not devices with high power and high speed, necessary for diffuse links. So the biggest problem is how to get enough power to the receivers.

We can find two alternatives for the organization of receivers and transmitters [9]:

2.4.2.1. Pure diffuse

The power is launched in a wide-aperture angle, and after a few reflections on the walls the optical power is converted in isotropic radiation. With this distribution, it is no important the receiver orientation because of the optical power comes from every direction. The multipath propagation causes a limitation on the speed of the channel. The main advantage of this links is the mobility of the transmitters and receivers.

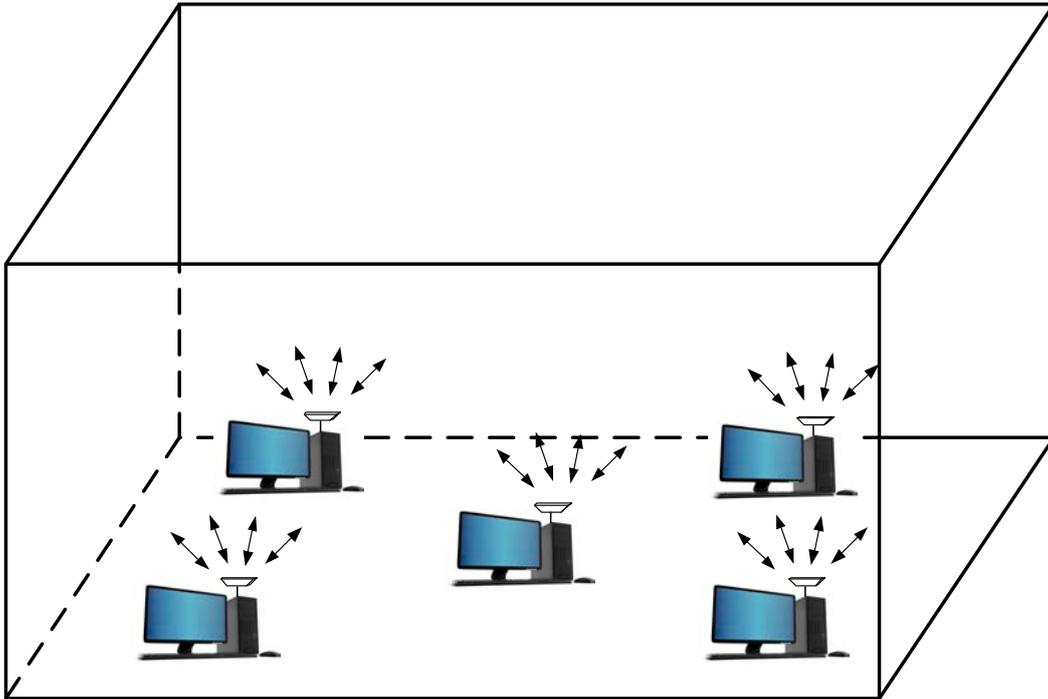


Figure 23 - Diffuse link

2.4.2.2. *Quasidiffuse links*

This links do not have de problem of the multipath dispersion, so the speed is higher. The power is sent to a fixed place on the ceiling (*satellite*) as a thin light beam and the receivers are facing the satellite. Quassidiffuse links can be divided in two little groups: passive and active links.

In the first one, the beam is sent to a scattered surface (for example, a small area on a white painted ceiling). On the other hand, for large rooms is needed an active satellite. This is a repeater with several photodiodes and LEDs which cover more area. In this case, almost all the power needed is in the satellite so in the terminals the power is low.

Both active and passive satellites avoid multipath dispersion because the receivers have a little field of vision and only the signal from the satellite get to the photodiodes and not the reflected signal from the walls. In these kinds of connections, all the equipment is at the terminals, so it is not necessary any installation.

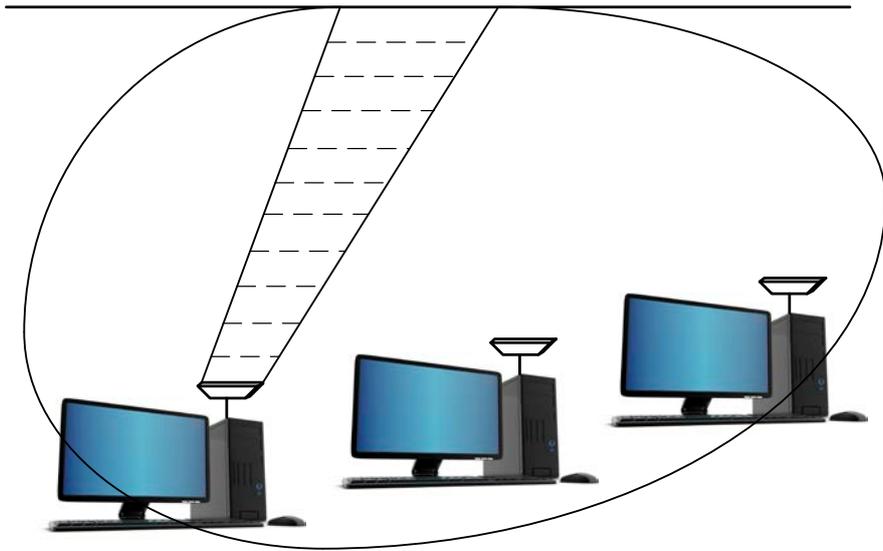


Figure 24 - Passive quasidiffuse link

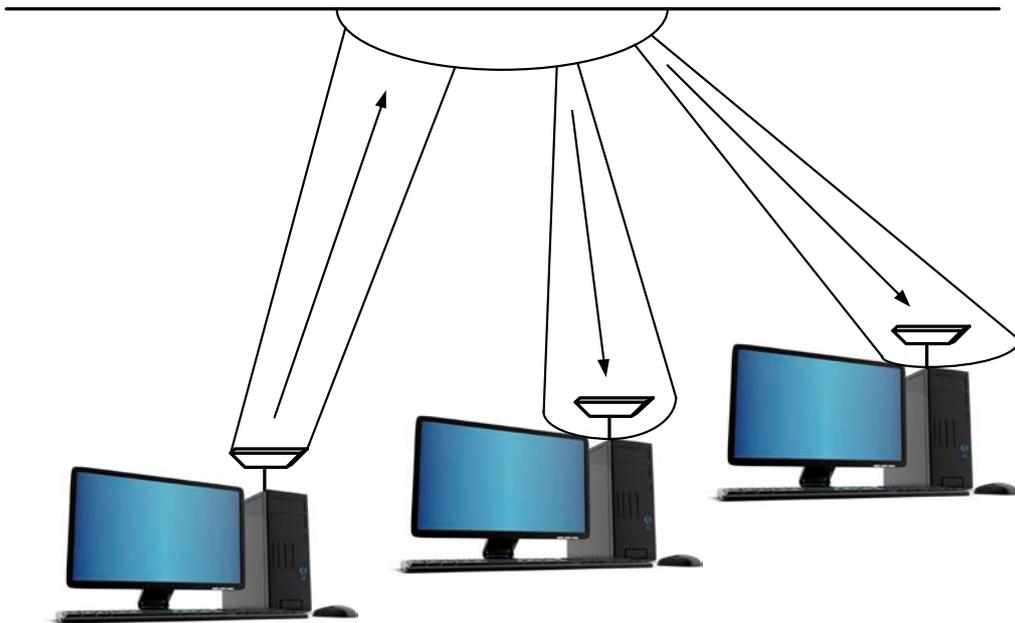


Figure 25 - Active quasidiffuse link

2.5. Indoor FSO links – Classification and requirements

2.5.1. Classification

The different kinds of links for indoor optical wireless communications can be taken, depending on the existence of a line-of-sight (LOS) path between the transmitter and the receiver, and the degree of directionality [10].

With the use of directed links, the power efficiency is maximised using directional transmitters and receivers. This kind of systems needs alignment of the transmitter, the receiver, or both, making them less convenient to use for certain applications. Non-directed links work with wide-angle transmitters and receivers, removing the need for pointing and make them more convenient.

Whereas non-LOS systems use reflections of light from the ceiling and walls, LOS systems have an uninterrupted line of sight path. Non-LOS links increase link robustness as they allow the system to operate even when obstacles are placed between the transmitter and receiver. On the other hand, LOS links improve power efficiency and minimise multipath distortion.

The transmitted power in directed-LOS links is concentrated into a narrow optical beam, making possible the use of narrower field-of-view (FOV) receivers. Also these links do not suffer from multipath distortion, and a maximum distance can be obtained independently of the reflective properties of the room, as far as the line of sight is not interrupted. The drawback of this configuration is that it is susceptible to blocking.

Hybrid-non-LOS systems do not present the blocking problem, but have multipath distortion that increase as the area is increased.

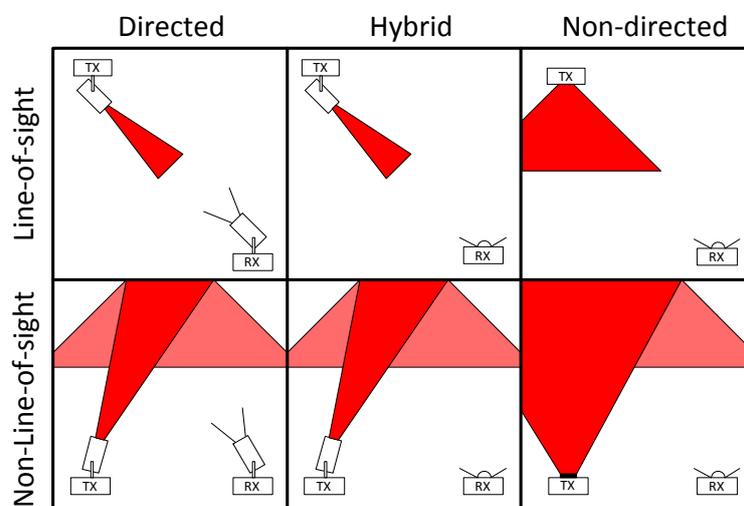


Figure 26 - Classification of infrared links

In conclusion, the most attractive configuration is the non-directed-non-LOS, also called diffuse. These links do not require a direct line of sight or alignment between the transmitter and the receiver. The reflective properties of the ceiling and the walls are used to spread optical waves as uniformly as possible. This makes it the most robust and flexible configuration. The only drawback is that suffer multipath dispersion and higher optical losses.

2.5.2. Requirements

2.5.2.1. Reliability

The new medium should be as reliable as a wire system. LAN communication relies on an almost errorless link (probability of error: $BER \leq 10^{-9}$). When an error is detected in a data packet it is not corrected, the packet is resent. WLANs must try to keep the error rate at the same level as cabled LANs and this task is not easy. Wireless systems use lower signal-to-noise ratios (S/N) than cabled links. An additional source of errors and signal losses are the cell changes. The European standard for wireless communications states that during a cell change, the link is kept with both cells on different channels at the same time. If the error rate is kept low (10^{-6}), the system performance will be good. Larger error rates should be solved by the hardware, not by the network operating system or drivers.

2.5.2.2. Transparency

The aim of optical wireless links is not replace the existing cabled LANs; they will share the environment, so the software has to work with both types. Only the first and second layers (physical and link) of the OSI model will be different. To do an installation interfaces equipment must be used to convert the electrical signals at the communication card to/from wireless signals. These interfaces must work with total transparency for the user.

2.5.2.3. Throughput

WLANs should be able to work at the same data rate as cabled LANs. The technological possibilities of WLAN are far from high-speed LANs. The best implementation is a hybrid network with medium speed WLAN cells working with data rates up to 20 Mbps connected to a high-speed cabled LAN.

2.5.2.4. Security

Data encryption is mandatory in WLANs. To avoid degrading the performance, this has to be done by hardware using encryption codes, or by the same method of transmission (using spread-spectrum techniques).

If no security controls are used, the network will be exposed to unfriendly access. These actions cannot be detected by the lower levels, so their control is done by the transport and upper levels.

2.5.2.5. Mobility

We can differentiate two kinds of mobility: Full mobility and weak mobility or portability. In the first one, you can send and receive information while moving inside the area covered by the WLAN. The different environments and relative orientations make this mobility hard to implement. On the other hand, in the second type you can have a connection to the network by placing a terminal within the area covered by the WLAN.

2.5.2.6. Network topology

The most popular topology used in WLANs is bus-based. Nevertheless, if terminals are grouped in clusters and in fixed places, a wireless ring can be made using point-to-point links between clusters. Interfaces for Ethernet, token-ring, SNA, and other well-known LANs are available by WLANs manufacturers. It is important to know that the whole WLAN is a cell-based network.

2.5.2.7. Flexibility

The number of active nodes in WLANs can change while the network is working, so the protocols for the inclusion or exclusion of a terminal should be minimized. Several types of networks need to know how many active nodes are there, so a protocol to get in or out is needed.

2.5.2.8. Price

It is known that the equipment of WLANs are more complex and expensive than the equipment of a cabled LAN. Take a couple of new corporate office buildings for example. Traditional wired costs may include CAT5 copper cable runs in the ceiling and through walls, along with their corresponding data drops needed on just about every wall feasible.

A wireless LAN also still requires installation and some degree of cabling; however, one access point can usually be installed in the amount of time it takes to terminate one data drop. To make this part of the solution complete, you may also need to throw in the cost of traditional RJ-45-based network cards, depending on whether the systems come with them preinstalled.

In conclusion, WLANs can be more expensive because of the equipment, but in a near future, they are more profitable than cabled LANs.

2.5.2.9. Safety and Regulations

With indoor wireless communications using infrared beams, the power levels must be innocuous to human beings, so eye safety issues must be addressed. Infrared radiation can damage the retina and the cornea of the eye when used inappropriately. The damage produced by an infrared source will depend on the exposure time, the wavelength and the power of the signal. The International Electrotechnical Commission (IEC) defines the allowable exposure limits (AEL).

This AEL ensures that the system is safe under all circumstances of use and it does not require warning labels. The limits are a function of the size of the optical sources, the wavelength and the viewing time. The sources are classified depending on if the eye can focus the source (point source), or if the source from an extended image on the retina (extended or large area sources).

The following formulae were established by the American National Standards Institute as a guideline for the safe use of lasers. The maximum permissible exposure (MPE) values of intrabeam viewing for a nearly point source are:

$$MPE = 1.8 \cdot C_A \cdot t^{-0.25} \quad mW/cm^2 \quad \text{for } 50\mu < t < 1k$$

$$MPE = 0.32 \cdot C_A \quad mW/cm^2 \quad \text{for } 1k < t < 30k$$

Where t is in seconds, $C_A = 10^{0.002(\lambda - 700)}$ for $\lambda = 700nm - 1050nm$ and $C_A = 5$ for $\lambda = 1050nm - 1400nm$.

The eye retina is safe up to $560 mW/cm^2$ for one second exposure and up to $100 mW/cm^2$ for 100 seconds or longer exposure when operating at $\lambda > 1400nm$. This sharp increase of safety threshold is due to water absorption in the cornea preventing laser power from reaching the retina.

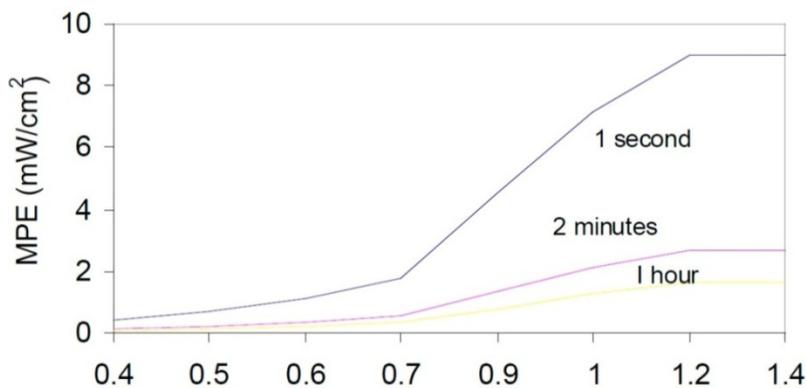


Figure 27 - Maximum permissible exposure value vs. wavelength for various intrabeam viewing exposure times

For instance, lasers are point source emitters that must have reduced emission power levels if they have to satisfy the standard. The follow table show a safety classification for laser sources.

	880 nm	1310 nm	1550 nm
Class 1	< 0.5 mW	< 8.8 mW	< 10 mW
Class 2	N/A	N/A	N/A
Class 3A	0.5 – 0.25 mW	8.8 – 45 mW	10 – 50 mW
Class 3B	2.5 – 500 mW	45 – 500 mW	50 – 500 mW

Table 1 - Safety classification

Class 1 products are defined as inherently safe, that means that they are safe even when viewed with optical instruments. Optical wireless systems are required to fall into this category. LED's are large-area emitters that can be operated safely at larger emission power levels. Those, and their reduced cost, make them the preferred optical source for indoor wireless systems.

2.6. Indoor FSO links – Limits

2.6.1. Bit Rate

It depends on the response time of the detector. The parameter which determines the bit rate is the raise time: the time required for a signal to change from a specified low value to a specified high value. Typically these values are 10% and 90%. The expression that determines de bit rate is:

$$B = \frac{1}{t_r}$$

Therefore, if a typical raise time of a detector is about 10 ns, the link is going to work with a bit rate of:

$$B = \frac{1}{10^{-8} \text{ ns}} = \frac{10^8 \text{ bit}}{\text{s}} = 100 \text{ Mbps}$$

2.6.2. Bit Error Rate (BER)

Although it depends on the Signal to Noise value, typically a good communication is about:

$$BER = 10^{-9}$$

2.6.3. Range

The limit in distance is the height of the room. Almost all offices have a height of 3 or 4 meters, so that's the range of the link in distance. This is one of the reason because of the atmosphere phenomena is not important for indoor wireless optical links.

2.6.4. Power of the LED

The power of the LED is limited by the wavelength which is used. The power levels must be innocuous to human beings, so eye safety issues must be addressed. Typical values of power sources are:

$$P_{LED}(\lambda = 1550 \text{ nm}) = 100 \text{ mW}$$

$$P_{LED}(\lambda = 850 \text{ nm}) = 10 \text{ mW}$$

2.6.5. Noise Equivalent Power (NEP)

Noise-equivalent power (NEP) is the radiant power that produces a signal-to-noise ratio of unity at the output of a given optical detector, operating wavelength, and effective noise bandwidth. Some manufacturers and authors define NEP as the minimum detectable power per square root bandwidth.

This parameter is a measure of the sensitivity of a photodetector and depends on two elements. The first one is the NEP_1 that the manufacturer offers and the seconds is the bit rate.

$$NEP = NEP_1 \cdot \sqrt{B}$$
$$NEP(dBm) = 10 \log(NEP_1 \cdot \sqrt{B})$$

If the photodetector has a noise equivalent power of $50 \text{ fW}/\sqrt{\text{Hz}}$ and the bit rate is 100 Mbps , we have a NEP of:

$$NEP = NEP_1 \cdot \sqrt{B} = 50 \cdot 10^{-15} \frac{W}{\sqrt{\text{Hz}}} \cdot \sqrt{100 \cdot 10^6 \text{ Mbps}} = 5 \cdot 10^{-10} \text{ W}$$
$$= 5 \cdot 10^{-7} \text{ mW}$$

$$NEP(dBm) = 10 \log(5 \cdot 10^{-7}) \approx -63 \text{ dBm}$$

2.6.6. Receiver sensitivity

To ensure a BER of 10^{-9} we need a Signal to Noise of 15.6 dB . Thus, the receiver sensitivity can be calculated as:

$$P_{p,0} = NEP(dBm) + SNR(dB) = -63 \text{ dBm} + 15.6 \text{ dB} = -47.4 \text{ dBm}$$

3. Atmospheric phenomena

Although IR wireless links are most commonly recommended for indoors communications, spanning long outdoor may sometimes be required for specific purposes. The impossibility to provide all-weather operation is the reason why atmospheric optical LANs are not commonly used. The use of atmospheric spans as the communication channel introduces a set of random, weather dependent disturbances that affect the quality of the communication.

There are two phenomena that make the atmospheric channel different from free space: refraction and attenuation. The refractive index of the air is very close to unity, but relative variations in long paths can lead to small angular variations in the beam direction. This may be comparable with the divergence of the beam.

On the other hand, absorption lines of atmospheric gases (mostly water vapour) and scattering by gas molecules, and more noticeably by aerosols, introduce additional losses to be taken into account in the link budget.

However, the main characteristic of the atmospheric channel is its randomness. The atmosphere is not an all-weather channel in the optical band. If we suppose “good weather”, there are random variations of the local value of the refractive index with impact on the quality of the communications link. Turbulence flux of hot air is usually induced near the ground due to its heating during the day. Turbulence is also intensive in the neighbourhood of thermal inversions, where two atmospheric layers with different temperatures slide one over the other.

3.1. Attenuation

The high energy transported by optical photons makes the radiation-matter interactions unavoidably important at the optical frequencies. Some atmospheric gas molecules absorb IR energy in exchange for modifying its internal vibration state and act as scattering centres. In conclusion, the atmosphere is a turbid fluid. Particles in suspension (aerosols) degrade the atmosphere transparency by acting as absorbing pollutants by producing a strong scattering of the incoming radiation.

Therefore, the total attenuation coefficient (per length unit) is the sum of the contributions of molecular absorption (α_m), molecular scattering (α_R) and aerosol extinction (α_a).

$$\alpha = \alpha_m + \alpha_R + \alpha_a$$

3.1.1. Gas absorption

A molecule can change its vibration state, absorbing or emitting an infrared photon of the adequate energy. The Earth's atmosphere is mostly composed by atomic gases, which does not form molecules or diatomic molecules with identical atoms where vibration transitions are not allowed.

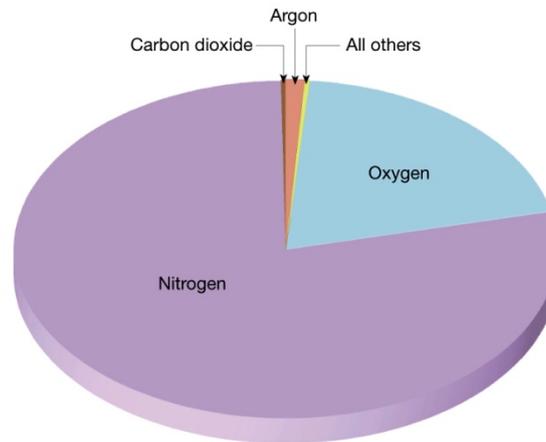


Figure 28 - Composition of Earth's atmosphere

The main absorbing gases that can be found in the atmosphere are water vapour and carbon dioxide. Evaporation and condensation processes make the water content of the atmosphere very unstable and its concentration varying along with time and space. The amount of water-vapour molecules can be determined from absolute humidity and atmospheric pressure. The CO_2 also have a variable concentration and it is more important in urban areas. Other absorbers are present in the atmosphere in very low concentrations.

The expression which determines the gas absorption attenuation is the following:

$$\alpha_m(\nu) = \frac{S}{\pi} \frac{\gamma}{(\nu - \nu_0)^2 + \gamma^2}$$

Where:

- S : line strength (each absorption is defined by a line strength related to the probability of the transition and a line width related to the statistical uncertainties involved in the transmission phenomenon).
- ν : wavenumber (i.e., the inverse of the wavelength)
- γ : line half-width, which depends on pressure and temperature.

The absorption lines are organised in different spectral bands related to vibration-rotation energy transitions.

The atmosphere is opaque in some regions where there is a high density of absorption lines. Also there are some “atmospheric windows” where the optical transmission is possible. One of them is the visible and near IR window where lasers and LEDs can be used. Therefore, for indoor communications the main absorber is water vapour.

3.1.2. Molecular scattering

Rayleigh scattering, named after the British physicist Lord Rayleigh, is the elastic scattering of light or other electromagnetic radiation by particles much smaller than the wavelength of the light (such as gas molecules when compared to visible light) [11]. These particles scatter part of the incident radiation with a scattering coefficient inversely proportional to the fourth of the wavelength. It can occur when light travels through transparent solids and liquids, but is most prominently seen in gases.

Rayleigh scattering of sunlight in the atmosphere causes diffuse sky radiation, which is the reason for the blue colour of the sky and the yellow tone of the sun itself.

The formula can be obtained if it is counted the scattering losses and the refractive-index dependence on atmospheric pressure and temperature [12].

$$\alpha_R = 2.9154 \cdot 10^{-4} \frac{1 + 6.6 \cdot 10^{-3} / \lambda^2 P}{\lambda^4 T}$$

Where:

- P : pressure [mb]
- T : temperature [K]
- λ : wavelength [μm]

3.1.3. Aerosol extinction

All aerosols have particles of many different sizes, each of them showing its characteristic wavelength dependence of scattering. Aerosol composition depends very much on the geographic location, which determines the nature and amount of particles in suspension.

For visible light, the relation between aerosol extinction and atmospheric visibility expressed in Km (vis) is:

$$\alpha(\lambda = 0.55 \mu m) = \frac{3.912}{vis}$$

Precipitation is a particular kind of aerosol, characterized by a large particle size. Since particles are large when compared to any relevant infrared wavelength, its contribution to the atmospheric attenuation is wavelength independent in the entire optical spectrum. The attenuation is proportional to a variable power of the precipitation rate R and depends on the rainfall type (parameters A and B)

$$\alpha_a = A \cdot R^B$$

3.2. Turbulence

The different refractive indices depend on the air masses at different temperatures. Therefore, the refractive index depends on the local temperature and air density. Atmospheric turbulences induce random variations in the local index of refraction. A measure of these fluctuations is given by the mean square difference between the refractive indices of two points as a function of the distance between them (index structure function).

The Kolmogorov analysis of the turbulent flux predicts that the index structure function is proportional to the distance between points raised to two-thirds, the proportionality constant being called the structure parameter C_n^2 :

$$D_n(\rho) = C_n^2 r^{2/3}$$

This means that the turbulence is isotropic. The variation of the refractive index between two points just depends on the distance between them and not on their relative position on the wave front. The value of the structure parameter depends on the type of soil, the height of the path, and the local temperature. The most typical value of this parameter is on the order of $10^{-15} m^{-2/3}$ at $15^\circ C$ and $10^{-14} m^{-2/3}$ at $25^\circ C$.

The most direct effect on the wavefront distortion is the loss of coherence. Ideally, any coherent or incoherent optical source can be considered as a point source when observed from a long distance compared to its size.

3.3. Ambient light

The three main sources of ambient light are sunlight, incandescent lamps and fluorescent lamps [13]. Sunlight represents an unmodulated source of ambient light with a very wide spectral width and a maximum power spectral density located at $\sim 0.5 \text{ nm}$ producing a D.C. photocurrent in the photodetector. Incandescent lamps are modulated at 100 Hz from the mains supply, with a maximum power spectral density around 1 nm , but their slow response time means that few higher harmonics are present. There are two varieties of fluorescent lamps.

The traditional type is driven by the mains frequency and the electrical spectrum contains harmonics into the tens of *KHz*. In the last few years, newer more energy efficient fluorescent lamps have been introduced, driven by high frequency electronic ballasts, with switching frequencies in the 20 – 40 *KHz* range.

Their detected electrical spectrum can contain harmonics into the *MHz* range. Thus, along with contributing to shot noise, fluorescent light sources also produce a periodic interference signal in the receiver.

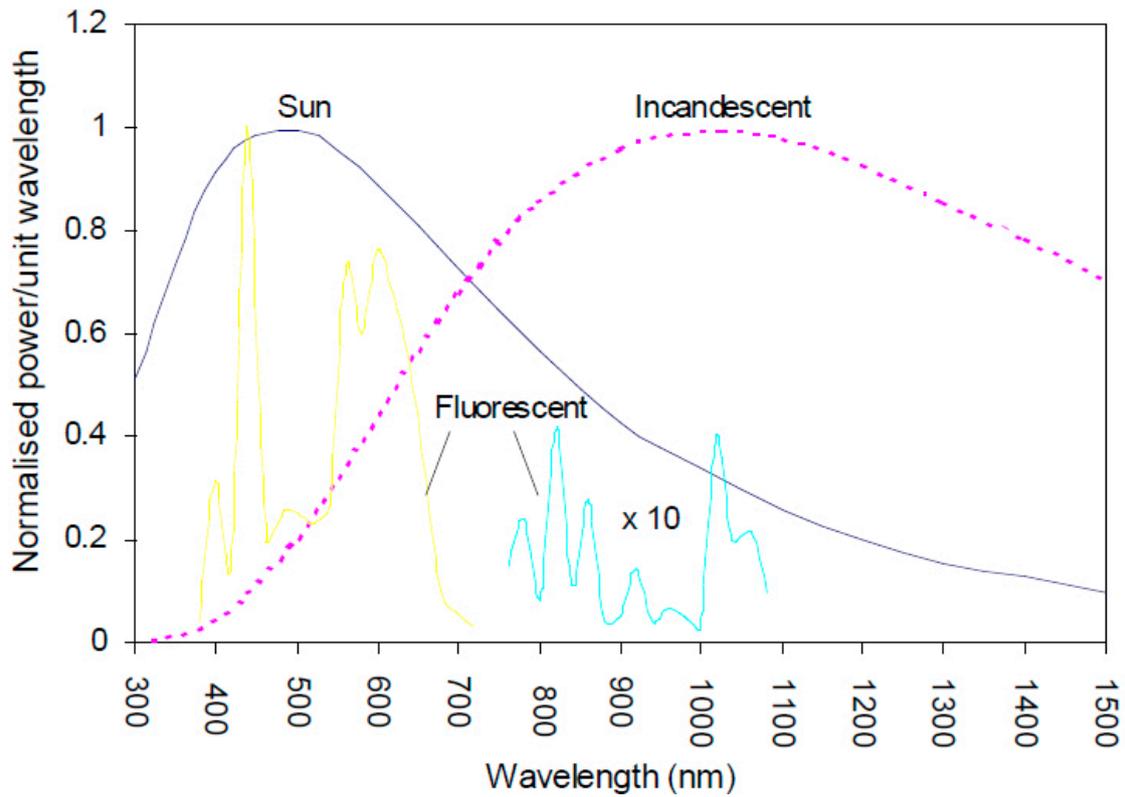


Figure 29 - Optical power spectrum of common ambient infrared sources

4. The surface affects on the link function

The amount of light reflected by an object, and how it is reflected, is highly dependent upon the smoothness or texture of the surface. When surface imperfections are smaller than the wavelength of the incident light (as in the case of a mirror), virtually all of the light is reflected equally. However, in the real world most objects have convoluted surfaces that exhibit a diffuse reflection, with the incident light being reflected in all directions.

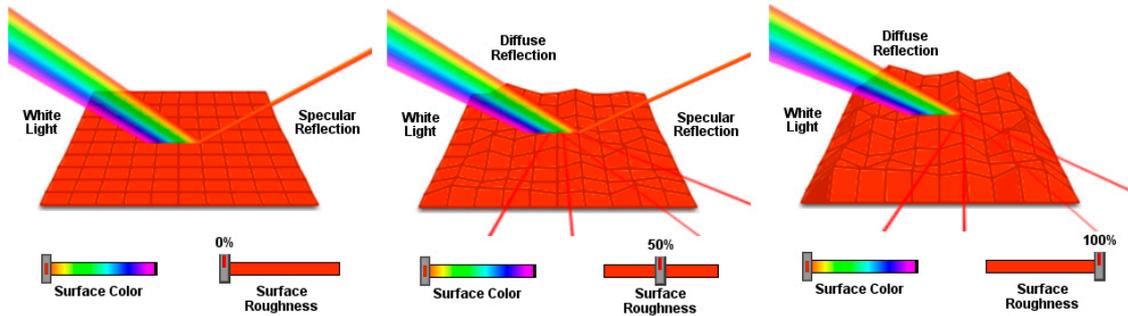


Figure 30 - Diffuse vs. specular Reflection [14]

4.1. Dispersive (diffused)

Diffuse reflection is the reflection of light from a surface such that an incident ray is reflected at many angles rather than at just one angle. An illuminated ideal diffuse reflecting surface will have equal luminance from all directions in the hemisphere surrounding the surface (Lambertian reflectance).

A surface built from a non-absorbing powder such as plaster, or from fibres such as paper, or from a polycrystalline material such as white marble, reflects light diffusely with great efficiency. Many common materials exhibit a mixture of specular and diffuse reflection.

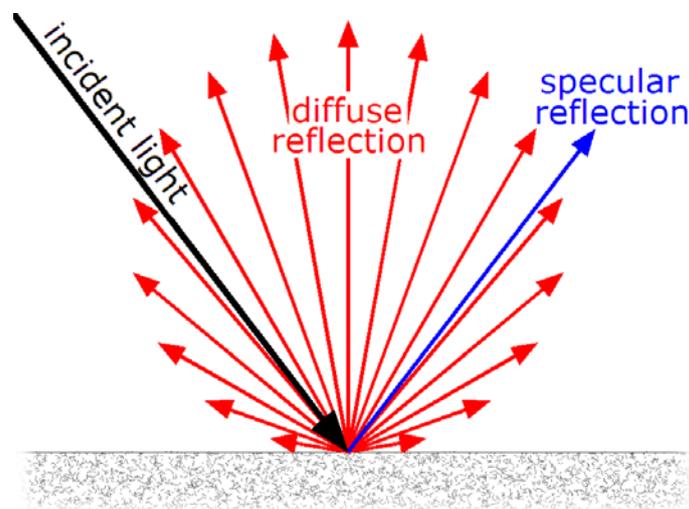


Figure 31 - Diffuse surface [15]

Diffuse reflection from solids is generally not due to surface roughness. A flat surface is indeed required to give specular reflection, but it does not prevent diffuse reflection. A piece of highly polished white marble remains white; no amount of polishing will turn it into a mirror. Polishing produces some specular reflection, but the remaining light continues to be diffusely reflected.

The most general mechanism by which a surface gives diffuse reflection does not involve exactly the surface: most of the light is contributed by scattering centres beneath the surface. For simplicity, "reflections" are spoken of here, but more generally the interface between the small particles that constitute many materials is irregular on a scale comparable with light wavelength, so diffuse light is generated at each interface, rather than a single reflected ray.

This can be very general, because almost all common materials are made of "small things" held together. Mineral materials are generally polycrystalline: one can describe them as made of a 3-D mosaic of small, irregularly shaped defective crystals. Organic materials are usually composed of fibres or cells, with their membranes and their complex internal structure. And each interface, inhomogeneity or imperfection can deviate, reflect or scatter light, reproducing the above mechanism.

Few materials don't follow it: among them metals, which do not allow light to enter; gases, liquids; glass and transparent; single crystals, such as some gems or a salt crystal; and some very special materials, such as the tissues which make the cornea and the lens of an eye. These materials can reflect diffusely, however, if their surface is microscopically rough.

4.2.Specular (reflective)

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.

Such behaviour is described by the law of reflection, which states that the direction of incoming light (the incident ray), and the direction of outgoing light reflected (the reflected ray) make the same angle with respect to the surface normal, thus the angle of incidence equals the angle of reflection and that the incident, normal, and reflected directions are coplanar.



Figure 32 - Example of Specular surface [16]

4.3. Relative directional reflectivity

Reflective and diffusive characteristics of the optical surfaces can be measured by means of the relative directional reflectivity (RDR) or the optical cross section (OCS). The surface reflectivity includes both diffusive and specular components and the directional properties of the surface reflectivity are characterized by the RDR. It is known that the directional distribution of the reflected and diffused components depends on the material, shape and treatment as well as on the direction, polarization and space and time coherence on the incident light.

To use an indoor optical wireless link, we have to study the multiple reflections and multipath distortions on walls and objects. To model the link, reflective and diffusive characteristics can be measured with two methods:

- a) RDR (Relative Directional Reflectivity): allows quantify the reflective and diffusive characteristics.

$$RDR_{M,\alpha}(\beta) = \frac{\pi L_{M,\alpha}(\beta)}{I_i \cos(\alpha)}$$

- $L_{M,\alpha}(\beta)$: Radiance of a real surface at the point M.
- I_i : Radiance of the intensity incident.
- α : Angle between normal line and the direction of incident.
- β : Angle between normal line and the direction of observation.

- b) OCS (Optical Cross Section): value which describes the maximum amount of optical flux reflected back to the source. It is the integral quantity characterizing reflective properties of the entire illuminated surface of the object.

$$OCS = \frac{1}{\bar{k}} \int_{S_{C_n}} RDR_{M,\alpha}(\beta) k(x, y) \cos(\alpha) dS_{C_n}$$

- \bar{k} : Average value of k in cross-section of the beam at the object location.
- S_{C_n} : Projection of the illuminated surface to the plane perpendicular to the direction of radiation.
- $k(x, y)$: Ratio of the local optical intensity and maximum optical intensity in the cross-section of the beam at the object location.

If the cross-section of the beam is smaller than the object projection to the plane perpendicular to the radiation direction, OCS relation can be expressed as local effective OCS:

$$OCS_{loc} = RDR_{M,\alpha}(\beta) \pi w^2 \cos(\alpha)$$

- w : radius of the beam cross-section

4.4.Applications

There are several interesting applications of this distinction between specular and diffuse reflection. One application pertains to the relative difficulty of night driving on a wet asphalt roadway compared to a dry asphalt roadway. Most drivers are aware of the fact that driving at night on a wet roadway results in an annoying glare from oncoming headlights. The glare is the result of the specular reflection of the beam of light from an oncoming car.



Figure 33 - Application of Specular surface

Normally a roadway would cause diffuse reflection due to its rough surface. But if the surface is wet, water can fill in the crevices and smooth out the surface. Rays of light from the beam of an oncoming car hit this smooth surface, undergo specular reflection and remain concentrated in a beam. The driver perceives an annoying glare caused by this concentrated beam of reflected light.

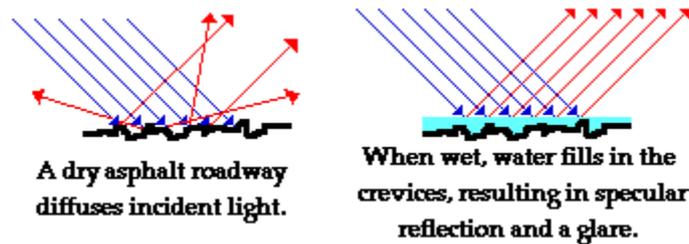


Figure 34 - Application on night driving [17]

A second application of the distinction between diffuse and specular reflection pertains to the field of photography. Many people have witnessed in person or have seen a photograph of a beautiful nature scene captured by a photographer who set up the shot with a calm body of water in the foreground. The water provides for the specular reflection of light from the subject of the photograph. Light from the subject can reach the camera lens directly or it can take a longer path in which it reflects off the water before traveling to the lens. Since the light reflecting off the water undergoes specular reflection, the incident rays remain concentrated (instead of diffusing). The light is thus able to travel together to the lens of the camera and produce an image (an exact replica) of the subject which is strong enough to perceive in the photograph. An example of such a photograph is shown below.



Figure 35 - Example of Specular surface

5. Steady model of the indoor wireless optical link

5.1. Power budget

One of the most important aspects in indoor optical links is the relationship between optical received P_R and OCS (Optical Cross Section). For the measurement of the power received by a system with a receiving aperture A_{RxA} , can be used the following equation [18]:

$$P_R = \frac{1}{\pi} \frac{P_L}{S_{C_n}} A_{RxA} \frac{OCS}{R_L^2}$$

Where P_L is the optical power emitted, R_L is the distance between object and receiving aperture and S_{C_n} is the projection of the illuminated surface to the plane perpendicular to the direction of irradiation.

If the object surface is locally exposed, it means that $S_{C_n} \ll S_c$, the optical received power is:

$$P_R = \frac{P_L}{\pi^2 w^2} \frac{A_{RxA}}{R_L^2} OCS_{loc}$$

The receiving aperture depends on the diameter of the receiving optical system D_{RxA} .

$$A_{RxA} = \frac{\pi D_{RxA}^2}{4}$$

Therefore, substituting the OCS_{loc} expression saw before and the receiving aperture equation we can obtain the following:

$$P_R = P_L \frac{1}{4} \left(\frac{D_{RxA}}{R_L} \right)^2 RDR_{M,\alpha}(\beta) \cos(\alpha)$$

Where α is the angle between the laser transmitter and the ceiling (in this case is 0), $RDR_{M,\alpha}(\beta)$ is the relative directional reflectivity and β is the angle between normal line and the direction of observation.

The relative directional reflectivity can be written as a function of the angle β and the relation between reflected power and transmitted power (ρ). Therefore, the experimental equation for the link is:

$$P_R = P_L \frac{1}{4} \left(\frac{D_{RxA}}{R_L} \right)^2 \rho \cos(\beta)$$

5.2.Link margin

The basic parameter to design an optical link and evaluate the communication system quality is the optical received power (P_R). In real situations, there are more losses which are necessary to take into account such as attenuation of the transmitting optics (α_{TX}) and receiving optics (α_{RX}). In conclusion, if all the terms of the equation are expressed in “dB”, the optical received power can be written as [18]:

$$P_{R;PIN}(dBm) = P_{LED}(dBm) + 10 \log\left(\frac{1}{4}\right) - \alpha_{TX} + 20 \log\left(\frac{D_{RxA}}{R_L}\right) + 10 \log(\rho \cos(\beta)) - \alpha_{RX}$$

$$P_{R;PIN}(dBm) = P_{LED}(dBm) - 6dB - \alpha_{TX} + 20 \log\left(\frac{D_{RxA}}{R_L}\right) + 10 \log(\rho \cos(\beta)) - \alpha_{RX}$$

As we can see, the first term is the power emitted; the next two terms are the attenuation of the optics of the transmitter and the receiver. The fifth one represents the propagation losses and the last term is the reflective and diffusive properties of the optical surface.

In this model of indoor optical links, we are going to work with lambertian surfaces. It means that they are perfectly diffuse reflectors, where the apparent brightness is equal in all directions of view. The relative directional reflectivity is an important parameter that affects the power budget. If the surface has a lambertian character, the RDR is equal to power reflectivity.

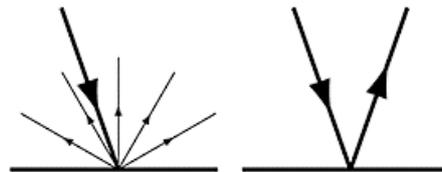


Figure 36 - Diffuse reflection vs. specular reflection

In the Figure 37 we can see a typical power level diagram of an indoor optical link. The intervals between two red points on the horizontal axis represent a specific part of indoor link:

1. After the transmitter
2. After transmitter's optics
3. After the reflexion on the ceiling
4. Before receiver's optics
5. Before the receiver

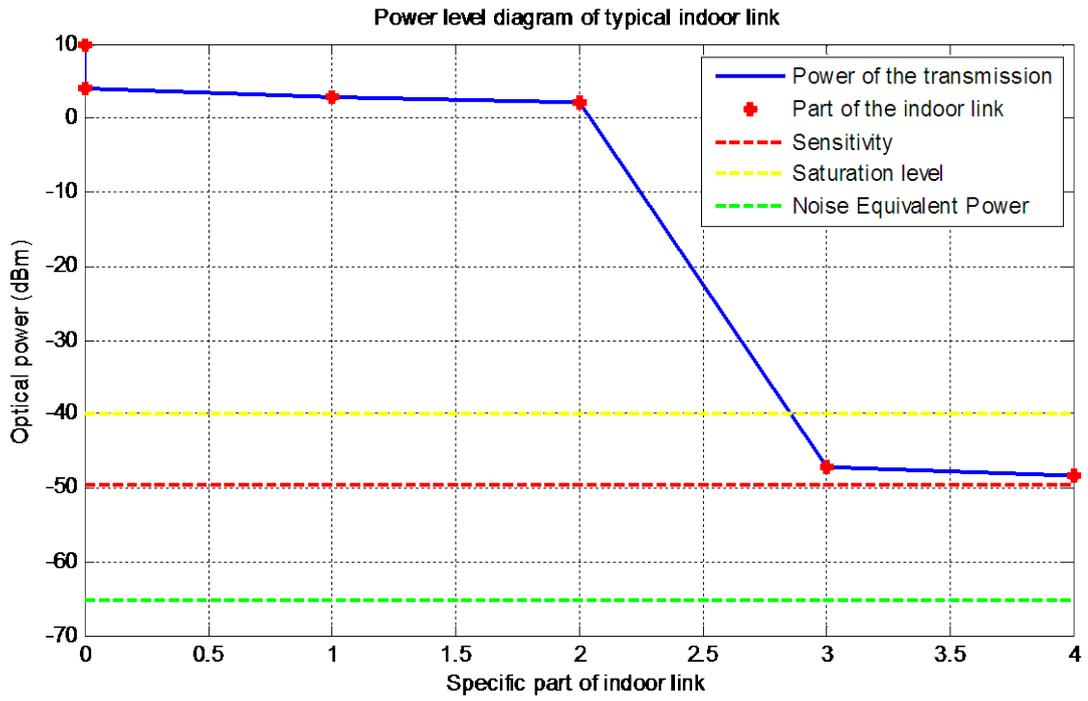


Figure 37 - Power level diagram of the typical indoor link including the RDR function

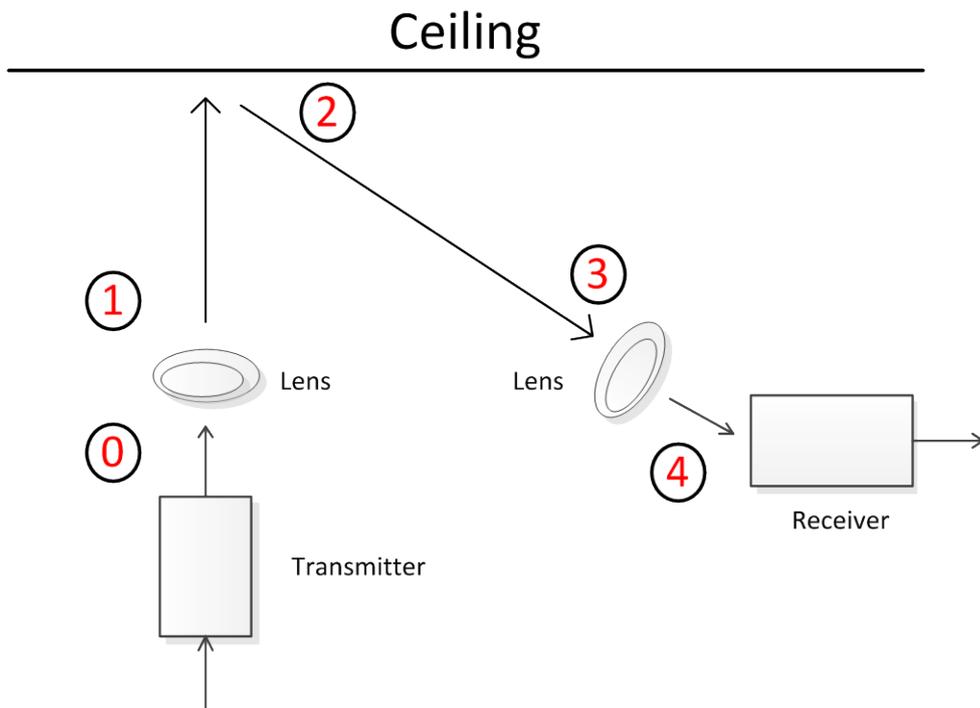


Figure 38 - Specific parts of the link

The blue line is the power of the transmission. We can see how it decreases in each part of the link from the LED transmitter to the photodiode in the receiver at the other side of the link. The photodiode has a dynamic range in which the power received has to be. This range is limited by the receiver sensitivity (red line) and the receiver saturation (yellow line). These values are the minimum and maximum received optical power for a given BER. The green line shows the noise equivalent power of the photodiode, and as we saw before, the sum of the NEP and the SNR is the sensitivity.

5.3. Matlab design

To simulate the indoor optical wireless link, I have used Matlab, a numerical computing environment that allows the simulation with Graphic User Interfaces (GUIs). The main program looks as follow:

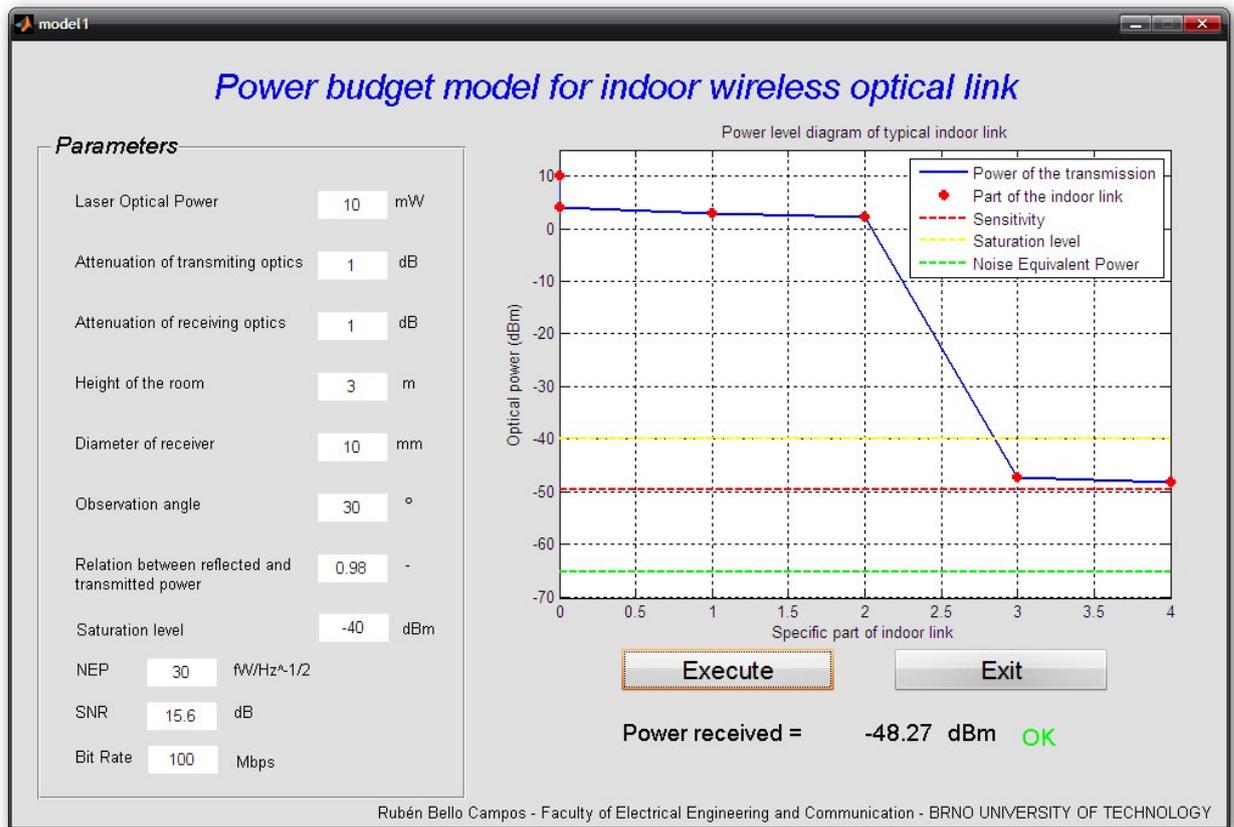


Figure 39 - Main Program

In the window, we can see different parts. Firstly, we have the box of parameters in which there are by default all the characteristic parameters of the optical link. We can change easily all of these parameters and then execute the program. Secondly, on the right of the window it is the diagram of the link with a legend of all types of lines.

Finally, if the power received is between the dynamic range of the photodiode, a message with a green “OK” will appear. If the power received is higher than the saturation or lower than the sensitivity, a message with a red “ERROR” will appear.

Firstly, the structure of the graphical interface is generated by the parameters of the design. Then, the program is centred in the middle of the window. All the spaces of edit are generated to obtain their value later. When we push the button of “Execute”, the first thing we do is obtain these values and assign a variable for each one. When we have all these values, we operate according to the equation described above to obtain the power received.

The next step in the code is compare if the power received is between the dynamic range. If this condition is right, we print in the windows a message with a green “OK”. If is not right, we print a red “ERROR” message.

Finally, we draw the power diagram with all the different parts: power received, parts of the link, noise equivalent power, sensitivity and saturation level.

The second program designed is a simple “reflectivity calculator”. In the main window you can introduce the parameters of the indoor wireless optical link such as transmitter power or noise equivalent power of the receiver, and the program returns the reflectivity needed to ensure the communication.

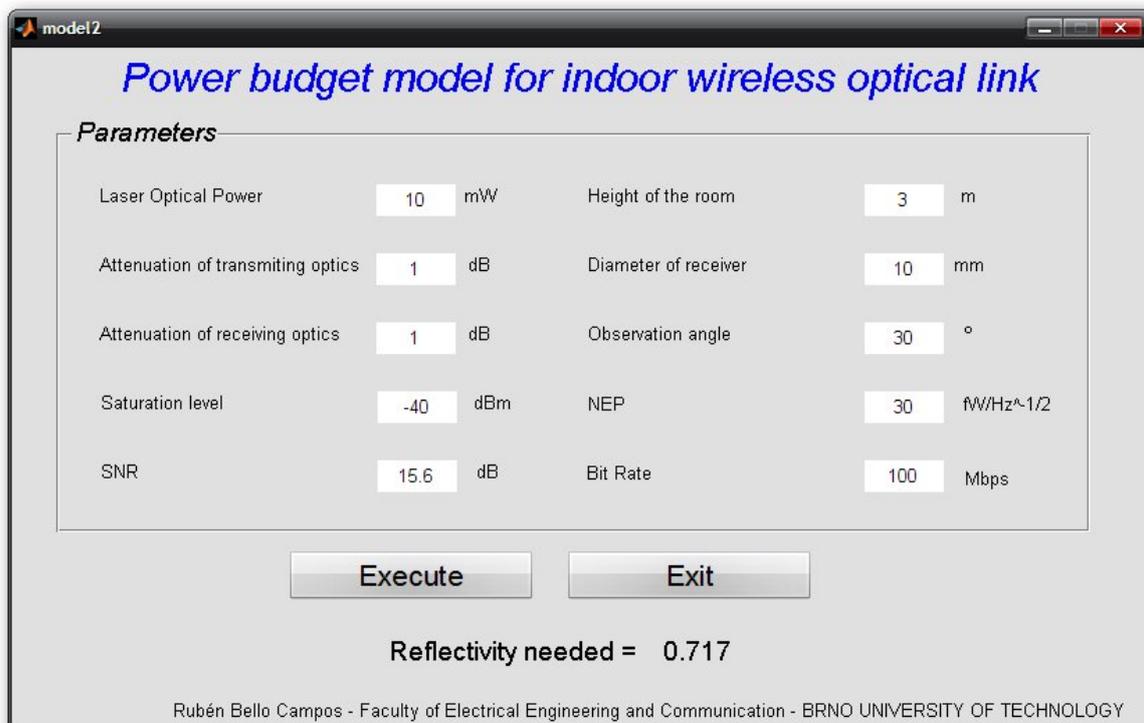


Figure 40 - Reflectivity calculator

5.4.Examples

5.4.1. Black and white ceiling

If we are working in an office where the ceiling is white, we are going to have a reflectivity higher than a black one. In terms of power received, the communication would be very different as we can see in the following examples:

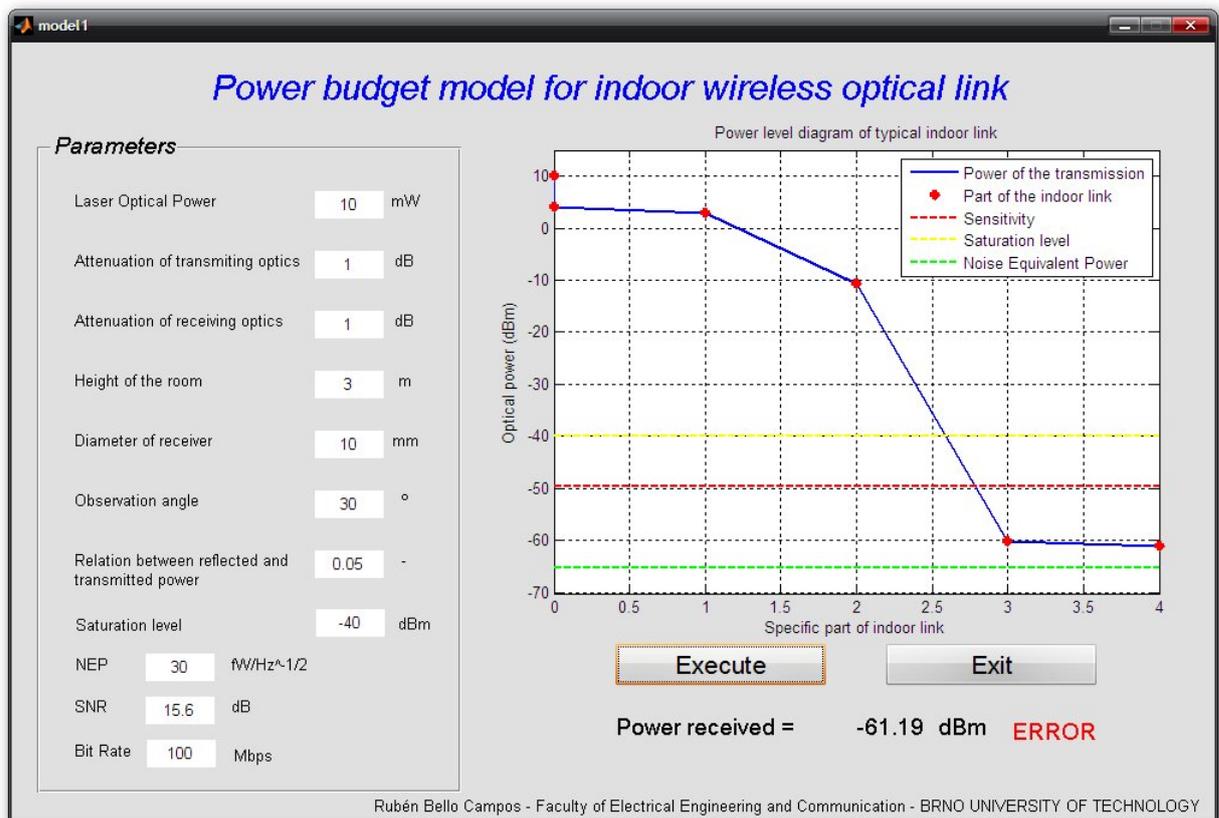


Figure 41 - Power budget with black ceiling

We can see in this figure how the power received is not enough to ensure a communication with a Bit Error Rate of 10^{-9} ($SNR = 15.6 \text{ dB}$). With the same parameters, we are going to calculate the power budget of a white ceiling with a reflectivity of 0.87. In this case, the power received is between the sensitivity and the saturation level, so the link is designed correctly.

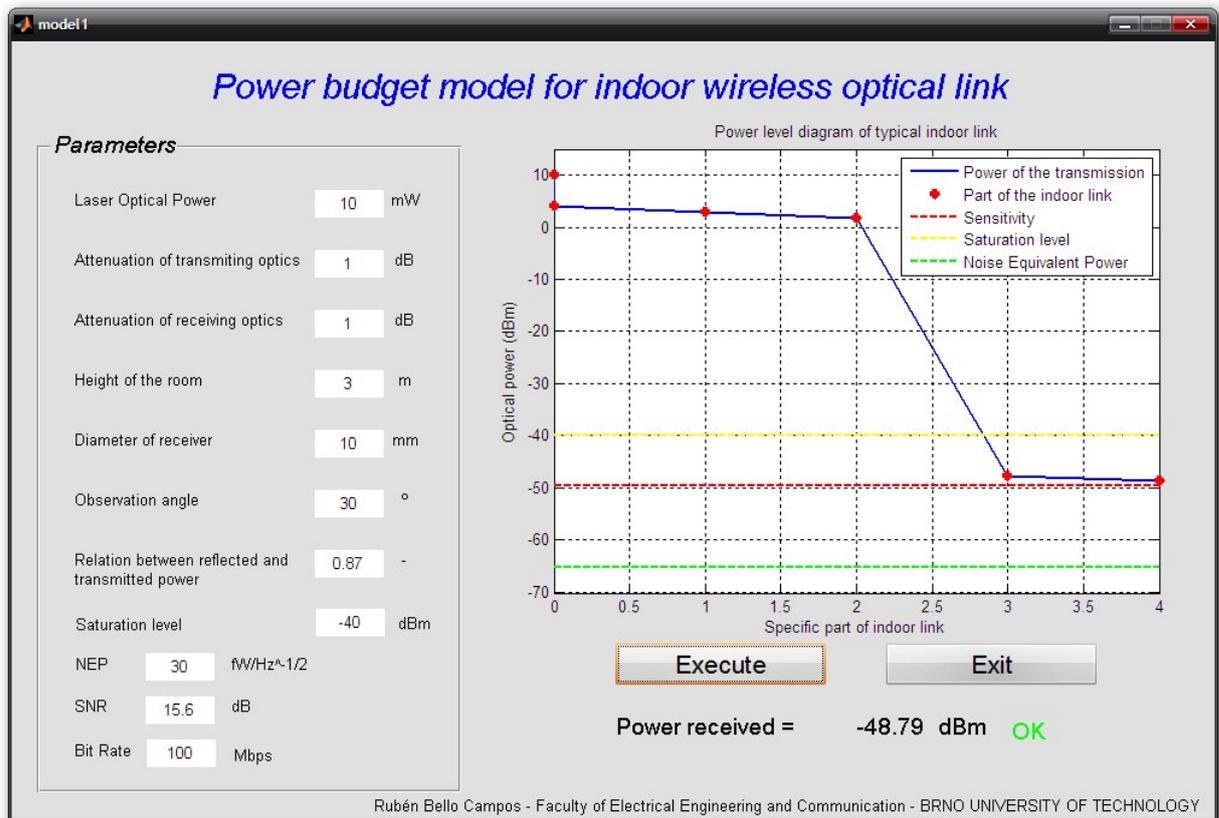


Figure 42 - White ceiling

5.4.2. Modifications of a black and white ceiling

We have seen that a black ceiling has a low reflectivity and this fact affects the power received. To solve this problem, we can modify several parameters of the transmitter and the receiver.

The first thing you can do is to increase the power transmitted. We know that we have to ensure the eye safety so this parameter is not going to change so much the power level. If we cannot change the transmitter, the other option is modify the parameters of the receiver.

The parameter which mainly affects the received power is the noise equivalent power. We can find photodiodes with a low noise which permits to work with a low sensitivity.

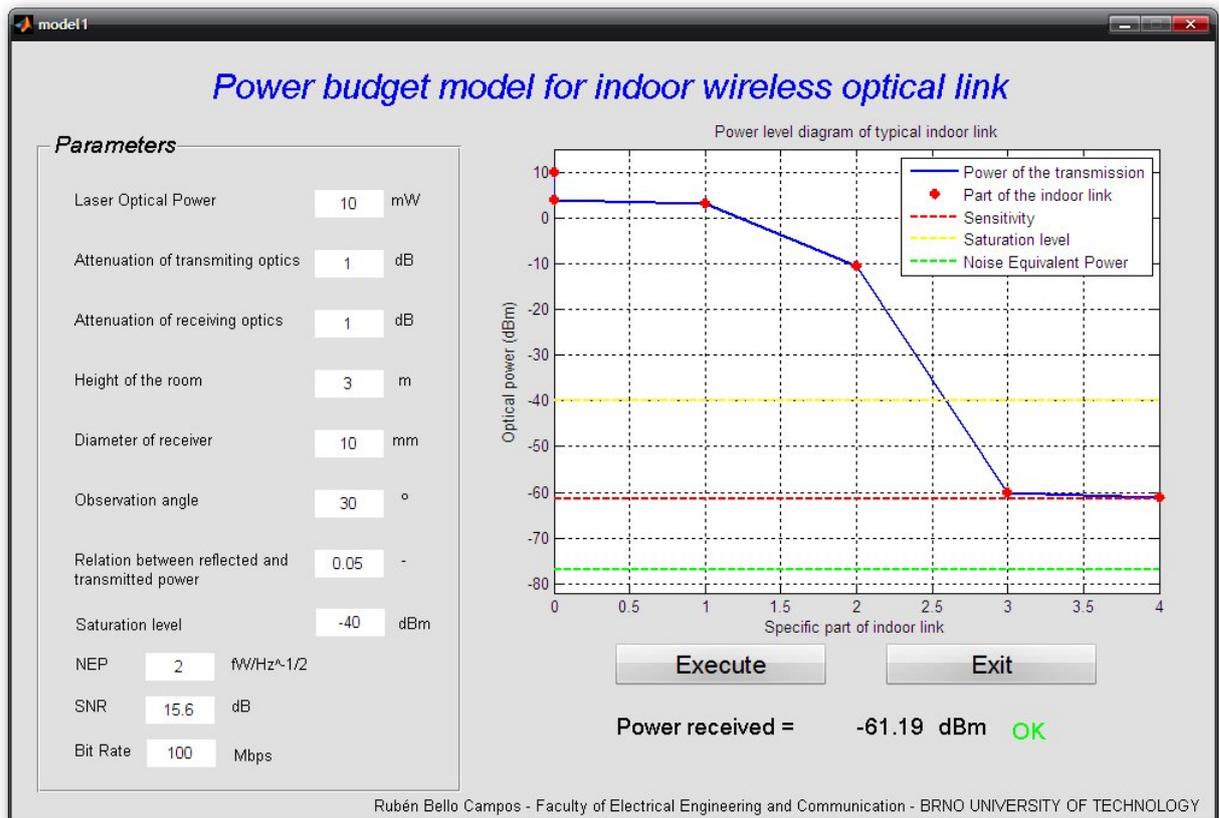


Figure 43 - Black ceiling with modified parameters

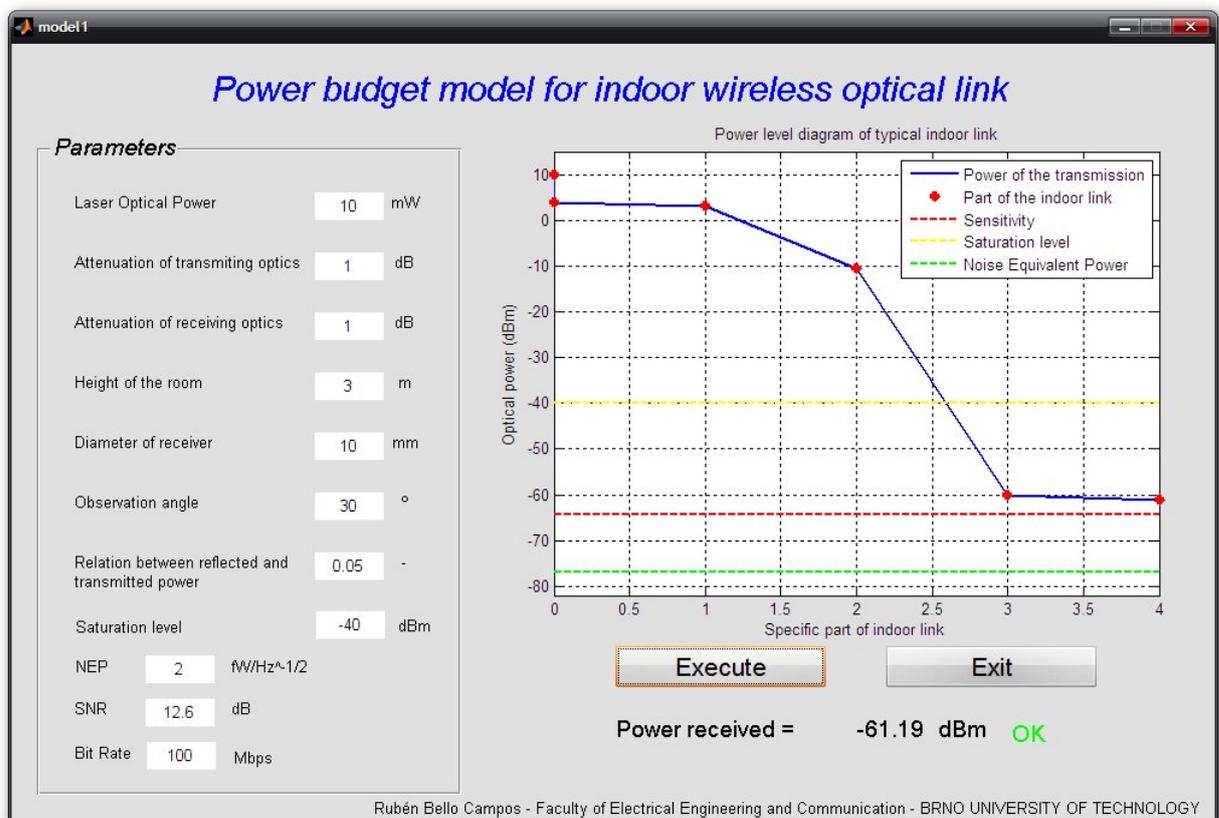


Figure 44 - Black ceiling with modified parameters

Although in the figure 43 we can see that the power received is between the dynamic range, we have not margin between the sensitivity and the power received. This fact will cause problems because some attenuation, however small, will cause very small losses which can weaken the link.

The best solution for this case is to lower the signal to noise relation. Although we will have a worse bit error rate (BER), we have margin to be sure that any attenuation can damage the link.

We can modify the same parameters with a white ceiling. In this case, if we work with a photodetector which has a noise equivalent power of $30 \text{ fW}/\sqrt{\text{Hz}}$ and the signal to noise relation is about 12.6 dB , we are able to transmit less power and the power received will be in the dynamic range.

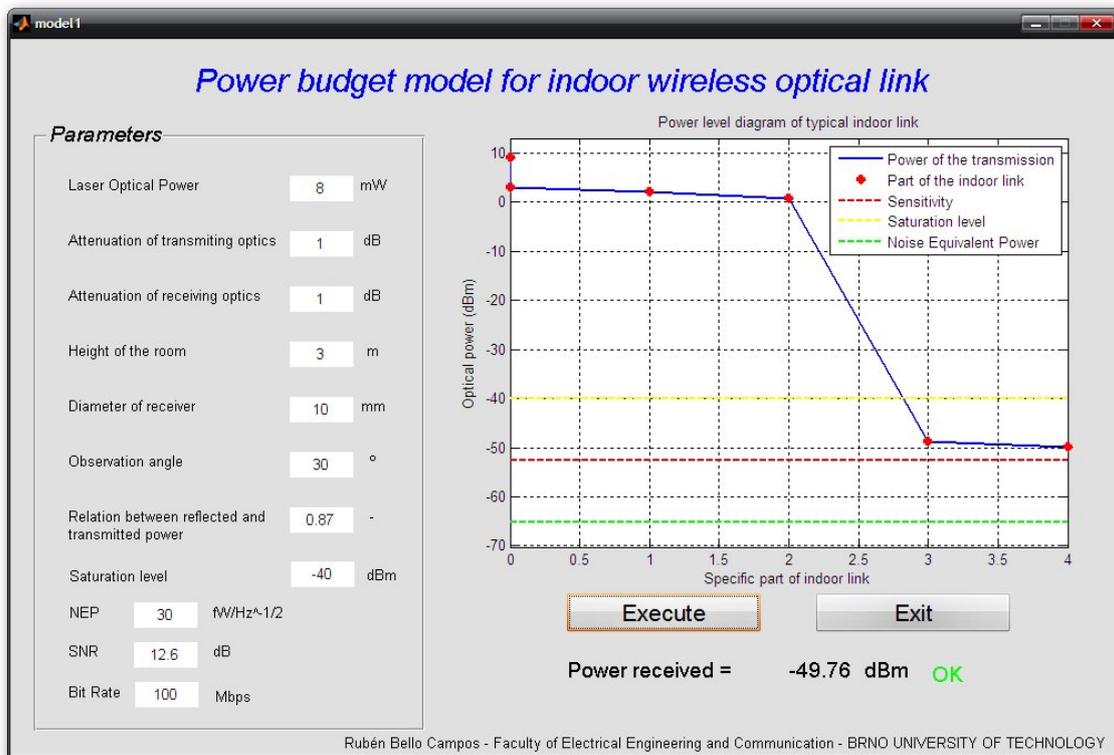


Figure 45 - White ceiling modified

6. Conclusion

After taking contact with the basics of FSO transceivers, their parameters and how they work in a wireless optical link, we could study the complex model of an FSO link. The use of lens is vital for the good performance of the link and they are necessary at both the transmitter and the receiver.

As we saw, the most attractive configuration is the non-directed-non-LOS (diffuse) because of they do not require a direct line of sight or alignment between the transmitter and the receiver. A wireless infrared link must have requirements such as a bit error rate at the same level as cabled LANs and the power levels must be innocuous to human beings.

Because of the link is to be installed in an office environment, the atmospheric phenomena is not a problem for this situation. The only sources which can interference are sunlight, incandescent lamps and fluorescent lamps, but their power is so small that it will not influence the link.

The influence of the surface reflectivity is an important aspect to take into account. Reflective and diffusive characteristics of the optical surfaces can be measured by means of the relative directional reflectivity (RDR) or the optical cross section (OCS). In the power diagram the influence of the given surface (in terms of the surface reflectivity) is included with the propagation losses of the link.

Finally, this thesis presents a model of a wireless optical link with an easy graphic interface programmed by Matlab. The flexibility and simplicity of the program makes it perfect for a quickly configuration of an indoor link.

My contribution to the problem is the application of the relative directional reflectivity (RDR) in the link budget of an indoor wireless optical link.

It is possible to conclude that, in spite of the advances achieved so far, there is still a lot for work to be done to exploit completely the advantages and the potential offered by the optical medium.

7. References

Introduction

- [1] A. Santamaría, F.J. López-Hernández, “Wireless LAN Systems”. London: Artech House, 1994.

FSO transceiver – structure and function

- [2] H. Kressel, G. Arnold. “Semiconductor Devices for Optical Communication”. Springer-Verlag, 1982.
- [3] <http://www.hamamatsu.com>
- [4] <http://www.dsif.fee.unicamp.br/~moschim/cursos/simulation/transmissor.htm>
- [5] [http://en.wikipedia.org/wiki/Lens_\(optics\)](http://en.wikipedia.org/wiki/Lens_(optics))
- [6] http://www.newport.com/Meniscus-Lenses/515893/1033/info.aspx#tab_Drawings
- [7] <http://www.edmundoptics.com/optics/optical-lenses/fresnel-lenses/aspherically-contoured-fresnel-lenses/2039>
- [8] <http://demonstrations.wolfram.com/LensmakersEquation/>
- [9] A. Santamaría, F.J. López-Hernández, “Wireless LAN Systems”. London: Artech House, 1994. Chapter 1.
- [10] Z. Ghassemlooy, A. R. Hayes, “Indoor Optical Wireless Networks”, Electronic Research Group, School of Engineering, Sheffield Hallam University, City Campus, Pond Street, Sheffield, S1 1WB. U.K.

Atmospheric phenomena

- [11] <http://mintaka.sdsu.edu/GF/explain/optics/scatter.html>
- [12] http://en.wikipedia.org/wiki/Rayleigh_scattering
- [13] A. Santamaría, F.J. López-Hernández, “Wireless LAN Systems”. London: Artech House, 1994.

The surface affects on the link function

- [14] <http://micro.magnet.fsu.edu/primer/java/reflection/specular/index.html>
- [15] http://en.wikipedia.org/wiki/Diffuse_reflection
- [16] http://en.wikipedia.org/wiki/Specular_reflection
- [17] <http://www.physicsclassroom.com/class/refln/u13l1d.cfm>

Steady model of the link

- [18] Wilfert, O., Prokes, A., Diblik, J., Hrbackova, P., "Power budget for indoor wireless optical link". In Proceedings of Free-Space and Atmospheric Laser Communications XI. San Diego: SPIE, Volume 8162, 2011, p. 8162W-1 - 6.

8. Appendix A

As we saw before, the code of the main program has a GUI interface, so the most part of it is related with the buttons and edits spaces. The main information about the equation is after the function of the button 4 (execute button), at the end of the code.

```
% Title: Power budget for an indoor wireless optical link
% Author: Rubén Bello Campos - Brno University of Technology
% Created date: 5 - December - 2011
% Last update: 7 - May - 2012

function varargout = modell(varargin)
% modell M-file for modell.fig
%     modell, by itself, creates a new modell or raises the existing
%     singleton*.
%
%     H = MODEL1 returns the handle to a new modell or the handle to
%     the existing singleton*.
%
%     modell('CALLBACK',hObject,eventData,handles,...) calls the
local
%     function named CALLBACK in modell.M with the given input
arguments.
%
%     modell('Property','Value',...) creates a new modell or raises
the
%     existing singleton*. Starting from the left, property value
pairs are
%     applied to the GUI before modell_OpeningFcn gets called. An
%     unrecognized property name or invalid value makes property
application
%     stop. All inputs are passed to modell_OpeningFcn via varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help modell

% Last Modified by GUIDE v2.5 02-May-2012 17:23:15

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @modell_OpeningFcn, ...
```

```

        'gui_OutputFcn', @modell_OutputFcn, ...
        'gui_LayoutFcn', [], ...
        'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if narginout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before modell is made visible.
function modell_OpeningFcn(hObject, eventdata, handles, varargin)
scrsz=get(0,'ScreenSize');
pos_act=get(gcf,'Position');
xr=scrsz(3)-pos_act(3);
xp=round(xr/2);
yr=scrsz(4)-pos_act(4);
yp=round(yr/2);
set(gcf,'Position',[xp yp pos_act(3) pos_act(4)]);
handles.output = hObject;
guidata(hObject, handles);

handles.output = hObject;
% Update handles structure
guidata(hObject, handles);
% UIWAIT makes modell wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = modell_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

function edit1_Callback(hObject, eventdata, handles)
% hObject handle to edit1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

```

```

% Hints: get(hObject,'String') returns contents of edit1 as text
%         str2double(get(hObject,'String')) returns contents of edit1
as a double

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit2_Callback(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit2 as text
%         str2double(get(hObject,'String')) returns contents of edit2
as a double

% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit3_Callback(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

```

% Hints: get(hObject,'String') returns contents of edit3 as text
%         str2double(get(hObject,'String')) returns contents of edit3
as a double

% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit4_Callback(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit4 as text
%         str2double(get(hObject,'String')) returns contents of edit4
as a double

% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit5_Callback(hObject, eventdata, handles)
% hObject    handle to edit5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit5 as text

```

```

%      str2double(get(hObject,'String')) returns contents of edit5
as a double

% --- Executes during object creation, after setting all properties.
function edit5_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%      See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit6_Callback(hObject, eventdata, handles)
% hObject    handle to edit6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit6 as text
%      str2double(get(hObject,'String')) returns contents of edit6
as a double

% --- Executes during object creation, after setting all properties.
function edit6_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit6(see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%      See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit7_Callback(hObject, eventdata, handles)
% hObject    handle to edit7 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit7 as text
%      str2double(get(hObject,'String')) returns contents of edit7

```

```

as a double

% --- Executes during object creation, after setting all properties.
function edit7_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit7 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit8_Callback(hObject, eventdata, handles)
% hObject    handle to edit8 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit8 as text
%         str2double(get(hObject,'String')) returns contents of edit8
as a double

% --- Executes during object creation, after setting all properties.
function edit8_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit8 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit9_Callback(hObject, eventdata, handles)
% hObject    handle to edit9 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit9 as text
%         str2double(get(hObject,'String')) returns contents of edit9
as a double

```

```

% --- Executes during object creation, after setting all properties.
function edit9_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit9 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit10_Callback(hObject, eventdata, handles)
% hObject    handle to edit10 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit10 as text
%         str2double(get(hObject,'String')) returns contents of edit10
as a double

% --- Executes during object creation, after setting all properties.
function edit10_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit10 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit11_Callback(hObject, eventdata, handles)
% hObject    handle to edit11 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit11 as text
%         str2double(get(hObject,'String')) returns contents of edit11
as a double

```

```

% --- Executes during object creation, after setting all properties.
function edit11_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit11 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit12_Callback(hObject, eventdata, handles)
% hObject    handle to edit12 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit12 as text
%         str2double(get(hObject,'String')) returns contents of edit12
as a double

% --- Executes during object creation, after setting all properties.
function edit12_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit12 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function pushbutton4_Callback(hObject, eventdata, handles)
% Laser optical power (mW)
P_tx = str2double(get(handles.edit1, 'String'));

% Transmitter losses (dB)
L_tx = str2double(get(handles.edit2, 'String'));

% Receiver losses (dB)

```

```

L_rx = str2double(get (handles.edit3, 'String'));

% Diameter of the receiving optical system
D = str2double(get (handles.edit5, 'String'));
D=D*10^-3;

% Distance between object and receiving aperture (m)
RL = str2double(get (handles.edit4, 'String'));

% Angle between normal line and the direction of observation (degrees)
beta = str2double(get (handles.edit6, 'String'));
% Angle in radians
beta=beta*pi/180;

% Relation between reflected power and transmitted power
(reflectivity)
p = str2double(get (handles.edit7, 'String'));

% Saturation level
Pr_max = str2double(get (handles.edit9, 'String'));

% Noise Equivalent Power
NEP = str2double(get (handles.edit10, 'String'));
NEP=NEP*1e-15;
% Noise Equivalent Power in mW.
NEP=NEP*sqrt(B)*1000;
% Noise Equivalent Power in dBm.
NEP_dBm=10*log10(NEP);

% Signal to Noise Relation (dB)
SNR = str2double(get (handles.edit11, 'String'));

% Bit Rate (Mbps)
B = str2double(get (handles.edit12, 'String'));
B=B*1e6;

% Sensitivity
S=NEP_dBm+SNR;

% Limits of the diagram
limit1=NEP_dBm-5;
limit2=P_tx+5;

NEP1=[NEP_dBm NEP_dBm NEP_dBm NEP_dBm NEP_dBm NEP_dBm];
S1=[S S S S S S];
Pr_max1=[Pr_max Pr_max Pr_max Pr_max Pr_max Pr_max];

x=[];
y=[];

```

```

% Specific part of indoor link
x=[0 0 1 2 3 4];
y1=10*log10(P_tx);
y2=y1+10*log10(1/4);
y3=y2-L_tx;
y4=y3+10*log10(p*cos(beta));
y5=y4-20*log10(RL/D);
y6=y5-L_rx;
y=[y1, y2, y3, y4, y5, y6];

% Power received
P_rx=y6;
set(handles.text26, 'String', num2str(P_rx));

% Is the power received in the dynamic range?
if P_rx>S && P_rx<Pr_max
    % The power is in the dynamic range
    set(handles.text29,'String','OK','ForegroundColor','g')
else
    % The power is not in the dynamic range
    set(handles.text29,'String','ERROR','ForegroundColor','r')
end

% Diagram
hold off;
plot(x,y,'LineWidth',2)
hold on
plot(x,y,'r*','LineWidth',3)
plot(x,S1,'--r','LineWidth',2)
plot(x,Pr_max1,'--y','LineWidth',2)
plot(x,NEP1,'--g','LineWidth',2)
hold off;
axis([0 4 limit1 limit2])
grid on;
title('Power level diagram of typical indoor link');
xlabel('Specific part of indoor link');
ylabel('Optical power (dBm)');
legend('Power of the transmission','Part of the indoor
link','Sensitivity','Saturation level','Noise Equivalent Power')

function pushbutton3_Callback(hObject, eventdata, handles)
% Close the program
close

```

9. Appendix B

Again, the code of the program has a GUI interface, so the major part of it is related with the buttons and edits spaces. The main information about the equation is after the function of the button 4 (execute button).

```
% Title: Reflectivity Calculator
% Author: Rubén Bello Campos - Brno University of Technology
% Created date: 23 - April - 2012
% Last update: 7 - May - 2012

function varargout = model2(varargin)
% MODEL2 M-file for model2.fig
%     MODEL2, by itself, creates a new MODEL2 or raises the existing
%     singleton*.
%
%     H = MODEL2 returns the handle to a new MODEL2 or the handle to
%     the existing singleton*.
%
%     MODEL2('CALLBACK',hObject,eventData,handles,...) calls the
local
%     function named CALLBACK in MODEL2.M with the given input
arguments.
%
%     MODEL2('Property','Value',...) creates a new MODEL2 or raises
the
%     existing singleton*. Starting from the left, property value
pairs are
%     applied to the GUI before model2_OpeningFcn gets called. An
%     unrecognized property name or invalid value makes property
application
%     stop. All inputs are passed to model2_OpeningFcn via varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help model2

% Last Modified by GUIDE v2.5 02-May-2012 17:23:15

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @model2_OpeningFcn, ...
```

```

        'gui_OutputFcn', @model2_OutputFcn, ...
        'gui_LayoutFcn', [] , ...
        'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before model2 is made visible.
function model2_OpeningFcn(hObject, eventdata, handles, varargin)
scrsz=get(0,'ScreenSize');
pos_act=get(gcf,'Position');
xr=scrsz(3)-pos_act(3);
xp=round(xr/2);
yr=scrsz(4)-pos_act(4);
yp=round(yr/2);
set(gcf,'Position',[xp yp pos_act(3) pos_act(4)]);
handles.output = hObject;
guidata(hObject, handles);
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes model2 wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = model2_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
close;

```

```

function pushbutton4_Callback(hObject, eventdata, handles)

% Laser optical power (mW)
P_tx = str2double(get (handles.edit12, 'String'));
P_tx_mW=P_tx*1e-3;
% Laser optical power (dBm)
P_tx_dBm=10*log10(P_tx);

% Transmitter losses (dB)
L_tx_dB = str2double(get (handles.edit13, 'String'));
% Transmitter losses (lineal)
L_tx_lineal=10^(L_tx_dB/10);

% Receiver losses (dB)
L_rx_dB = str2double(get (handles.edit14, 'String'));
% Receiver losses (lineal)
L_rx_lineal=10^(L_rx_dB/10);

% Diameter of the receiving optical system (mm)
D = str2double(get (handles.edit16, 'String'));
% Diameter of the receiving optical system (m)
D=D*10^-3;

% Distance between object and receiving aperture (m)
RL = str2double(get (handles.edit15, 'String'));

% Angle between normal line and the direction of observation (degrees)
beta = str2double(get (handles.edit17, 'String'));
% Angle in radians
beta=beta*pi/180;

% Saturation level (dBm)
sat_dBm = str2double(get (handles.edit18, 'String'));
% Saturation level (mW)
sat_mW=10^(sat_dBm/10);
% Saturation level (W)
sat_W=sat_mW*1000;

% Noise Equivalent Power (fW/sqrt(Hz)) at datasheets
NEP = str2double(get (handles.edit19, 'String'));
% Noise Equivalent Power (W/sqrt(Hz))
NEP=NEP*1e-15;

% Signal to Noise Relation (dB)
SNR = str2double(get (handles.edit20, 'String'));
% Signal to Noise Relation (lineal)
SNR=10^(SNR/10);

```

```

% Bit Rate
B = str2double(get(handles.edit21, 'String'));
% Bit Rate (Mbps)
B=B*1e6;

% Noise Equivalent Power
NEP=NEP*sqrt(B);

% Reflectivity
p1=L_tx_lineal*L_rx_lineal*((4*NEP*SNR)/(P_tx_mW*cos(beta)))*(RL/D)^2;
set(handles.text52, 'String', num2str(p1));

function edit12_Callback(hObject, eventdata, handles)
% hObject    handle to edit12 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit12 as text
%         str2double(get(hObject,'String')) returns contents of edit12
as a double

% --- Executes during object creation, after setting all properties.
function edit12_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit12 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit13_Callback(hObject, eventdata, handles)
% hObject    handle to edit13 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit13 as text
%         str2double(get(hObject,'String')) returns contents of edit13
as a double

```

```

% --- Executes during object creation, after setting all properties.
function edit13_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit13 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit14_Callback(hObject, eventdata, handles)
% hObject    handle to edit14 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit14 as text
%         str2double(get(hObject,'String')) returns contents of edit14
as a double

% --- Executes during object creation, after setting all properties.
function edit14_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit14 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit15_Callback(hObject, eventdata, handles)
% hObject    handle to edit15 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit15 as text
%         str2double(get(hObject,'String')) returns contents of edit15

```

```

as a double

% --- Executes during object creation, after setting all properties.
function edit15_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit15 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit16_Callback(hObject, eventdata, handles)
% hObject    handle to edit16 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit16 as text
%       str2double(get(hObject,'String')) returns contents of edit16
as a double

% --- Executes during object creation, after setting all properties.
function edit16_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit16 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit17_Callback(hObject, eventdata, handles)
% hObject    handle to edit17 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

```

% Hints: get(hObject,'String') returns contents of edit17 as text
%         str2double(get(hObject,'String')) returns contents of edit17
as a double

% --- Executes during object creation, after setting all properties.
function edit17_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit17 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit18_Callback(hObject, eventdata, handles)
% hObject    handle to edit18 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit18 as text
%         str2double(get(hObject,'String')) returns contents of edit18
as a double

% --- Executes during object creation, after setting all properties.
function edit18_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit18 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit19_Callback(hObject, eventdata, handles)

```

```

% hObject    handle to edit19 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit19 as text
%         str2double(get(hObject,'String')) returns contents of edit19
as a double

% --- Executes during object creation, after setting all properties.
function edit19_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit19 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit20_Callback(hObject, eventdata, handles)
% hObject    handle to edit20 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit20 as text
%         str2double(get(hObject,'String')) returns contents of edit20
as a double

% --- Executes during object creation, after setting all properties.
function edit20_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit20 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function edit21_Callback(hObject, eventdata, handles)
% hObject    handle to edit21 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit21 as text
%        str2double(get(hObject,'String')) returns contents of edit21
%        as a double

% --- Executes during object creation, after setting all properties.
function edit21_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit21 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
%            called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```